

**SOURCE BOOK**  
for  
**Higher Secondary Teachers**

**Computer Science**  
**Class XII**



**Government of Kerala**  
**Department of Education**

**2006**  
**SCERT**

**State Council of Educational Research and Training (SCERT)**  
**Vidyabhavan, Poojappura, Thiruvananthapuram - 12, Kerala**

*Prepared by*

State Council of Educational Research and Training (SCERT)  
Vidyabhavan, Poojappura, Thiruvananthapuram - 695 012, Kerala  
*Website* [www.scert.kerala.gov.in](http://www.scert.kerala.gov.in) *e-mail* [scertkerala@asianetindia.com](mailto:scertkerala@asianetindia.com)

Typeset and design by SCERT Computer Lab

©Government of Kerala • Department of Education • 2006

# *Preface*

*Dear Teacher,*

*We implemented the learner centered, activity based and process oriented approach for teaching and learning of Computer Science in standard XI last year. It was a new experience for many of you and I hope you have owned the pedagogy in the real spirit. Due to many reasons, some of you may not have effectively applied this approach. But this year, all of you are expected to realise this approach in your classes in the true sense.*

*This source book will eliminate all of your ambiguities and apprehensions regarding the new approach. Since our children have to appear for the public examination this year, we should not go for any compromise in implementing the pedagogy.*

*This book has two parts. Part I begins with refreshing the basis and concerns of the approach concisely. In this part more stress is given to enlighten the learning activities that can be tried out in and outside the class. The list of curriculum objectives are given followed by the syllabus details. Detailing about the scheme and planning also is provided. The details given for evaluation consists of the particulars of continuous evaluation (CE), practical evaluation (PE) and term-end evaluation (TE).*

*Part II of this book describes the suggested activities for the effective attainment of curriculum objectives through the transaction of topics related to each chapter in the syllabus. Being teachers in different school environments, you have the freedom to find and try out alternate activities. But, while choosing an activity or strategy, you should ensure the feasibility, students' involvement, possibility of development of process skills, time required, etc.*

*Suggestions as well as innovative learning activities are invited to improve the quality of the book and for the betterment of the learning process.*

*With regards,*

Thiruvananthapuram  
June, 2006

**Dr. E. Valsala Kumar**  
Director,  
SCERT, Kerala

# C O N T E N T S

<b>Chapter No.</b>	<b>Chapter Name</b>	<b>Page</b>
<b>Part I</b>	<b>Approach to Teaching and Learning of Computer Science in Std. XII</b>	<b>7 - 41</b>
1.	Introduction	7
2.	Basis and Concerns of the Approach	8
3.	Teaching - Learning Strategies	13
4.	Curriculum Objectives	19
5.	Syllabus and Contents	24
6.	Planning of Curriculum Implementation	27
7.	Evaluation	29
<b>Part II</b>	<b>Chapter wise Detailing of Learning Activities</b>	<b>45 - 167</b>
1.	Review of C++	47
2.	Structures	55
3.	Concepts of Object Orientation	63
4.	Classes and Objects	69
5.	Constructors and Destructors	79
6.	Function Overloading and Inheritance	87
7.	Pointers	97
8.	Files in C++	109
9.	Boolean Algebra	117
10.	Systems Analysis and Design	127
11.	Concepts of DBMS	135
12.	Structutred Query Language	143
13.	Data Communications	155
	<b>Reference</b>	<b>168</b>

Part I

Approach to  
Teaching and Learning of  
Computer Science  
in Standard XII



# 1

## Introduction

This Source Book is an important aid for teaching Computer Science in standard XII. It is a ready reference for implementing the learner centered pedagogy. We introduced the learner-centered, process-oriented and activity based pedagogy in the higher secondary education of Kerala in the year 2005. It was implemented in standard XI and we teachers have become facilitators for our children who have been uplifted to the status of researchers of various topics under different subjects. Many teachers have enjoyed the new pedagogy while a very few may not have, due to the lack of infrastructure, undesirable strength of the students in the class, improper training or lack of understanding about the pedagogy etc. This Source Book will be of great help particularly for those teachers.

The prime objectives of this book are:

1. To refresh the new pedagogy and be well-planned before beginning the scholastic activities of the new academic year.
2. To get a brief and clear-cut idea about the teaching-learning strategies that can be implemented in standard XII for the effective attainment of the curriculum objectives and/or the transaction of concepts in and outside the class-room.
3. To conduct evaluation processes systematically and transparently without bias.

You are also provided with the curriculum objectives of the subject, syllabus and three levels of planning for the year, each unit and each day. Yet another resource given is the detailed description of the practical evaluation (external) as part of the term-end evaluation. Towards the end of this book, the list of reference books and other web resources are provided.

We begin this book by highlighting the characteristic features of the pedagogy to remind you of the approach to be used in Computer Science education, the role of teachers in the changed classroom, and the strategies to be applied to develop the process skills and/or mental abilities of the students.

So let us clearly understand, own and share the pedagogy to build a dynamic and talented younger generation that possesses enhanced problem solving skills, other process skills, vivid kind of learning experiences and is rich in morale, social commitment, human values etc.

## 2 Basis and Concerns of the Approach

As the subject of Computer Science belongs to the science stream, it is better to learn and teach the concepts through activities so that the learners will be enriched with sufficient problem solving skills and a variety of learning experiences. This could bring unexpected but great benefits to the learners and the society. Knowledge is the product of learning processes which may lead to developments and discoveries. Our children can play an important role in the developments and discoveries in the field of IT and computer, if learning of Computer Science is made process-oriented. We teachers should try to make sure that today's classrooms provide enough learning experiences through various processes by which the new concepts and ideas are constructed. There are sufficient opportunities for activities to enrich the problem solving skills and mental abilities of the learners. Here, the teacher is changing his/her role to that of a facilitator, friend, co-learner, guide, scaffolder etc. In short, learning Computer Science needs to be

- ◆ Activity based
- ◆ Process oriented
- ◆ Learner centered
- ◆ Life oriented

The learning approach of Computer Science focusses on the the construction of knowledge through the following activities:

- ◆ Acquire the facts, ideas, principles and theories.
- ◆ Cultivate skills such as observing, classifying, charting, communicating, using number relationships, predicting, inferring, making operational definitions, interpreting, experimenting etc. to get acquainted with the concepts and principles.
- ◆ Apply and connect the concepts and principles in real-life situations and solving problems to enhance the process skills and mental abilities.
- ◆ Create new software products by identifying and defining problems, formating hypothesis, finding solutions and developing software for the solutions.
- ◆ Utilise the knowledge to inculcate positive attitude and human values to realise the aims of education.

### **Theoretical Foundations of the New Approach**

The basis of the new curriculum is derived from the developments in philosophy, psychology, anthropology and sociology taking place in India and other countries. The new curriculum gives adequate thrust to the following:

- ◆ Constructivist Interactivist Approach
- ◆ Multiple Intelligence
- ◆ Emotional Intelligence

## **The Constructivist Paradigm**

The concept of learning has been changed radically in the past decades. The contribution of Piaget, Vygotsky, Bruner and others have given a new direction to the developments in this area. The following are important among these new concepts:

- ◆ Learning is construction of knowledge.
- ◆ Learning takes place as part of problem solving and other processes.
- ◆ Learning takes place by incorporating new elements of knowledge into the cognitive structure of the learner.

The important features of constructivism may be summarised as follows:

- ◆ Learners must be given opportunities to discover ideas and facts through inquiries.
- ◆ The possibilities of collaborative learning should be exploited by which the learning responsibilities are shared among the members of a group to attain a common objective.
- ◆ To inculcate human values as well as for the optimum utilisation of the resources, co-operative learning can be established, by which learners share their learning experiences and knowledge with others.
- ◆ The level of achievement that can be reached by the learners vary and hence by providing support from the peer group, teachers and others who can contribute to, the learners can reach their proximal level of development. According to Vygotsky, the area between the level achieved by own efforts and that which can be achieved with the help of others is termed as Zone of Proximal Development (ZPD).
- ◆ Teachers should provide help in completing the learning activities to the needy, but not taking over and doing it for the students. Teachers may provide hints, examples, evidences or ask questions to direct the thoughts to the right path. This kind of help, called scaffolding, is inevitable in constructivist paradigm.
- ◆ The learning process is made fully learner-centered, locally specific and life-related by making it interactive, problem based, activity based, project based, etc.

## **Multiple Intelligence**

Modern theories recognise that, by giving suitable environments and through continued efforts, students can improve the intelligence related to a specific area. According to the theory of Howard Gardner, human intelligence has different components and all these components are

---

present in all individuals in different proportions. Some components may be more prominent in some individuals. We should try to provide opportunities to develop these components of intelligence while organising learning activities in and outside the classrooms. The components include the following:

- ◆ **Verbal/Linguistic intelligence**, that provides abilities for reading, writing, verbal production, speaking and lecturing and communicating effectively. This type of intelligence can be developed through activities such as preparing notes, assignments, seminar documents, Lab Diary (Observation Book), project reports etc. and presenting the discussion points, seminars, etc.
- ◆ **Logical/Mathematical intelligence**, that supports abilities like logical thinking and finding out patterns and relations. This type of intelligence is developed through writing algorithms or programs, identifying errors, converting the code segments from one form to another, predicting outputs etc.
- ◆ **Visual/Spatial intelligence**, with which people will be able to design models, draw flowcharts and write readable source codes. Using and developing graphical software and developing presentations with animations will also help develop this component.
- ◆ **Bodily-Kinesthetic Intelligence**, which is related to the ability to move various parts of the body. In computer education, typing skill, mouse movement skills, handling hardware components etc. will be more in those who have this kind of intelligence.
- ◆ **Musical Intelligence**, which is highly developed in people with the ability to distinguish the different elements in music, performing musicians, people who can hum tunes and those who can appreciate music. We can provide opportunities to develop programs using functions or such facilities to incorporate the musical sense of the learners.
- ◆ **Interpersonal Intelligence**, provides leadership qualities and such persons are able to interact in a positive way with others. Debates, role play and other group activities that ensure co-operative and collaborative learning, can be utilised to cultivate this intelligence.
- ◆ **Intra Personal Intelligence**, which is essentially the ability to understand oneself. Such people will have the ability to understand their abilities and shortcomings. We can make the children confident, in developing logic for programs and performing other learning activities, by proper scaffolding and encouragements. Their limitations are to be sympathetically considered and a friendly environment is to be created.
- ◆ **Naturalistic Intelligence**, which is characterised by deep interest in nature, love for fellow beings and interest in spiritual and naturalistic phenomena. The effort to make the desktop of computer attractive, is an instance that reveals the taste of the students. Let us direct it properly.

## Emotional Intelligence and EQ

Recent researchers say that there is evidence for another area of intelligence, namely emotional intelligence. The teacher who aims at improving the emotional intelligence of students need to concentrate on the following.

**(i) *Ability to take decisions***

Rather than imposing decision on students while planning and executing activities, the students may be allowed to take part in the decision making process. While conducting activities like quiz, class test, etc., determining the formats of assignments, project report etc., and fixing the date of submission of assigned works, tests etc., it is better to take decisions through open discussion in the class and inviting students' suggestions.

**(ii) *Ability to reach consensus***

When different opinions, ideas and positions arise, the students may be given the responsibility to reach a consensus. While developing logic, selecting the tools, constructs or facilities of the language, listing merits or demerits of some concepts, comparing similar concepts, etc. students should be allowed to intervene in a healthy way to reach consensus.

**(iii) *Ability to solve problems***

The teacher should make the students aware of the fact that any problem will have a solution and they can find the solution with continued efforts. Students are to be given chances for:

- finding reasons for problems.
- suggesting solutions through individual or group efforts.
- analysing the shortcomings in methods to solve problems.
- discussing the optimality of the solutions.

Daniel Goleman and some others have developed devices for measuring emotional intelligence and the unit of measurement is known as Emotional Quotient (EQ). The EQ is more relevant than the IQ in the process of learning and development. The fact, that one's EQ is the greatest factor affecting one's success in life, is now widely accepted.

## Role of Teachers and Students

If the activities envisaged in the curriculum have to take place in the class room, teachers should take up the following roles as and when needed:

- An assistant who sympathetically solves the problems of students.
- A friend who simplifies and supports learning activities.

- A group leader who shows by example the democratic methods.
- A co-learner who actively participates in learning activities.

In addition to these, the new curriculum envisions the responsibilities of teachers as follows:

- Diagnose the learning activities and provide suitable experiences or remediations.
- Plan and provide challenging situations.
- Continuously evaluate the progress of learners.
- Promote divergent thinking.
- Facilitate inter-personal and intra-personal interactions.

The new approach also envisions the learner in a new perspective. In the learner centered activity oriented pedagogy, the learner :

- Actively participates in the learning process.
- Acts as a researcher.
- Shares information.
- Shares responsibilities.
- Collects information.
- Leads a group.
- Interacts in a group.
- Works as a co-participant.
- Observes and demonstrates.
- Experiments.
- Interprets and makes inferences.
- Judges the merits and demerits of an idea.

## 3 Teaching-Learning Strategies

Learning Computer Science in the classroom takes place through activities like discussion, illustrations, observation, demonstration, quiz, preparation of notes, program writing, error corrections, output predictions, assignments, album preparation, seminars, lab work, class tests, project, etc.

Among these, assignments, notes preparation, question making for quiz and tests etc. are individual activities and could be done at home. But they have to be discussed and shared in the classroom. The group activities should ensure co-operative and collaborative learning. Though the students do the work, the teachers have a crucial role. They are expected to facilitate, encourage, help, guide and scaffold as and when required. The foresaid activities can be classified in another perspective. The activities such as assignment, lab work, class test and project are considered for Continuous Evaluation process, but this does not mean that other activities need not be evaluated. There should be some means for evaluating all the classroom activities. For example, after a discussion to transact a new concept, the students may be asked to prepare notes by covering the points evolved, list the features of some facility, construct a comparison table or illustrate a concept.

The CE items are to be evaluated on the basis of uniform criteria and the scores obtained will be taken into consideration for fixing the grade during term-end evaluation. Being learning activities, they are briefly mentioned in this chapter, but will be detailed in chapter 7 of this book. Let us now see how these activities can be conducted in our class-rooms.

### Discussion

Discussion is the most commonly used medium for communication. To transact the concepts of Computer Science and to develop logic for the programs, discussion is a very good learning activity. Effective discussion will stimulate the intelligence of the learner. It will make the absorption of knowledge easier. Discussion is neither merely asking questions and answering nor explanations given by teachers.

Classroom discussions will become effective only if it is systematically organised. The following points are to be considered while conducting a discussion.

- There should be a problem or a question or a topic that requires clarification or a decision.
- If possible, the topic of discussion should be informed in advance so that the students get an opportunity to think about it and study.
- There should be a self-attempt to communicate the discussion points with others.
- The discussion can be within small groups or in the whole class.

- If it is in small groups, each individual shares his/her discussion points with others in the group and should reach a consensus after negotiations. Sufficient time should be provided and it should be time bound.
- If the whole class is involved in the discussion, strategies are to be used to make effective use of the available time.
- The discussion should be guided properly to make sure that the students are discussing the points that are intended to be covered. Questions or hints may be supplied to make the discussion in the right track.
- The ideas or concepts generated during the discussion within small groups should be presented in the class so that they are shared and refined.
- The teacher should consolidate the discussion and the students should be asked to make notes covering the points that evolved in the discussion.
- The notes prepared by the students may be evaluated in peer by presenting the points in the class in the prescribed format.

## Quiz

Quiz is a better learning activity to brush up the ideas which are essential to proceed to a new topic. The demand for this activity should be naturally evolved so that it will be challenging, need based and motivative. There should be a planning session for the quiz in which the type of quiz, the topics to be covered, etc. are decided. Sample questions may be formed in the class itself and shared to get an idea about framing questions. There should be clarifications by the teacher for the questions and answers wherever necessary during the quiz. Different types of quiz can be conducted as described in the Source Book of standard XI.

## Illustrations

When illustration is taken as a learning activity, we mean that it is the picturisation of a new concept, establishment of the concept with real-life situations, or explanation of the concept with the help of example. Illustrations are best suited for transacting concepts like facilities or features of C++ language and SQL, the syntax for their usage, their working etc. If we picturise the memory allocation for objects and program, we are actually illustrating the concept. First of all a need for the new concept or construct is to be generated by issuing a problem and then the new concept is introduced as its solution with the help of an example. Then allow the students to try themselves to construct the syntax or to develop an operational or working definition of the concept. An open discussion may be conducted for the contribution of others and the teacher should consolidate the points evolved. More exercises may be given to the students for illustration to get a clear-cut idea about the new concepts or syntax.

---

## **Code Manipulations**

Code manipulation is an activity that is exclusively applicable in learning Computer Science. The understanding about the facilities of programming languages and their usage, logical reasoning etc. can be enhanced by providing learning experiences through this activity. By code manipulation we mean error correction, output prediction, code conversion, code modification, code optimisation etc. These activities ensure higher level thinking and enrichment of higher order mental processes. All activities under this category begins by issuing a program or code segment. The activities are performed individually and the products are compared within groups. The consolidated result or product is presented in the class, explaining their findings or its working. Other students or groups are given opportunities to express their opinion about the product and allow manipulations if required. Finally teacher consolidates or refines the findings and all the students take down notes.

### ***Error Correction***

Error correction is an effective learning process for acquiring a good understanding in programming aspects. Incorrect codes written in C++ language are supplied by the teachers and asked the students to identify the errors and correct them without affecting the logic. The formalities of student participation are completed as described above. Note that whenever an error is identified, the reason should be presented and when correction is made, proper justification should be given.

### ***Output Prediction***

A program or code segment in C++ may be given to the students and asked to find the output. Similarly, by supplying tables of database and SQL statements, the students may be asked to find the output of the query.

### ***Code Conversion***

The teacher gives a code segment that uses some facility of the programming language (for example use of if statement) and ask the students to rewrite the code using some other facility (say, using switch statement) without changing the output.

### ***Code Modification***

A program for a specific purpose is given to the students and asked to modify the program so that it can be used for solving some other program. For example, a program to find the sum of the first 10 natural numbers may be given and can be asked to modify the program to find the factorial of 10 (A clue may be given by stating that factorial of 10 is the product of first 10 natural nos.).

### ***Code Optimisation***

When programs are developed and presented in the class, this activity can be executed by setting some constraints in using the facilities of the language or in the logic. For example, in a program to find the factorial of a number, optimisation may be imposed by not allowing the use of

a variable for just controlling the loop. Similarly, in a program for checking a string for palindrome, constraint may be imposed by not allowing an extra string variable or without forming the reversed string. To proceed the activity, we can follow the procedure of other code manipulation activities.

## **Program Coding**

After the transaction of each new concept or facility of C++ language or SQL, problems should be given to write programs for the solution. A discussion may be initiated to define the problem clearly and to develop the logic for the solution. Similar cases may be brought to the attention or hints may be given to frame the logic. Let the students try themselves individually, discuss in small groups and present the code segment. If none of them is able to develop the logic, more clues in the form of missing steps may be supplied and they may be asked to complete the program. The consolidated program having different logics may be compared and optimality may be identified. The logic developed in the class room should be completed as a program individually in the Record Book to get it done in the lab. More problems may given for developing programs for their solutions as home assignments. Such work should be presented and discussed in the classroom.

## **Lab Work**

Lab work is an essential learning activity for the better construction of the programming aspects. It is also considered as a CE component. However the proceedings in implementing this activity is described here in three stages: *Preparation*, *Development* and *Recording*. A book, called Record Book, plays an important role in all of these stages.

### ***Record Book***

The Record Book may be considered as a diary in which student records the activities done in connection with the Lab Work. It will be continuously evaluated by teacher and the summative evaluation will be done by an external examiner during Practical Examination which is a part of TE. It is a collection of program codes that are developed in the class/home and tried out in the lab. The observations, corrections and sample outputs are also noted in the book. The corrected programs may be re-written in the same book, if needed. Teacher verifies and evaluates the contents in it frequently and scores are recorded in his/her evaluation register.

### ***Preparation***

Before entering the computer lab, the students should have clarity about the work they intend to do. They should have their Record Book with them and it should contain the details of work to be done in the lab. In the case of C++, the programs that are discussed in the class and the exercises executed at home should be available so that the work in the computer lab will be very easy and the lab time can also be effectively utilised. The students may be instructed to have a clear-cut idea about the logic and features to be used. Let the learners do as many programs as

possible by ensuring that they are provided with problems that require the application of all the constructs or facilities of the language. Emphasis is to be given for object oriented programming (i.e., using classes and objects) in the case of C++.

In the case of SQL, they should have the structure of table and queries for data manipulation in their Record Book. It should also contain the statements required for the queries that are discussed in the class or prepared at home. There should also be provision to execute SQL statements in the lab with the help of Oracle, MS Access or SQL Server.

### ***Development***

The program development should not be mere typing of the ready-made program. Rather, the students understand the logic clearly and type the program without looking at the written code. The Record Book may be referred to, only if there is no other way. After the completion of program entry, the knowledge about the language should be explored during compilation. Let us not correct the errors directly, but allow them to try individually and in group. Let us give them hints or clues. If we are correcting any error, the reason and remedy should be clarified to the students. Once the error free program is evolved, execute it by using test data. Make sure that the program is free of logical errors and check whether the outputs possess maximum accuracy.

### ***Recording and Evaluation***

Corrections and modifications made in the programs/procedure/steps are to be recorded in the Record Book after obtaining the correct output. If possible, sample output(s) can be noted down. The outputs should be verified and evaluated in peer and by the teacher.

## **Class Test**

While conducting class test, we should make sure that it provides learning experiences to the students. We have to assure the involvement of students in all the stages starting from the preparation of the questions till the recording of scores obtained. We can spend one period for the planning and another period for test and evaluation. This activity is detailed in chapter 7.

## **Assigned Works**

In this section we will discuss some other activities such as preparation of notes, project report, assignments etc. which are done individually at home.

### ***Notes Preparation***

After conducting discussion, the points evolved and consolidated should be written in the note book. The formats may be discussed in the class and they can be in tabular form, lists in bullets or numbers, schematic diagrams, hierarchical charts, brief descriptions etc. The notes prepared by the students may be evaluated in peer and often verified by the teacher. Remedial measures are to be taken for the defaulters.

---

### ***Home Assignments***

These are small works for practising the activities carried out in class. Exercises related to code manipulation activities can be executed at home in their notes or files. They should be shared and verified in the class.

### ***Preparation of Assignment Document***

The document in connection with the assignment taken for CE is also prepared individually at home by following the format prescribed in the class covering the points discussed. It should not be in the note book, but in separate paper for submission. The valued document may be returned to the students as it is their learning material.

### ***Project Report Writing***

A report is to be made as a part of the project work carried out by the students. It should reflect the stages and processes through which the students have completed their project. Even though the project is a group activity, the report should be written individually after each stage of the project, it should be compared with that of group members and verified by the teacher. After completing the project, the rough copy of the project report should be submitted to the teacher and finalisation is done only after the teacher's consent. It is desired to submit one copy from each group to the school library, as it can be a reference material for other students.

### **Seminar**

Even though seminar is not considered as a CE tool for class XII, it can be applied in many situations (*for example chapters 10, 11 and 13*) as it is a very effective learning activity which helps to go deeper into the different aspects of a particular topic in the syllabus. The information collected from various sources with the guidance of teacher are systematically organised and presented in the class so that the information is shared among the students. The students from the audience side can raise doubts and questions regarding the content and even contribute to.

As usual there should be a planning session in which the topics and subtopics (if any) are to be identified and assigned to the students. If there are sufficient content for a particular topic, it may be assigned to a group, but it should be ensured that each member of the group has a role in the work and all of them are work in a collaborative manner. A seminar paper is to be prepared and submitted to the teacher for verification and authentication. When the seminar is presented the students from the audience are expected to record the points in their note books and each student prepares his/her own seminar paper of the same topic. At the same time the student who presents the paper also modifies his/her paper to reflect the points emerged during the discussion. All the papers are compared and evaluated to ensure the uniqueness of the efforts made by each student to acquire the information regarding the topic.

## 4 Curriculum Objectives

Learning a subject depends on the attainment of the curriculum objectives by the learners through various activities. Hence learning Computer Science in standard XII proceeds whenever the learners:

1. Familiarise the concepts about basic data types, operators and control structures in C++ through quiz, discussion, error correction, illustrations, lab work etc. and develop programs.
2. Develop the idea about arrays and functions through discussion and develop programs for array operations.
3. Identify the needs and relevance of user defined data type through discussion, problem solving etc. and prepare notes on them.
4. Familiarize the concept of structure through discussion, problem solving etc. and develop programs using structures.
5. Understand the need and use of array of structures through problem solving and prepare notes comparing arrays and structures.
6. Familiarize the concept of nested structures through discussion and problem solving and define some nested structures..
7. Understand different methods for implementing symbolic constants through illustration, discussions etc. and prepare a comparison chart.
8. Compare procedural oriented approach and OOP through discussion, simulation etc. and prepare a comparison table.
9. Understand the basic concepts of OOPs through illustration of a real life problem and prepare notes on the concepts.
10. Familiarise the concepts of classes and its implementation in C++ through discussion, illustration etc. and define C++ classes.
11. Understand different methods and relevance of defining member function of classes through problem solving, discussion etc. and apply these methods for member function definition.
12. Familiarise the concepts of friend function and inline function and their impact in programming through observation, illustration, coding etc. and develop programs.
13. Identify the need for array of objects and exchanging objects between functions through discussion, illustrations etc. and develop program.

14. Familiarise the concepts of static member function, constant member function and nesting of classes through illustration, discussion, memory allocation charting etc. and prepare notes differentiating them.
15. Identify situations for implementing classes and develop programs using them.
16. Identify the need of constructors through illustration, problem solving, discussion etc. and define constructors for various classes.
17. Familiarise the concepts and characteristics of constructors through discussion and list out the characteristics.
18. Identify different types of constructors through problem-solving, illustration, discussion etc. and develop programs using each type of constructors.
19. Understand different methods for calling constructors with the help of a program and prepare notes.
20. Understand the concept of constructors with the default arguments through a discussion, compare them with default constructors, and list out the points identified.
21. Identify the need for destructors through illustration, observation, discussion etc. and list out the characteristics.
22. Familiarise the concepts of function overloading through function definitions, output prediction etc. and elicit the features of over loading.
23. Identify the role of function over loading in implementing polymorphism through discussion and prepare notes.
24. Recognise the different types of constructors as constructor overloading through observation, discussion etc. and prepare notes by comparing constructor overloading and function over loading.
25. Analyse functions with default arguments and compare them with function overloading and develop sample programs.
26. Formulate the idea about different types of polymorphisms through discussion and prepare notes.
27. Formulate the idea about inheritance through illustrating real life situations and prepare notes.
28. Familiarise the impact of visibility modes in inheritance through illustration, picturing, discussion etc. and prepare a table showing the features of various modes.
29. Familiarise different type of inheritances through illustration of real life situation, charting, discussion and develop the lay out, syntax of each.

30. Acquire the idea about abstract and virtual base classes through observation, discussion etc. and prepare notes.
31. Analyse inheritance, friendship and containership through group discussion and summarise similarities and dissimilarities among them.
32. Familiarise the memory allocation for the execution of a program through observation of chart and prepare notes and diagrams.
33. Familiarise different types of memory allocation techniques through discussion and distinguish them.
34. Formulate the concept of pointers and understand its declaration and initialization through observation, discussion etc. and create and initialise pointers.
35. Familiarise pointer arithmetic through discussion, output prediction etc. and prepare notes on pointer arithmetic.
36. Understand the memory allocation operators through discussion and list out its use.
37. Familiarise the memory leak problem through illustration and list out the situation that result in memory leak.
38. Familiarise the use of pointers in handling string data through illustration by an example and develop programs for string manipulation.
39. Examine the role of pointers as function arguments through the discussion, output prediction etc. and develop programs.
40. Examine the role of structure pointers and self referential structures through illustration, discussion etc. and prepare notes covering definitions of self referential structures.
41. Understand the role of pointers to objects through illustrations, discussion and develop sample program.
42. Familiarise the concept of linked list through the illustration of real life situations and list down its advantages.
43. Acquire the knowledge about stream and familiarize stream object through discussion and prepare notes.
44. Familiarise the operators and functions involved in the file operation through illustration and develop sample program.
45. Understand the concept of file modes through discussion, illustration, output prediction etc. and correct the erroneous code segments.
46. Familiarise the idea about random access and error handling function through problem solving, discussion etc. and prepare a list of functions in each category.

47. Integrate the features and facilities of C++ to develop a software for real life application through a project.
48. Acquire the knowledge about basic gates through discussion, illustration and prepare circuit diagrams for Boolean expressions.
49. Understand the basic postulates and theorems in Boolean Algebra through discussion, illustration etc. and prepare a chart showing them.
50. Familiarise the concepts of principles of duality through observation, discussion etc. and find the dual of given expressions.
51. Familiarise evaluation of Boolean expression through illustration, discussion etc. and work out some examples.
52. Familiarise advanced gates through table preparation, discussion etc. and prepare chart showing them.
53. Develop skills for implementing basic logic gates using universal gates through illustration truth table creation etc. and draw corresponding circuit diagram.
54. Acquire knowledge about applications of logic gates through discussion, illustration etc. and develop circuit diagrams for sample applications.
55. Understand the primary characteristics of a system so as to define it through general discussion.
56. Establish the need and importance of system and design through discussion, illustration of real life examples etc. and prepare notes.
57. Understand the phases of system development life cycle (SDLC) and familiarize the various activities involved in system study and system analysis through group discussion, illustrations etc. and write notes on SDLC.
58. Familiarise various aspects of system design through discussion, illustration etc. and list down the components involved in system design.
59. Acquire knowledge in system implementation, maintenance and review through discussion, case study etc. and prepare notes.
60. Identify the advantages of DBMS over conventional system and recognize the components of DBMS through discussion and real life examples and write notes.
61. Understand the detailed structure of DBMS through illustration and prepare a chart.

62. Understand the various levels of database abstraction and the concept of data independence through observation, discussion etc. and prepare notes with diagrams.
63. Familiarise the different data models through discussion, comparison etc. and form a table showing the features of each model.
64. Acquire the concept of RDBMS and its terminologies through general discussion and prepare a glossary for RDBMS.
65. Understand various relational operations through problem solving, illustrations, discussion, and write statements for relational operations.
66. Understand the different processing capabilities of SQL and its application through discussion and list them.
67. Identify the various data types available in SQL through group discussion and prepare a table describing the SQL data types.
68. Familiarise different commands used in SQL and write an assignment to apply them for table creation and manipulation for various real-life situations.
69. Familiarise column constants and clauses in SQL through illustration, output prediction etc. and list out them.
70. Familiarise different SQL functions through illustration, output predictions etc. and use them for table manipulations.
71. Understand the concept of data communication and compare different modes of transmission through discussion, illustration etc. and prepare notes by comparing different modes.
72. Understand various types of communication through discussion, illustration etc. and prepare charts with proper examples.
73. Familiarise various communication channels through observation, comparison, diagrammatic representation etc. and prepare detailed notes.
74. Acquire knowledge about protocols and understand the various types of protocols through discussion.
75. Familiarise different data access methods through discussion, role play etc. and prepare notes on each of the access methods.
76. Understand features and application of various data communication devices through general discussion, and prepare notes on each device.
77. Familiarise various data communication terminologies related to Internet through observation, experimentation etc. and prepare notes.

### **Module CS I: Object Oriented Programming**

#### **Chapter 1: Review of C++**

Basic Data types; Operators; Control Structures; Arrays and Functions.

#### **Chapter 2: Structures**

User defined Data types; Structure – Definition, Creation of structure variable, Referencing structure elements; Array of structures; Nested structures; Use of typedef; Enumeration – Defining enum type, declaration, changing default ordinal values; Symbolic Constants – const, enum, #define.

#### **Chapter 3: Concepts of Object Orientation**

Procedural V/s Object Oriented Programming; Concepts of OOP – Abstraction, Encapsulation, Modularity, Inheritance, Polymorphism.

#### **Chapter 4: Classes and Objects**

Class Declaration; Data Members; Member Functions; Access Labels; Default Labels; Data hiding and Encapsulation; Arrays within a Class; Methods of Member function definitions; Scope Resolution Operator; Inline functions; Creating objects; Accessing members; Array of objects; Objects as function arguments – Pass by value and reference; Memory allocation of Class and objects; Static Members; Friend functions; Constant Member functions; Nesting of Classes.

#### **Chapter 5: Constructors and Destructors**

Need of Constructors; Declaration & Definition; Types of Constructors – Default, Parameterized, Copy; Call to Constructors – Implicit and Explicit; Constructors with Default arguments; Characteristics of Constructors; Destructors – Definition and Use; Characteristics.

### **Module CS II – Advanced C++ and Boolean Algebra**

#### **Chapter 6: Function Overloading and Inheritance**

Function Overloading – Declaration, Definition and Use; Calling Overloaded functions; Constructor overloading V/s Function overloading; Functions with Default arguments V/s Overloading; Types of Polymorphism – Static and Dynamic (only theoretical concept).

Inheritance – Concept; Base and Derived Classes; Defining Derived class; Visibility Modes; Types of Inheritance; Abstract Class and Virtual Base Class (only theoretical concept); Derivation V/s Friendship; Inheritance V/s Containership.

### **Chapter 7: Pointers**

Memory mapping for a Program; Dynamic and Static Memory allocation; Declaration and Initialization of Pointers; Pointer Arithmetic; Dynamic Allocation/De-allocation Operators; Pointers and Arrays – One dimensional and Two dimensional; Memory leak; Pointers and String; Pointers as Function arguments; Pointer to a Structure; Self-referential structures; Pointers to objects; Concept of Linked Lists.

### **Chapter 8: Files in C++**

Use of header file fstream.h; Types of stream objects – fstream, ifstream, ofstream; Reading and writing characters from/to disk; Detecting end of file; open() and close() functions; Reading and writing objects from/to disk using read() and write() functions; File mode Constants; File pointers for Random Access – seekg(), tellg(), seekp(), tellp(); Error handling functions.

### **Chapter 9: Boolean Algebra**

Basic Logic Operations – OR, AND, NOT; Truth Tables; Basic Gates – OR, AND, NOT; Basic Postulates of Boolean algebra; Basic theorems of Boolean algebra; De Morgan's theorems; Principle of Duality; Evaluation of Boolean expressions (using Truth table and Algebraic method); Advanced Gates – NAND, NOR, XOR, XNOR; Implementation of Basic Logic Gates using NAND and NOR Gates; Application of Logic Gates – Half adder, Full adder, Encoder, Decoder (using Basic Logic Gates).

## **Module CS III – DBMS and Data Communication**

### **Chapter 10: System Analysis and Design**

Definition of System; Why System Analysis and Design?; System Development Life Cycle – System Study (Preliminary Survey, Feasibility Study); System Analysis (Investigation and Fact Recording, Interviewing, Questionnaire, Onsite Observation); System Design (Input, Output, Files, Procedure); Implementation, Maintenance and Review.

### **Chapter 11: Concepts of DBMS**

Databases; Advantages over Conventional file system; DBMS and its structure; Components of DBMS – Databases, DDL, DML, Users; Database Abstraction – Various Levels; Data Independence; Types of Users of Database; Data Models – Relational, Network, Hierarchical;

---

Elementary Terminologies of RDBMS – Relation, Domain, Tuple, Attributes, Cardinality, Degree, Views, Keys (Primary, Candidate, Alternate, Super, Foreign); Relational Operations – Select, Project, Cartesian Product, Union, Intersection, Set Difference.

### **Chapter 12: Structured Query Language**

Processing Capabilities; DDL; DML – Retrieving, Manipulating, Updating tables; SQL Data types – Numeric, Character, Date, Time; SQL Commands – CREATE TABLE, CREATE VIEW, DROP TABLE, ALTER TABLE, SELECT, INSERT, DELETE, UPDATE; Column Constraints – UNIQUE, PRIMARY KEY, NOT NULL, DEFAULT, CHECK; Clauses used with SQL Commands – DISTINCT, INTO, FROM, WHERE, LIKE, GROUP BY, ORDER BY, HAVING, SET; SQL functions – AVG, COUNT, MAX, MIN, SUM.

### **Chapter 13: Data Communication**

Concept of Data communication; Modes of Transmission – Digital V/s Analog, Serial V/s Parallel, Synchronous V/s Asynchronous; Types of Communication – Simplex, Half Duplex, Full Duplex; Modem; Communication Channels – Twisted Pair Cables, Coaxial Cables, Optical Fiber, Radio Waves, Micro Waves, Satellites; Communication Protocols – FTP, HTTP, TCP/IP; Access methods - CSMA/CD and Token Passing; Data Communication Devices – Repeater, Bridge, Router, Gateway; Backbone Network; Data Communication Terminologies in Internet – WWW, Website, Web page, HTML, URL.

## 6 Planning of Curriculum Implementation

A proper planning is required for the implementation of the curriculum in the real sense. Planning is required in three levels - year plan, unit plan and daily plan. The year plan is given in this section, which may be modified conveniently in the School Resource Group for the effective implementation of the curriculum depending on your school environment. This will be helpful for planning the common programmes to be conducted in and outside the school as a part of the curriculum. Activities that are common for various subjects can be clubbed together so that the resources can be utilised to the maximum.

### Year Plan

Term	Month	Chapter	Periods	Total
I	Jun	1. Review of C++	20	32
		2. Structures	8	
		3. Object Orientation	4	
	Jul	3. Object Orientation Contd...	2	32
		4. Classes and Objects	22	
		5. Constructors and Destructors	8	
	Aug	6. Function Overloading and Inheritance	14	24
		7. Pointers	10	
		First Term Examinations		
II	Sept	8. Files in C++	12	20
		10. Systems Analysis and Design	8	
		Project work Planning		
	Oct	Project - Analysis & Design	8	
Nov	9. Boolean Algebra	16		
	11. Concepts of DBMS	8		
	Nov	11. Concepts of DBMS Contd..	8	32
		12. SQL	20	
	Dec	Project Completion	4	16
		Project Submission and Viva	16	
III	Jan	Second Term Examinations		24
		13. Data Communications	16	
		Model TE	8	

The unit plan is given along with the detailing of learning strategies of each chapter in Part II of this book. It will be in tabular form and it help the teachers to capture the chapter at a glance and be prepared for the class. The format is given below:

Chapter No. & Title						
Curriculum Objectives	Concepts/ Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)

The daily planning can be done by the teacher by himself/herself or in consultation with other computer teachers, if any. The planning is to be recored in a notebook, say teaching mannuel, preferably in the following format:

Planning Page	Response Page
<p>Unit/Chapter/Topic: Review of C++</p> <p>Date &amp; Period: June 1<sup>st</sup> week (8 Pds.)</p> <p>Objectives:</p> <ol style="list-style-type: none"> <li>1. Familiarise the tokens and control statements</li> </ol> <p><b>Activities:</b></p> <ol style="list-style-type: none"> <li>1. Students are asked to prepare a list of questions for conducting Quiz by referring the notes, TB and OB of std. XI</li> <li>2. Grouping into 3</li> <li>3. Topic for each group is to be provided.</li> <li>4. Sample questions are to be developed in the class.</li> </ol>	<p>Qns. could have been collected earlier so that better questions would have come out.</p> <p>Some students prepared with qns. but not with answers..</p> <p>Some of them have no idea about looping.</p> <p>Remedial measures to be planned.</p>

Evaluation helps to assess the rate of progress of the students and to recognise to what extent each student has achieved the study target. This is an assessment of how and to what extent the student has achieved the specified capabilities. These evaluation results are the basis on which the student, teachers, parents and society can assess the educational progress.

### Components of Evaluation

In order to flourish continuous and comprehensive evaluation we make use of the following components of evaluation.

1. Continuous Evaluation (CE)
2. Practical Evaluation (PE)
3. Terminal Evaluation (CE)

### Continuous and Comprehensive Evaluation (CE)

Most of our traditional evaluation methods are related only to the area of knowledge. There are criticisms about their comprehensiveness and usefulness precisely because of that. As of now, it is a weakness of our evaluation method that practical capabilities are not being assessed in any subject. It is to rectify this deficiency that the evaluation method is being changed to include assessment of practical capabilities, attitudes, products associated with study, and so on. It is through continuous evaluation that the knowledge related fields and products are evaluated. This is possible only that way. Through this every learning activity that takes place within an year can be subjected to continuous evaluation. Evaluation should be done not by the teachers alone. Students can evaluate themselves. Classmates can evaluate one another. These should be considered as means of improving education.

The CE items are to be evaluated based on the standardised criteria. There are five criteria for each CE item depending on the nature of the activity. Each of the criteria is assigned with a maximum of 4 scores. That is, if a criterion is satisfied for a CE item, it can be given 1, 2, 3 or 4 scores depending on its quality. Usually each criterion is associated with each stage of an activity as there are different stages for the completion of each activity. Teachers are liable to record the score for each criterion or indicator of each CE item of each learner from time to time. The scoring is mainly intended for uplifting and scaffolding the weaker section and not for filtration.

By considering the practicability and the nature of the subject, the following tools are advised for continuous evaluation.

- |            |               |               |                      |
|------------|---------------|---------------|----------------------|
| 1. Project | 2. Lab work   | 3. Assignment | 4. Class test        |
| 5. Seminar | 6. Collection | 7. Debate     | 8. Model preparation |

Teachers have freedom to extend the list, but evaluation criteria should be fixed, transparent and objective.

When we look into these items we can see that grouping or categorizing of them is possible. Categorizing of the tools give freedom to the teacher in selecting the most suitable and appropriate tool for continuous evaluation. Let us group them under five heads as given below.

### **1. Investigative Activities**

This category give stress to the following aspects.

- To inculcate spirit of inquiry and discovery
- To create research mind
- To induce spirit of investigation

The activities in this area include project and field study. But in Computer Science, it is proposed that a project to develop a software is ideal.

### **2. Interactive Activities**

The quality concerns in this category are

- Give scope to collect first hand information about specific issues/areas.
- Enable the learner to precise the idea about a theme.
- Enable the learner to present the ideas in a group and to answer the queries and issues raised in the discussions.

Items under interactive activities in our subject are seminar and debate.

### **3. Practical Work**

The points to be focused here include:

- To enable the learner to apply the ideas/concepts in practical situations.
- To collect and use materials/data collected for practical work.

In Computer Science, items under practical work are Lab Work and practicals/experiments

### **4. Performance Based Activities**

Focal point in this category is to exhibit performance in the various class based works/ activities/programmes. We include class test under this category.

### **5. Assigned Tasks (Activities)**

Quality concerns here are

- To enable the learner to collect additional information with regard to the area under study.
- To collect new information and utilize them for better understanding.
- To enable the learner for creative performance/presentation.

Items under assigned activity is assignment, collections and model preparation.

### Distribution of CE Items

No.	Area	Items
1.	Investigative Activities	Project, Field study
2.	Interactive Activities	Seminar, Debate
3.	Practical Work	Labwork/Practicals
4.	Performance Based activities	Class Test
5.	Assigned Tasks	Assignment, Collection and Model Preparation

*Note: Teacher should select any 4 areas (select suitable item from the selected area) from the given 5. There is absolute freedom to select one CE item from each area. Evaluation may be done on the basis of specific indicators and should record in the Student's Evaluation Profile (SEP) under the head CE.*

## How to Evaluate CE Items?

This section deals with the learning activities considered for continuous evaluation. Each of these activities known as CE items is evaluated based on five indicators. Let us see the evaluation activities to be considered in standard XII along with their evaluation indicators.

### 1. Assignment

Assignment is an activity to achieve the curriculum objectives undertaken by the students, in continuation with activities carried out in the class. It is a self learning cum evaluation activity and it should ensure that the work is completed within a stipulated time according to the teacher's directions by utilising the maximum capabilities of the students and exploring maximum possibilities. If same topic is given to all the students, the involvement of the student in his/her work should be assured using some tools like viva voce. If different tasks are assigned, the level of tasks should be uniform and evaluation indicators should be specified in advance. The activity begins in the class room with proper planning and preliminary discussion and carried out completely outside the class. In the planning session, the data to be collected, their sources, the format of assignment and the date of submission should be discussed and finalised. The collected data or information should be authenticated by consulting with the teacher. The document containing the assigned task should be submitted in time for the completion of evaluation. The document may be shared if it is relevant. In standard XII, developing SQL statements for table creation and manipulation may be given as assignment. Another possibility is chapter 12 by which applications of the postulates and theorems may be given as assignment.

***Evaluation Indicators:***

1. Data Collection	4/3/2/1
2. Comprehensiveness of the content	4/3/2/1
3. Systematic and sequential arrangement	4/3/2/1
4. Awareness of the content	4/3/2/1
5. Form and structure of the product	4/3/2/1
<b>Total Score</b>	<b>20</b>

## **2. Lab Work**

Lab work is an activity by which the concepts acquired and observations noted are practically implemented in the lab thereby more clarity about the concepts and operational skills are achieved. The students are also convinced about the use of computer for problem solving with the help of user developed programs. This activity improves the students' ability to utilize computer for developing applications. The procedure of lab work is already explained the chapter 3 of this book. It should be ensured that all the students are involved in this activity and they are awarded scores for CE.

Programming in C++ and table manipulation with SQL statements can be given for labwork. The Record Book may be the same as that was used for the first year or a new one, but it should contain a minimum of 15 C++ programs based on the standard XII syllabus, out of which 10 programs should use class data type and 5 exercises in SQL each with a table creation, record insertion and five other queries.

***Evaluation Indicators:***

1. Preparation for the work (understanding the steps / procedure / statements)	4/3/2/1
2. Skills in using tools / facilities / features of the language	4/3/2/1
3. Accuracy in using syntax and specificity in variables	4/3/2/1
4. Format and accuracy of output	4/3/2/1
5. Recording of the work and timely submission	4/3/2/1
<b>Total Score</b>	<b>20</b>

## **3. Class Test**

Class Test is considered as a class-room activity in which the students have to play a vital role from the beginning till the end. This is not simply a written test performed by using questions previously set by the teacher. In the changed class room situation, a class test can be conducted based on the following guidelines:

- ◆ After the completion of a unit, learners are assigned the task of framing multi level questions and their scoring indicators. Samples may be developed in the class.
- ◆ Questions brought by the learners may be modified by forming informal groups.
- ◆ Question bank thus developed by the learners is submitted to the teacher.
- ◆ Teacher selects or modifies or supplies questions based on selected COs for the test.
- ◆ Teacher conducts the test and discusses the scoring indicators by ensuring the participation of learners. Scoring indicators may be written on the board (or in a chart).
- ◆ Peer evaluation must be adopted for giving scorings. Teacher should interfere wherever necessary.
- ◆ Execution of the test, discussion of scoring indicators and Peer evaluation should not exceed one period.
- ◆ The class tests may be conducted for a maximum of 10 scores. The scores obtained by the students are converted into **20**.
- ◆ In each term after conducting one or two class tests, the teacher and learner together decided in advance the unit from which the class test to be conducted for the purpose of recording.
- ◆ The average of scores obtained in the class tests conducted in two terms will be the final CE score for the item.

#### **4. Project**

In Computer Science, a software development is regarded as a project. As in the case of any project, a software development is a process oriented activity, in which there are different stages to be evaluated and a product in tangible as well as intangible form is evolved. It is a group activity through which a software for an application is developed using C++ language.

The project begins with identifying the application area and the associated problems. The data involved in the processing and the information to be attained are recognised and procedures or steps are derived. The students explore possibilities of co-operative and collaborative learning and apply their problem solving skills using C++ language.

Once the group is formed, it is intimated to the teacher and initial planning is done in consultation with the teacher. The students are constantly in touch with the teacher throughout the stages of project work. A project diary is to be maintained by each group and the activities are to be recorded, which will be helpful in preparing project report.

After designing and testing the software, it should be verified by the teacher and preparation of project report with the specified format is begun. Each of the group member should have a

copy of the report and one copy should be submitted to the teacher which will be kept in the library for reference. While setting the CE, the teacher may conduct a viva voce to ensure the involvement of the student in this activity.

### ***Project Guidelines***

1. A group consisting of five students can be chosen for a single project (such ten groups)
2. The project development has to be evaluated by the teacher from time to time.
3. Separate reports have to be presented by each member of the group.
4. Report may be either in printed form or in hand-written form.
5. It can be attached either along with the Lab Diary or as a separate book, but the teacher-in-charge should certify it.
6. This report has to be produced for verification during practical examination (otherwise teacher can take it for granted that the candidate has not done it).

### ***Evaluation Indicators:***

1. Problem Study & Planning (Problem definition, time scheduling, division of responsibilities)	4/3/2/1
2. Input, Output, and file design (Specifying the screen layouts for input /output design, file designing, if necessary)	4/3/2/1
3. Process design (Developing logic, efficiency in language/ tools for coding, debugging capabilities, format and accuracy of output)	4/3/2/1
4. Ability to prepare project report (Reflection of processes, Communicability and authenticity of the report)	4/3/2/1
5. Viva Voce (Knowledge of the content and process)	4/3/2/1
<b>Total Score</b>	<b>20</b>

### ***Structure of Project Report***

1. Cover Page (Project Title, Name of Students, Course and Duration)
2. Certificate
3. Preface/Abstract
3. Content  
Introduction

Aims (Objectives)  
Problem Study/Analysis  
Development Tools and Facilities used  
Source Code/Procedure/Steps  
Ouputs  
Conclusion

4. Annexure (Sample data, data sheet etc.)
5. Bibliography

### Sample Projects

1. Student information system
2. Library information system
3. Employee information system
4. Help line for Computer fundamentals
5. Admission software.
6. Electricity Bill
7. Salt Analysis
8. Periodic table
9. Quiz
10. Help line for OOP concepts
11. Simple games
12. Calculating the total mark and the rank
13. Calculator
14. Implementation of any problem related to Physics, Chemistry or Mathematics.

### Evaluation Indicators of other CE Items

#### Collection

- |                                     |           |
|-------------------------------------|-----------|
| 1. Comprehensiveness                | 4/3/2/1   |
| 2. Relevance to the topic           | 4/3/2/1   |
| 3. Presentation                     | 4/3/2/1   |
| 4. Awareness of the content         | 4/3/2/1   |
| 5. Timebound completion of the work | 4/3/2/1   |
| <b>Total Score</b>                  | <b>20</b> |

**Seminar**

1. Planning and organisation  
(time, topics, sources of data, method of presentation) 4/3/2/1
  2. Data collection  
(relevance, authenticity, variety of sources etc.) 4/3/2/1
  3. Preparation of report (sequence in the presentation of the concept,  
(authenticity and clarity of ideas/views/concepts) 4/3/2/1
  4. Awareness of the content (presentation of paper, participation  
in discussion, ability to substantiate own ideas/views) 4/3/2/1
  5. Skill in communication (presentation of paper, participation  
on discussion) 4/3/2/1
- Total Score 20**

**Debate**

1. Participation at different stages 4/3/2/1
  2. Democratic approach 4/3/2/1
  3. Relevance and logic of the facts and concepts presented 4/3/2/1
  4. Communication skill 4/3/2/1
  5. Preparation of report 4/3/2/1
- Total Score 20**

***Processing of CE Scores***

Each CE item is evaluated to a score with a maximum of 20 using the given evaluation indicators. If the performance of the students is to be disclosed, it should not be in numeric figure, rather it should be converted into grade as follows:

<b>Scores</b>	<b>Grade</b>
16 - 20	A
12 - 15	B
08 - 11	C
04 - 07	D
Below 4	E

When CE is submitted for publishing the result, since there are 4 CE items for Computer Science, the total score is calculated out of 80. This total score is divided by 4 and is fixed as CE score.

For example, if a student gets 16 marks in lab work, 17 marks in seminar, 15 marks in assignment and 15 marks in class tests, the total is 63 out of 80. The CE is  $63/4 = 15.75$  and is rounded to 16.

---

## Practical Evaluation

Practical Evaluation (PE) is to evaluate the ability of the learner to develop program using C++ language and the skill in using SQL statements to create and manipulate database. Questions are framed in such a way that it should be capable of evaluating the learner's problem solving skills. This would certainly be able to express the competency of the learner in C++ and SQL to solve a problem by using the best programming logic. A bunch of questions have been given in the Source book (SB) covering the syllabus and each learner has to do at least 20 programs from this list as per the directions of the teacher concerned. In C++, they are supposed to do 15 questions and out of which 10 questions are based on class data type. In SQL, 5 questions are to be done. These questions and their related algorithm, program code and all such details have to be written in the Record Book. This book has to be certified by the teacher concerned before submitting for the Practical Examination, which will be conducted by the external examiner during the month of February every year. The duration of the Practical Examination is 3 hours.

During Practical Examination the examiner prepares a list of questions and each student be given a question at random to write the program code for solving the problem. The source code/statements, which are written by the student, are to be verified and if it satisfies the logic, allow the student to do it in the computer. Once that process is completed, the examiner checks the correctness and scores are awarded as per indicators given below. A maximum of 17 scores can be awarded to a C++ program or to an SQL statement and 3 scores for the viva. Total score is 20. Project is compulsory from this year onwards and viva questions can be based on project.

### *Evaluation Indicators:*

1. Use of Proper Data types/Commands	: 4
2. Correctness of Logic/ Syntax of Commands	: 10
3. Accuracy/ Perfection of output	: 2
4. Viva (1 x 2 questions)	: 2
5. Record Book	: 2
<b>Total Scores</b>	<b>: 20</b>

### **Chapters related to PE**

Chapters 1 to 8 and 12 given in second year syllabus of Computer Science.

### **Suggested Practical Questions**

1. Program to enter a 3-digit number and display it in words.
2. Program to check whether the given number is prime or not.
3. Program to find whether the given number is in palindrome or not.

4. Program to find whether the given string is in palindrome or not.
5. Program to find the largest of n numbers by using an array.
6. Program to find the smallest of n numbers by using an array.
7. Program to print the binary equivalent of a decimal number.
8. Program to print the decimal equivalent of a binary number.
9. Program to exchange value of two variables using pointers.
10. Program to print ten elements of an array in the reverse order.
11. Program to check whether the given number is even or odd or prime. The program should continue as long as the user wants.
12. Program to sort the given N numbers in the ascending order.
13. Program to sort the given N numbers in the descending order.
14. Program to print the sum of natural numbers up to 100.
15. Program to create and display a matrix.
16. Program to find the sum of all elements of a Matrix.
17. Program to print the sum of the digits of a number.
18. Program to find the value of  $C(n,r)$  using a function to find the factorial of a number.
19. Program to convert diagonal elements of square matrix into zero.
20. Program to check whether the number is Armstrong's Number or not.

The following programs have to be done using **class data type**

21. Program to add two matrices.
22. Program to subtract one matrix from another.
23. Program to find the product of two matrices, if they are conformable to do so.
24. To create a file to store the details of some students such as Roll No., Name and Marks of 6 subjects. Also read this file to prepare the result and store the result in another file.
25. Program to check whether a particular character is present in a string.
26. Program to merge two integer arrays after creating them in ascending order. The resultant array should be in a sorted order.
27. Class for telephone number and name of subscriber. Store the details of 10 subscribers and identify the phone number when name is entered.
28. Student class with data members roll no., name, marks of six subjects (int array), and member functions to: (i) Read the details of a student, (ii) Calculate the total marks and (iii) Display information along with total and percentage. Continue the program as long as the user wants.
29. Account class with data members: account number, name, balance amount and member functions to (i) Input values and (ii) Display information.

30. Quadratic equation class with data members a, b, c (coefficients), constructor with values 0, member functions to i. Input values ii. Calculate discriminant and iii. Find the roots and display it.
31. Complex class with data members to store the real and imaginary parts. Include member functions to read the number and to display both number and its conjugate.
32. Temperature class with data member temperature and member functions: (i) To input value (ii) Conversion from Celsius to Fahrenheit (iii) Conversion from Fahrenheit to Celsius and (iv) Display the result. Menu driven program with options for conversion.
33. Swap class with data members: a, b; and member functions to (i) Input values (ii) Swap the values and (iii) Display the values.
34. Calc class with data members: two integer numbers and has member functions to perform the operations addition, subtraction, multiplication and division.

### **MS Access/Oracle/SQL Server**

1. Create a table Student with fields

RollNo	Integer	Primary key
Name	Character(25)	
Sub1	Integer	
Sub2	Integer	
Sub3	Integer	
Total	Integer	

Write SQL Queries to

1. Insert data in the fields RollNo, Name, Sub1, Sub2, Sub3. (At least 10 Records)
2. Update field total with the sum of Sub1, Sub2 and Sub3.
3. Display highest marks in sub3
4. Display the names in ascending order
5. Display the name of the student with highest total

2. Create a table Student with fields

RollNo	Integer	Primary key
Name	Character(25)	
Batch	Character(20)	
Total	Integer	

Batch have values COMMERCE, SCIENCE or HUMANITIES

Write SQL Queries to

1. Insert data in the fields RollNo, Name, Batch, Total (at least 10 records)
2. Display a list of students having mark < 500
3. Display the number of students in each batch
4. Display the highest and lowest mark in SCIENCE batch

3. Create a table Student with fields

RollNo	Integer	Primary key
Name	Character(25)	
Batch	Character(20)	
Total	Integer	

Batch have values COMMERCE, SCIENCE or HUMANITIES

Write the SQL Queries to

1. Insert data in the fields RollNo, Name, Batch, Total. (at least 10 records)
2. Display the Names of all students in SCIENCE batch.
3. Display the highest mark in the table.
4. Display the name of students in the descending order of total mark
5. Display the details of students with RolNo from 3 to 7

4. Create a table Employee with fields

EmpNo	Integer	Primary key
Name	Character(25)	
Designation	Character(20)	
Department	Character(25)	
BasicPay	Number(8,2)	
DA	Number(8,2)	
Gross	Number(8,2)	

Write SQL queries to

1. Insert data in the fields EmpNo, Name, Designation, Department and BasicPay. (at least 10 records)
2. Update DA in the table as 40% of BasicPay
3. Update Gross as the sum of BasicPay and DA
4. Display the EmpNo, Name of the employee with minimum Gross
5. Display the details of the employee with maximum Gross

5. Create a table Employee with fields

EmpNo	Integer	Primary key
Name	Character(25)	

Designation	Character(20)
Department	Character(25)
BasicPay	Number(8,2)
DA	Number(8,2)
Gross	Number(8,2)

Write SQL Queries to

1. Insert data in the fields EmpNo, Name, Designation, Department and BasicPay, DA. (at least 10 records)
2. Update Gross as the sum of BasicPay and DA
3. Display the Department wise listing of all employees in ascending order of name.
4. Display the total Gross salary paid to salesmen
5. Display the details of the employees arranged in the order of designation

6. Create a table Employee with fields

EmpNo	Integer	Primary key
Name	Character(25)	
Designation	Character(20)	
Department	Character(25)	
BasicPay	Number(8,2)	
DA	Number(8,2)	
Gross	Number(8,2)	

Write SQL Queries to

1. Insert data in the fields EmpNo, Name, Designation, Department and BasicPay, DA. (at least 10 records)
2. Update Gross as the sum of BasicPay and DA
3. Display total salary paid in each Department
4. Display average salary of Managers
5. Display the details of the employee with maximum salary

7. Create a table Deposit with fields

AccNo	Integer	Primary key
Name	Character(25)	
Branch	Character(20)	
Amount	Number(8,2)	Not Null

Branches can be CALICUT, COCHIN or TRICHUR.

Write SQL queries to

1. Insert data in all the fields (at least 10 records)
2. Display the total deposit in each branch
3. Display the accounts branch wise in ascending order of AccNo.
4. Display the details of deposit with maximum amount
5. Display the AccNo and Name of deposits in the branch CALICUT.

8. Create a table Customer with fields

AccNo	Integer	Primary key
Name	Character(25)	
Branch	Character(20)	
Amount	Number(8,2)	Not Null

Branches can be CALICUT, COCHIN or TRICHUR.

Write SQL Queries to

1. Insert data in all the fields (at least 10 records)
2. Display the AccNo and Name of Customers
3. Display the details of the Customer who has deposited exactly Rs. 10000
4. Display the account no.'s of account holders who have deposited more than Rs. 10000 and less than Rs. 50000.
5. Display the names and AccNo. of depositors in Calicut branch who have more than 20000 as deposit.

9. Create a table Employee with fields

EmpNo	Integer	Primary key
Name	Character(25)	
Designation	Character(20)	
Department	Character(25)	
BasicPay	Number(8,2)	
DA	Number(8,2)	
Gross	Number(8,2)	

Write SQL Queries to

1. Insert data in all the fields. (at least 10 records)
2. Display the name of all Departments without repetition
3. Update DA as 50% of the BasicPay
4. Update Gross as the sum of BasicPay and DA
5. Display the details of employees with gross between Rs.5000 and Rs.10000

10. Create a table Customer with fields

AccNo	Integer	Primary key
Name	Character(25)	
Branch	Character(20)	
Amount	Number(8,2)	Not Null

Branches can be CALICUT, COCHIN or TRICHUR.

Write SQL Queries to

1. Insert data in all the fields (at least 10 records)
2. Display the details of Customers from AccNo 1 to 5
3. Display the list of Customers with Amount less than 100
4. Display the Name of Customers with maximum Amount in each Branch
5. Display the details of Customers in a Branch other than CALICUT

## Terminal Evaluation

The TE means that it is the written examination conducted by the Directorate of Higher Secondary Education, Kerala at the end of the academic year of standard XII. There will be questions for a maximum of 60 scores and the maximum time for writing examination is 2½ hours. The candidates will be provided with 15 minutes cool off time get acquainted with the question paper to plan themselves for giving their responses. The test items will be based on the selected COs of the subject. The weightage to the CO/content and the weightage to type of questions will be fixed by the question setters. There may be questions for 1 score, 2 scores, 3 scores and 5 scores, but there will not be separate sections for the type of questions. Most of the questions may have sub sections numbered as (a), (b) and (c). The questions having such subdivisions will mostly based on a CO or a cluster of related COs. Out of these sub divisions, the first question will be an easy question, the second will be an average difficulty level question and third one will be comparatively a difficult question.

All questions may need to be answered, but some internal choices may be allowed. Still, the score for choice questions will not exceed 25% of the total score. The questions with 1 score may not always be multiple choice, but an answer that requires a word, phrase or a sentence. Most of the questions will be thematic and hence the students should be provided with such learning experiences. The important feature of the test items is that the questions are process oriented. That is, the higher order mental processes will be the target of the questions. Let us see the mental process from lower to higher order.

- 1 retrieve/recollects/retells information
- 2 readily makes connections to new information based on past experiences and formulates initial ideas/concepts.
- 3 detects similarities and differences.
- 4 classifies/ categorises/ organises information appropriately.
- 5 translates/ transfers knowledge or understanding and applies them in new situations.
- 6 establishes cause-effect relationships.
- 7 makes connections/relates prior knowledge to new information/applies reasoning and draw inferences.
- 8 communicates knowledge/understanding through different media.
- 9 imagines/ fantasises/ designs/ predicts based on received information.
- 10 judges/appraises/evaluates the merits or demerits of an idea/develops own solutions to a problem.

### **Grading**

Consider an achievement test of total score 100. There is a possibility of getting 0 score to 100 score for individuals. In other words, we can say that this is an 101 point scale. It is very difficult to distinguish a learner scored 89 with another learner who scored 90. There may not be any difference in the abilities of these two learners. To overcome such limitations a popular mode of evaluating students performance known as grading system has been evolved. It is used all over the world. At the higher secondary stage it is desirable to use nine point absolute grading to coordinate and record the evaluation. In Computer Science, the scores obtained in CE out of 20, PE out of 20 and TE out of 60 are added and the total score is converted into percentage and appropriate letter grades are awarded corresponding to each score. This system is termed as absolute grading. The score percentage and corresponding letter grades are given below:

<b>Score in Percentage</b>	<b>Grade</b>
90 - 100	A+
80 - 89	A
70 - 79	B+
60 - 69	B
50 - 59	C+
40 - 49	C
30 - 39	D+
20 - 29	D
Below 20	E

Part II

Chapter wise  
Detailing of  
Learning Activities



**1**

# **Review of C++**



## UNIT PLAN

<b>Curriculum Objectives</b>	<b>Concepts/Ideas</b>	<b>Process Skills</b>	<b>Activities</b>	<b>Learning Aids/ Materials</b>	<b>Evaluation</b>	<b>Time (Periods)</b>
Familiarise the concepts of basic data types, operators and control structures	Data types, variables, Operators, Expressions, Program structure, Selection constructs, Loop constructs and Jump statements	Observation, Classification, Communication and understanding, Inference, Prediction, Making operational definition	Discussion, Observation, Quiz, Problem solving, Coding, Code conversion, Error correction, Lab work	Text Book, Edumate - XI, Record Book	Record Book, Notes containing program code, Output for a code segment, Error corrections etc.	12
Familiarise the concepts of arrays and functions	Array declaration and initialisation, array operations, types of arrays, functions, role of arguments, type of arguments and type of function calls	Observation, Classification, communicating and understanding of communication of others, inference, prediction and making operational definition	Discussion, Observation, Coding, Problem solving, Error correction, Output prediction, Lab work	Text Book, Edumate - XI, Record Book	Record Book, Notes containing program code, Output for a code segment, Error corrections etc.	8

## Introduction

This module explores the advanced aspects of C++ language. The elementary concepts of C++ are already covered in the previous year. This chapter provides a quick review of basic features of C++.

Concepts such as data types, operators, control structures, arrays and functions are revised through discussions and illustrations.

## Curriculum Objectives

1. Familiarise the concepts about basic data types, operators and control structures in C++ through quiz, discussion, error correction, illustrations, lab work etc. and develop programs.
2. Develop the idea about arrays and functions through discussion and develop programs for array operations.

## Content Details

Basic Data types; Operators; Control Structures; Arrays and Functions.

## Pre requisites

- Knowledge about data types, operators, control structures, arrays and functions in C++
- Experience in coding and testing of basic C++ programs

## Learning Activities

### 1.1 Group Discussion to identify the tokens in a C++ program (40 Mins.)

1.1.1 Problem: *Identify and categorise the tokens in a C++ program*

1.1.2 Students are divided into five groups

1.1.3 Each group is provided with a program code

1.1.4 The groups prepare a categorised list of tokens such as data types, operators, control structures etc. in the program and present their findings

1.1.5 An assignment is given to prepare a list of tokens in different categories along with examples

### 1.2 Quiz on basic concepts of C++ (60 Mins.)

1.2.1 Students are divided into groups of four to five members

1.2.2 Each group prepares a set of questions based on fundamentals of C++ already studied in the previous year.

1.2.3 Team quiz is conducted

1.2.4 If there is some concept that is not answered clearly by the majority, it may be revised through some activity

**1.3 Discussion on basic data types in C++ (30 Mins.)**

- 1.3.1 Problem: *Write a program to find the average of two numbers given as input*
- 1.3.2 Students write the program individually and list out the variables along with the data type used in the program
- 1.3.3 A discussion is conducted to point out the following aspects.
  - (a) Suitability of float data type for the variable to store average
  - (b) Change in the output of the program when the data type of variable to store average is int and float.Discussion may be carried out based on three or more sample inputs
- 1.3.4 After the consolidation of the discussion, students prepare a list of data types and type modifiers, along with a sample data item that can be properly represented through the data type.
- 1.3.5 Prepare a chart showing basic data types and size

**1.4 Discussion on operators in C++ (30 Mins.)**

- 1.4.1 Students list out the operators used in the program given for activity 1.1
- 1.4.2 Discuss the need of parenthesis in the expression to find the average and point out the role of operator precedence in evaluating the expressions
- 1.4.3 Discuss the changes needed in the program to check whether the average is greater than a value given as input
- 1.4.4 Through a discussion on different types of operators, prepare a chart classifying the operators according to the type. Arrange the operators in the order of precedence.
- 1.4.5 Select some interesting data processing operations. Conduct a group discussion to implement the operation with a minimum number of operators and variables. (eg: Swapping of values stored in two integer variables)

**1.5 Discussion on the selection statements in C++ (40 Mins.)**

- 1.5.1 Problem: *Write a program to display the grade corresponding to the mark given as input (using if - else ladder)*
- 1.5.2 Students write the program individually
- 1.5.3 Discuss the changes needed in the program to replace the if else ladder with the switch case statement. One or two students illustrate the syntax of if else and switch case statements
- 1.5.4 Give assignment to write the program using switch case statement
- 1.5.5 Students prepare a chart showing syntax of selection statements

**1.6 Discussion on loop statements (40 Mins.)**

1.6.1 Problem: *Write a program to check whether a number given as input is prime or not*

1.6.2 Students are divided into five groups

1.6.3 Implementation of the program using one loop is given to two groups

1.6.4 Each group presents their program

1.6.5 Give assignment to implement the program using all the loops

**1.7 Discussion on the concept of arrays (30 Mins.)**

1.7.1 Problem: *Identify the use of arrays in a C++ program*

1.7.2 A discussion is conducted on following points

- Need of arrays
- Syntax for declaring an array
- Accessing of array elements

1.7.3 Students are divided into five groups and each group is provided with a program that may be implemented only using array

1.7.4 Each group presents their program and others comment on the program

1.7.5 Give assignment to implement a set of programs that need arrays

**1.8 Discussion on concept of functions (40 Mins.)**

1.8.1 Problem: *Write a program to output  $nCr$  using function to find factorial*

1.8.2 Students write the program individually

1.8.3 Discuss the merit of using the factorial function in the program

1.8.4 Students list out the advantages of functions. Also point out the steps to implement user defined functions in a program

1.8.5 Prepare a chart showing commonly needed library functions classified in the order of corresponding header file

**1.9 Test on programming skills (40 Mins.)**

1.9.1 A test is conducted with questions from the following areas

- (a) Predict the output of a program/code segment for sample inputs
- (b) Correct the error in a code segment
- (c) Write the code segment to work out a logic

1.9.2 After conducting the test, answer for the questions is discussed. Performance in the test may be evaluated through peer evaluation

**1.10 Brainstorming on code optimisation techniques (60 Mins.)**

- 1.10.1 Students are divided into smaller groups
- 1.10.2 The groups are provided with a program for which they have to prepare optimised code. Interesting problems that can be implemented in different methods may be selected for this purpose. The following are some examples.
- Program to find factorial of a number without using running variable stepping from 1 to N
  - Function to swap values of two integer variables without using temporary variable to swap values
- 1.10.3 Each group presents the code they have worked out
- 1.10.4 Efficiency of the program may be evaluated and select the best one based on the following points
- Minimum number of variables
  - Expressions with minimum operators
  - Lesser code size and faster output
- 1.10.5 Based on the above criteria, the programs may be evaluated

**Sample TE Questions**

- Examine the code segment given below
 

```
for(int i=0, f=1; i<=3; i++)
    f*=i;
cout<<f;
```

  - Write the output of the code segment.
  - Rewrite the code segment using while loop
  - Discuss the changes needed in the code segment to get the factorial of 7

**Scoring Indicators:**

- Output is 0, as the variable f is initialised to 0 1 score
  - While loop with correct syntax 1 score
  - The variable i must be initialised with value 1 and change the condition to i <= 7 1 score
- Identify the errors and its type in the following code segment. Also correct the errors
    - `cout>>x;`
    - `for(int i=0; i<10; i++)`  
`{ cout<<x/i; }`
    - `(y<x)? cout<<y<<"is greater" : cout<<x<<" is greater";`

**Scoring Indicators:**

- Syntax error : Output operator is wrong 1 score
- Run time error : `x/i` results in error when value of i is 0 1 score
- Logical error : Condition output the lesser value 1 score

3. Write the output of the following code segment when i is an integer variable with value 10

```
int x, i=10;
```

```
(a) i += i + i ; cout<<i;
```

```
(b) x= i ++ + ++ i ; cout<<x;
```

**Scoring Indicators:**

(a) 30 1 score (b) 22 1 score

4. Write the syntax of declaring variables to store roll number of all students in a class of strength 50. Also assign values 1 to 50 to the variables

**Scoring Indicators:**

(a) Declare an int array of size 50 1 score

(b) Through a loop, assign values to the array elements 1 score

5. Is it possible to call a user defined function without declaring its prototype? Illustrate your argument through an example

**Scoring Indicators:**

Define the function before the calling statement. 1 score

Give a sample function definition and call 2 scores

6. Define a C++ function to (a) find factorial (b) increment value of an integer variable passed as argument.

Elicit the advantages and disadvantages of function, considering the above functions

**Scoring Indicators:**

Function definition 3 scores

Advantage and disadvantage 2 scores

7. Discuss the method used in C++ for storing string data.

Write a C++ function that takes a string and a character as argument and checks whether the character is present in the string

**Scoring Indicators:**

(a) Concept of character array 2 scores

(b) Function definition with correct syntax 1½ scores

(c) Correct logic 1½ scores

8. Match the following

<b>A</b>	<b>B</b>
for	user defined function
if else	iostream.h
main	selection statement
cout	basic data type
array	iteration statement
int	derived data type

**Scoring Indicators:**

½ score for each correct match

½ x 6 = 3 scores

**2**

# Structures



## UNIT PLAN

Curriculum Objectives	Concepts/Contents	Process Skills	Learning Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Identify the needs and relevance of user defined data type.	User-defined data types, Grouped data items.	Classification, Communication, Listing.	Problem Solving Discussion	Text Book, Notes	Classified list of data items.	1
Familiarize the concept of structure	Structure declaration and definition, Initialisation of structure variables, Accessing <u>structure members</u> .	Inference, Definition, Communication.	Illustration Problem Solving, Discussion, Lab work	Text Book Record Book	Declarations & definitions of various kinds of structures	2
Understand need and use of array of structures	Connecting arrays with structures, Declaration and accessing of elements.	Inference, Prediction, Definition, Comparison.	Illustration, Discussion, Error Correction, Output Prediction, Table Preparation, Lab work	Text Book, Notes, Record Book	Correct versions of codes, Correct outputs, Record Book	2
Familiarize the concept of nested structure	Nesting of structures, Accessing the members, Different types of nestings	Inference, Definition, Application	Problem Solving, Error Correction, Output Prediction, Program coding,	Text Book, Notes, Record Book	Definitions, Corrected versions of code segments.	2
Understand different methods for implementing symbolic constants	Symbolic constants, Enumerated data types, #define statement	Communication, Inference, Prediction, Definition, Application	Problem Solving, Illustrations, Error Correction, Output Prediction, Coding, Lab work	Text Book, Notes, Record Book	Correct versions of code segments, Correct outputs, Record Book	1

## **Introduction**

The students are now familiarised with the basic concepts of programming in C++ language. The ability to apply the elementary features or facilities of C++ to develop programs is also verified in the last chapter. In this chapter, let us make the students aware of the need for user-defined data types and familiarise them with the simplest user-defined data types like structures and enumerators. The relevance and use of the structure data type in programs are to be brought to the attention of the student as they are approaching the core concepts of C++ in the form of classes and objects. The concepts included in this chapter may be transacted through the learning strategies such as discussion, illustration, program development, lab work etc. Also, we can utilise activities like error correction and program development for evaluation.

## **Curriculum Objectives**

1. Identify the needs and relevance of user defined data type through discussion and problem solving.
2. Familiarize the concept of structure through discussion, problem solving etc. and develop programmes using structures.
3. Understand need and use of array of structures through problem solving and prepare notes comparing array and structures.
4. Familiarize the concept of nested structure through discussion and problem solving.
5. Understand different methods for implementing symbolic constants through illustration and discussions and prepare a comparison chart.

## **Content Details**

User defined Data types; Structure – Definition, Creation of structure variable, Referencing structure elements; Array of structures; Nested structures; Use of typedef; Enumeration – Defining enum type, declaration, changing default ordinal values; Symbolic Constants – const, enum, #define.

## **Pre requisites**

- Idea about basic data types of C++.
- Use of variables and memory allocation for them.
- Concept of arrays and accessing its members.

## Learning Activities

### 2.1 Discussion to identify the need of user-defined data types (40 Mins.)

- 2.1.1 Data items like name, age, sex, date of birth, address etc. are written on the board and the students are asked to classify them as elementary or atomic data items and grouped data items.
- 2.1.2 Students individually classify them and one or two classified lists are presented.
- 2.1.3 The students are asked to write individually the data types of C++ for each of the data items listed.
- 2.1.4 They consolidate the data through groups (bench-wise) discussion and present the items with the corresponding data types.
- 2.1.5 The difficulty to specify the data types for grouped data items is brought to the attention of the students and the need for user-defined data types is introduced.
- 2.1.6 The discussion is consolidated by pointing out the need and relevance of user-defined data types in programming.

### 2.2 Illustration to introduce the structure data type (40 Mins.)

- 2.2.1 One of the grouped data items (say date of birth) is taken for discussion.
- 2.2.2 Students are asked to decompose or split this data item into different components. (Such as day number, month number or name and year)
- 2.2.3 The students are asked to write variable declaration statements for each of these items.
- 2.2.4 Correct statements are gathered and grouped them by putting braces and giving a name using the keyword “struct”.
- 2.2.5 The students try to frame the syntax of structure definition and present it. The teacher gives support if required and consolidate the discussion by setting the correct syntax.
- 2.2.6 The teacher assigns the students to write structure definition for various data items and evaluate in peer.

### 2.3 Discussion to familiarise the method of accessing structure elements (40 Mins)

- 2.3.1 One of the assigned work is taken for discussion by writing its structure declaration.
- 2.3.2 Relevance of structure variable is introduced and the memory allocation for the variable is elicited through brainstorming.
- 2.3.3 Layout of the memory allocation for the variable is demonstrated through diagram.
- 2.3.4 The students may be asked to write statements for inputting values to the variable.
- 2.3.5 Let them recognise the difficulty in referring the elements of the structure variable. The teacher introduces the concept of dot operator.

2.3.6 The teacher demonstrates the usage of dot operator and asks the students to complete their task of inputting values.

2.3.7 Exercises may be given in the form of writing simple programs to evaluate the students.

2.3.8 A home assignment may be given to prepare notes on structures covering all the concepts transacted so far.

**2.4 Discussion to identify the need of using array of structure (90 Mins.)**

2.4.1 The students may be asked to write code segment to input 100 integers to recollect the concept of arrays and accessing its elements.

2.4.2 Correct code segments may be written on the board after presentation.

2.4.3 The teacher gives a problem that requires array of structure and asks the students to write necessary code segment to solve this.

2.4.4 The learners consolidate the code segment in groups and one or two programs may be presented.

2.4.5 Through open discussion, correct the code, if required.

2.4.6 The students are asked to complete the program in the Observation Book at home and do it in the lab.

2.4.7 More problems are given that require array of structures, and arrays within the structures.

2.4.8 The students are asked to prepare a table in their notes by comparing arrays and structures.

2.4.9 Notes may be evaluated in peer groups and programs may be verified as part of lab work.

**2.5 Discussion to familiarise nested structures (40 Mins.)**

2.5.1 Students are asked to define a structure to represent the details of a student consisting of admission number, name, date of birth and address.

2.5.2 Difficulties to include grouped data items (like date of birth and address) are elicited and the concept of nesting of structures is introduced.

2.5.3 The students complete their task and present the structure definition.

2.5.4 Exercises for accessing the members of the structure are given.

2.5.5 Students complete their work individually and consolidate in groups and present them.

2.5.6 Problems may be given to write programs using nested structures and be evaluated in peer.

**2.6 Discussion to familiarise enumerated data type (40 Mins.)**

2.6.1 The set of day names of a week (sun, mon, ..... sat), set of boolean values (true, false), set of nine point scale grades (A+, A, B+, ....., E) are written on the board and the student are asked to find out the suitable numerical values which can be associated with the members of each list.

- 2.6.2 The concept of enumerators is introduced by considering the reverse case (i.e., fix the numbers first and ask the students to assign the numbers with suitable names).
- 2.6.3 By providing examples, the students can be asked to develop a definition for enumerated data type.
- 2.6.4 The teacher then demonstrates the syntax of enumerated data type.
- 2.6.5 The learners are given more exercises for designing enumerated data type.
- 2.6.6 With suitable code segments that consist of enumerated data type definition, variable declaration, assigning values to the variable, using the enumerators in condition checking, etc. the concept of ordinal values is familiarised.
- 2.6.7 The students are asked to prepare notes on the advantages of enumerators in programming.

**2.7 Discussion to familiarise different types of symbolic constants (40 Mins.)**

- 2.7.1 A variable initialisation statement is provided to the students and the possibility of changing the contents of the variable is discussed.
- 2.7.2 The statement is modified by providing the keyword `const` just before the data type.
- 2.7.3 Its impact on changing the content of the variable is discussed.
- 2.7.4 The difference is highlighted by introducing the concept of symbolic constant.
- 2.7.5 The concept of enumerators is brought to the attention of the students and the names assigned with numbers are introduced as symbolic constants.
- 2.7.6 The formation of symbolic constants in the above two cases are discussed in terms of time of creation and memory allocation.
- 2.7.7 As an alternative method, `#define` is introduced and illustrated.
- 2.7.8 The learners are asked to prepare notes by comparing the three methods of defining symbolic constants.

**Sample TE Questions**

1. A program is to be written to process the details of employees in an organisation. The data of an employee consists of employee code, name, date of joining, department, designation and basic pay.
  - (a) Name the data type by which the above data can be represented together. (1 score)
  - (b) Give definition for the data type you suggested (3 scores)
  - (c) From the above definition, write statement(s) to input the date of joining of an employee. (1 score)

**Scoring Indicators:**

- |  |           |
|--|-----------|
| (a) Naming the data type as structure                  | 1 score   |
| (b) (i) Using the keyword struct                       | 1/2 score |
| (ii) Tag specification                                 | 1/2 score |
| (iii) Defining nested structure                        | 1 score   |
| (iv) Declaration of other variables                    | 1 score   |
| (c) For accessing the elements of the nested structure | 1 score   |
2. Observe the structure definition given below:

```
struct Outer
{
    int num[10];
    char str[20];
    struct Inner
        {short var;
          double dup;
        }s1;
    }s2;
```

Read the following C++ statements and check whether they are correct or not. If correct, give the output (or working of the statement), otherwise identify the errors and correct them.

- (a) `cin.getline(s1.str,20);`
- (b) `for (int i=0; i<10; i++) s2.num[i]=0;`
- (c) `Inner s3 = {1, 3.1414265};`
- (d) `cin>>s2.s1.var;`
- (e) `cout<<s2.num+s1.var;` (5 x 1 = 5 scores)

**Scoring Indicators:**

- |  |         |
|--|---------|
| (a) Invalid. s1 is Inner structure variable, but str is Outer element  | 1 score |
| (b) Correct. Initialises the array num of structure variable s2.   | 1 score |
| (c) Invalid. Inner variable cannot be declared outside Outer   | 1 score |
| (d) Correct. Inputs value to the element var of structure variable s1 through s2                                   | 1 score |
| (e) Invalid. The array num cannot be accessed for addition and the variable var cannot be accessed in this fashion | 1 score |

**3**

# Concepts of Object Orientation



### UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Compare procedure oriented approach and Object Oriented approach.	Programming paradigm, Types of paradigms.	Comparison between POP and OOP.	Discussion, Table preparation.	Text Book, Record Book	Notes	1
Understand the basic concepts of OOP.	Abstraction, Encapsulation, Polymorphism, Inheritance, Class and Objects.	Observation, Application	Real life exploration, Discussion.	Text Book, Record Book	Notes	3

## Introduction

In today's highly competitive and changing world, the complexity of software is also increasing. These large complex programs are more prone to errors, and debugging these errors is expensive and time consuming. Traditional languages are not powerful enough to cope with this complexity. They failed to produce the desired results in terms of error free, easy to maintain and reusable programs. The new programming approach, Object Oriented Programming, offers a new and powerful method to develop programs which are more reliable, clear and easily maintainable. Here we introduce the Object oriented programming concepts such as data abstraction, encapsulation, polymorphism, modularity and inheritance through discussions.

## Curriculum Objectives

1. Compare Procedural Oriented Approach and Object Oriented Programming through discussion and prepare a comparison table.
2. Understand the basic concepts of OOP through illustration of real life problem and prepare notes.

## Content Details

Procedural V/s Object Oriented Programming; Concepts of OOP – Abstraction, Encapsulation, Modularity, Inheritance, Polymorphism.

## Pre requisites

- Knowledge about any traditional programming language.

## Learning Activities

### 3.1 Discussion on Procedural V/s Object Oriented Programming (40 Mins.)

- 3.1.1 Problem: *Write down the time schedule for a tour of 3 days for covering three or more tourist places.*
- 3.1.2 Students discuss within groups and present time shedule of the tour which may probably contain the places, route and mode of conveyance in the order of days.
- 3.1.3 Teacher raises questions for forcing the students to modify the schedule so that they give more importance to places and dates for visit rather than the route.
- 3.1.4 The merits of the modified time schedule is highlighted and the importance of data such as date and places rather than procedure for the tour (rout) is brought to the attention of the students.
- 3.1.5 Teacher introduces the terminologies such as OOP and POP. Students prepare a comparison table for these two programming paradigms.

**3.2 Discussion on data abstraction (20 Mins.)**

3.2.1 Problem: *What are the essential features of a vehicle that must be known to a person for driving it?*

3.2.2 Through a general discussion, students list down the features and present them.

3.2.3 Teacher also asks to list the features that are not essential for a person to drive the vehicle.

3.2.4 Through a general discussion, students list down the features and present them.

3.2.5 Teacher consolidates the concepts of data abstraction with the help of above example and students write notes on it.

**3.3 Discussion on data encapsulation (20 Mins.)**

3.3.1 Problem: *What can be the difficulties, if the files of all departments are stored in a single room and employees of all departments sit in another room?*

3.3.2 Through a general discussion, the students list out the possible difficulties.

3.3.3 Teacher randomly chooses some students and asks them to suggest some method to overcome the above problem.

3.3.4 Teacher consolidates the solutions put forward by the students as the one given below:

The files associated with each department and the employees who handle the those files are to be placed in the same room.

3.3.5 Using this analogy, teacher introduces data encapsulation and students prepare notes.

**3.4 Discussion on modularity (20 Mins.)**

3.4.1 Problem: *What are the difficulties in conducting a youth festival?*

3.4.2 Through a group discussion each group lists the problems in conducting youth festival.

3.4.3 Each group presents its difficulties

3.4.4 Teacher asks the students to suggest some method to overcome the above difficulties.

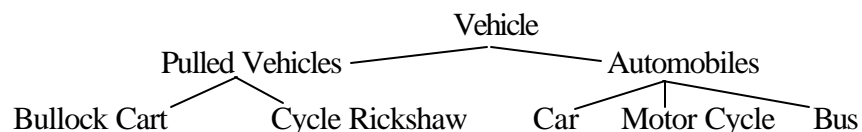
3.4.5 Through a general discussion, the students list out some methods.

(The discussion should focus on the formation of various committees for the distribution of responsibilities.)

3.4.6 Teacher consolidates and introduces the concepts of modularity and students prepare notes on it.

**3.5 Discussion on inheritance (20 Mins.)**

3.5.1 A hierarchical chart as follows may be given to initiate discussion:



- 3.5.2 Students are asked to list similar and dissimilar features of the entities in the chart.
- 3.5.3 Teacher consolidates the features to introduce the concept of inheritance.
- 3.5.4 Students are asked to prepare similar charts depicting the concept of inheritance.

### **3.6 Discussion on polymorphism (20 Mins.)**

- 3.6.1 Problem: *List out the meanings or messages conveyed by ringing the bell in a school. (Bell is a characteristic of school and ringing it is the behaviour or function performed on it)*
- 3.6.2 Through an open discussion, the responses from the students are noted on the board.
- 3.6.3 Teacher consolidates discussion and introduces the concept of polymorphism.
- 3.6.4 Students are asked to identify similar situations showing polymorphism.

### **Sample TE Questions**

1. Raju develops programs using BASIC language. Tom develops programs using the basic concepts of C++ language.
  - (a) How do you differentiate these two programs based on the approach?
  - (b) What are the features of the language used by Tom?
  - (c) Explain these features in detail.

#### **Scoring Indicators:**

- (a) Procedural Oriented Programming & Object Oriented Programming      1 score
  - (b) Data abstraction, Encapsulation, Inheritance, Polymorphism       $\frac{1}{2} \times 4 = 2$  scores
  - (c) For definitions of above features       $\frac{1}{2} \times 4 = 2$  scores
2. While doing transactions in an ATM counter the account holder can check his balance, withdraw cash, deposit etc. He is not aware of the background processes like interest calculation, transactions that takes place in bank etc.
    - (a) Name the OOP feature that you can correlate with the above situation?
    - (b) Explain the feature in detail.

#### **Scoring Indicators:**

- (a) Data abstraction      1 score
- (b) For correct explanation      2 scores

**4**

# **Classes and Objects**



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Familiarize the implementation of classes in C++	Class as a datatype, Access Labels, Creation of objects, Distinction between class and structure	Observation, Comparison, Experimentation	Discussion, Illustration, Error correction	Text Book	Notes	1
Understand different methods for defining member function and recognising the situation for each.	Different methods of member function definition, Scope resolution operator.	Observation, Experimentation, Inference, Prediction	Illustration, Problem solving, Discussion, Coding, Error correction, Output prediction, Lab work	Text Book Record Book	Notes, Record Book	2
Familiarise the concepts of friend function and inline function.	Inline function, Friend function, Bridging of classes, Friend class	Observation, Experimentation, Inference, Prediction	Problem solving, Discussion, Coding, Error correction, Output prediction, Lab work	Text Book.	Notes	2

## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Recognise the need of array of objects and familiarise with the exchanging of objects between functions.	Array of objects, Objects as arguments, Functions returning objects.	Observation, Inference, Prediction, Experimentation.	Problem solving, Discussion, Error correction, Output prediction, Coding, Lab work	Text Book, Lab Diary	Notes, Lab Diary	4
Familiarize the concepts of static members constant member functions and nesting of classes	Static data member, Static member functions, Memory allocation for static members, Constant member functions, Nested classes	Observation, Inference	Discussion, Layouting	Text Book, Lab Diary	Notes	2
Identify situations for implementing classes and develop programs.	Applications of classes	Observation, Experimentation	Problem solving, Discussion, Coding, Labwork.	Text Book, Lab Diary	Lab Diary	2

## **Introduction**

This chapter discusses the features of classes and objects. The relevance and concepts of classes in OOP paradigm is introduced. A structure provides a way to group data elements. But a class combines related data and associated member functions under a single unit. Hence a thorough knowledge of class is essential for developing object oriented programs. These concepts may be transacted through discussions, lab work, error corrections, etc.

## **Curriculum Objectives**

1. Familiarise the concepts of classes and its implementation in C++ through discussion, illustration etc. and define C++ classes.
2. Understand different methods and relevance of defining member function of classes through problem solving, discussion etc. and apply these methods for member function definition.
3. Familiarise the concepts of friend function and inline function and their impact in programming through observation, illustration, coding etc. and develop programs.
4. Identify the need for array of objects and exchanging objects between functions through discussion, illustrations etc. and develop program.
5. Familiarise the concepts of static member function, constant member function and nesting of classes through illustration, discussion, memory allocation charting etc. and prepare notes differentiating them.
6. Identify situations for implementing classes and develop programs using them.

## **Content Details**

Class declaration; Data members; Member functions; Access labels: Default labels; Data hiding and Encapsulation; Arrays within a class; Methods of member function definitions; Scope resolution operator; Inline functions; Creating Objects; Accessing members; Array of objects; Objects as function arguments - Pass by value and reference; Memory allocation of class and objects; Static members; Friend functions; Constant member functions; Nesting of classes.

## **Pre requisites**

- Knowledge of control structures and arrays
- Idea about functions and structures
- Classes and Objects - Generic concepts.

## Learning Activities

### 4.1 Discussion on Generic concepts of classes and objects (20 Mins.)

- 4.1.1 A 'class' such as person class, vehicle class, etc. is given to each group.
- 4.1.2 Teacher asks each group to select an object of that class
- 4.1.3 Students are asked to write down the characteristics and behaviour of that object.
- 4.1.4 Teacher consolidates the concept of class and objects, and the students prepare notes.

### 4.2 Discussion on the need for classes (90 Mins.)

- 4.2.1 Problem: *Write a program to handle the details such as rollno, name and scores in CE items of a student. The program should accept as input the details of students, calculate the average CE mark and then print the result using separate functions.*
- 4.2.2 Students are asked to write program individually.
- 4.2.3 One or two programs are presented.
- 4.2.4 Teacher asks the students to subdivide the program into different modules for inputting, processing and displaying and to write functions for them.
- 4.2.5 Teacher consolidates the relevance of combining the data and associated functions into a single unit and introduces the concept of classes.
- 4.2.6 Students are asked to define some classes for various data like Books, Account, Employee etc.
- 4.2.7 The definitions are evaluated in peer and teacher provides necessary supports.
- 4.2.8 Through general discussion, the data members and member functions of above classes are identified and listed by the students.

### 4.3 Discussion on objects (40 Mins.)

- 4.3.1 Problem: *Why do we need structure variables, once a structure is defined in program?*
- 4.3.2 Students write the needs in their books, consolidate in groups and present their findings.
- 4.3.3 Teacher directs the discussion in a such a way that the following points are emerged:
  - structure specification is merely a data type definition
  - Memory allocation does not take place during the specification
  - To handle data, memory locations are needed
  - For memory allocation, variables are to be declared
  - Hence structure variable is essential for handling structured data in programs.

4.3.4 As a result of the above discussion, the class type variables are illustrated and introduces the term ‘object’ as the variable declared using a class.

**4.4 Discussion on allocation of memory for classes and objects (40 Mins.)**

4.4.1 Problem: *An object creation statement is given. Draw the layout of the memory allocation for the object.*

4.4.2 A bench wise group discussion is conducted for the above and present the layout with proper justification.

4.4.3 Teacher consolidates the presentations and the students prepare notes.

**4.5 Discussion on access lables (40 Mins.)**

4.5.1 Problem: *The class defined in activity 4.2.5 is given. How will you input values to an object of the class?*

4.5.2 Discussion is initiated by posing relevant questions to get the following points:

- Use of dot operator similar to that for structure variable
- Member functions are used to perform operations
- Need for object declaration statement
- Calling member functions using objects followed by dot operator.

4.5.3 Teacher reminds the concept ‘abstraction’ and the role of class in data hiding.

4.5.4 The access labels are introduced through diagramatic illustration and students prepare notes.

4.5.5 Simple programs using classes are discussed and taken down in Lab Diary and tried out in lab.

4.5.6 Assignment: *Define a class having an integer array as data member and member functions for inputting and displaying the values. Try out the program the lab, observe the compilation and note down the errors if any.*

**4.6 Discussion on inline functions (40 Mins.)**

4.6.1 Problem: *Write a program to find the sum of two numbers using a function.*

4.6.2 Students write the program individually, consolidate in groups and a smaple code is written on the board.

4.6.3 A general discussion is conducted to list the working of the program using the concept ‘fetch and execute cycle’. Also the limitations of such mode of execution is listed.

4.6.4 Illustrate the concept of inline function as a remedy by modifying the program.

4.6.5 Students prepare notes on inline functions with their merits and demerits.

4.6.6 The assignment given as activity 4.5.6 is referenced to initiate a discussion to identify the fact that the functions defined inside the class are automatically inline.

**4.7 Discussion on defining member functions outside class (30 Mins.)**

4.7.1 Brainstorming session is conducted to solve the problem associated with the inlining of functions defined inside the class.

4.7.2 The discussion is directed to elicit the points as follows:

- Functions having loop statement cannot be defined inside the class
- There should be provision to define member functions outside the class

4.7.3 Teacher consolidates the discussion by introducing the scope resolution operator for defining member function outside the class.

4.7.4 Students are asked to modify the program prepared as the assignment (4.5.6) in the Lab Diary and do it in the lab.

**4.8 Discussion on array of objects (30 Mins.)**

4.8.1 Students are asked to define a class, say student to represent the CE, PE and TE scores of Computer Science as data members and necessary functions.

4.8.2 Problem: *Write a program to process the scores obtained by 50 students.*

4.8.3 Through a discussion the need for array of objects is highlighted and students complete the program individually using array of objects.

4.8.4 The source codes are compared and consolidated in groups and one or two distinct codes are presented.

4.8.5 More programming exercises are given to use array of objects.

4.8.6 The works are completed in the Lab Diary and tried out in the lab.

**4.9 Discussion on using objects as function arguments and return values (60 Mins.)**

4.9.1 A suitable problem for using objects as arguments is given to the students and ask them to write program for solving the problem.

*(Eg: A class named Complex is defined for representing complex numbers. Include a member function to accept an object of the same class to find the sum of two complex numbers and return the sum as an object.)*

4.9.2 The students write individually, discuss in groups for completing the task and present their attempts.

4.9.3 Teacher uses the program presented by the groups and illustrates the use of objects as arguments and return values.

4.9.4 Learners complete the program in the Lab Diary and execute it in the lab.

**4.10 Discussion on friend function (90 Mins.)**

4.10.1 Problem: *Modify the above class definition by removing the member function that takes the object of the same class as argument.*

- 4.10.2 Modified version is presented by the students and teacher may make modifications if required.
- 4.10.3 Problem: *Define an ordinary function that accepts two objects of the Complex class and returns sum of these objects.*
- 4.10.4 Students write individually, compare and finalise the program code within the group and present them.
- 4.10.5 Let the students complete the program for finding the sum of two complex numbers.
- 4.10.6 Discussion is initiated to list out the errors that can occur in the program as follows:
- The function to find the sum is not a member function
  - The non-member functions are not allowed access to the private and protected member of the objects of a class.
  - Hence the function can not perform the addition of complex numbers
- 4.10.7 Teacher introduces the concept of friend function as a remedy by illustrating the syntax for declaration and definition.
- 4.10.8 Students are asked to modify the program, complete in the Lab Diary and execute in the lab.

**4.11 Discussion on nested classes (40 Mins.)**

- 4.11.1 Problem: *Are you familiar with the term nesting? What does it mean?*
- 4.11.2 Discussion may be directed by giving clues like nested structure, nested loop etc.
- 4.11.3 Students are asked to define sample class having nesting feature.
- 4.11.4 The products designed by the students are taken for discussion and consolidates the idea about nesting of classes and students prepare notes on nesting with examples.
- 4.11.5 More exercises like error correction and output prediction may be given to fix the idea about nesting in different ways.

**4.12 Discussion on static members (40 Mins.)**

- 4.12.1 Problem: *Define a class having roll number, name and group of study as data members. List the demerit of the class definition, if all objects of the class belong to Science group. Also suggest remedies.*
- 4.12.2 The discussion is directed to obtain the points as follows:
- Memory wastage, as same content is stored in different locations
  - Common location, as the solution
  - The possibility of sharing same location by all the objects.
- 4.12.3 Teacher introduces the concept of static members through diagrammatic illustrations
- 4.12.4 Students prepare notes on static members.

## Sample TE Questions

1. Define a class POINT with two private data members x and y and two member functions to do the following.

(a) One to read the coordinates of two points and another to print them.

(b) Define another function to find the distance between these two points

(Hint: Distance between two points = Root of  $((x_1 - x_2)^2 + (y_1 - y_2)^2)$ )

### Scoring Indicators:

(a) For correct class definition with data members and two member functions (2½)

(b) For correct function definition (2½)

2. Consider the following code fragment

```
class ABC
{
    int a;
public:
    void func( )
    {a = 10;
    . . . }
};
void main( )
{
    ABC O1;
    O1.func( );
    cout << O1.a;
    . . .
}
```

(a) What can be the probable error in the above code fragment?

(b) Re write the code fragment to correct the above error.

### Scoring Indicators:

(a) Since variable 'a' is a private member it cannot be accessed outside class 1 score

(b) For printing the value of variable 'a' using a member function 2 scores

3. Imagine five objects are created for a class.

(a) How many copies of member functions are available in the memory?

(b) How many copies of that class's data members are stored in memory?

(c) Justify your answer.

### Scoring Indicators:

(a) One 1 score

(b) Five 1 score

(c) For correct justification 2 scores

**5**

# **Constructors and Destructors**



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Identify the need of Constructors	Constructor	Observation, Understanding	Illustration, Discussion	Text Book	Notes	1
Familiarise the characteristics of constructors	Characteristics of Constructors	Observation	Discussion	Text Book	Chart listing properties of constructors	1
Identify different type of constructors	Default, Parameterised and copy constructors	Classification, Experimentation	Illustration, Lab Work	Record Book, Text Book	Notes, Programs	3
Discussion on different methods to call constructors	Implicit call, Explicit call	Observation, Understanding	Discussion	Text Book	Notes	1
.Understand the concept of constructor with default argument	Default arguments of constructors	Observing Understanding	Discussion	Record Book, Text Book	Notes	1
Identify the need of destructors and list the characteristics	Destructors	Observation, Understanding	Illustration, Discussion	Text Book	Chart listing Properties of destructors	1

## **Introduction**

This chapter discusses the means for creating, initialising and disposing of objects of classes. Constructors are special member functions used for initialising objects of its class. This chapter deals with constructors, its characteristics, different type of constructors, method of calling constructors, destructors and their characteristics through learning strategies illustration, discussion, observation, chart preparation, lab work etc.

## **Curriculum Objectives**

1. Identify the need of constructors through illustration, problem solving, discussion etc. and define constructors for various classes.
2. Familiarize the concepts and characteristics of constructors through discussion and list out the characteristics.
3. Identify different types of constructors through problem-solving, illustration, discussion etc. and develop programs using each type of constructors.
4. Understand different methods for calling constructors with the help of a program and prepare notes.
5. Understand the concept of constructors with the default arguments through a discussion, compare them with default constructors, and list out the points identified.
6. Identify the need for destructors through illustration, observation, discussion etc. and list out the characteristics.

## **Content Details**

Need of Constructors; Declaration & Definition; Types of Constructors – Default, Parameterized, Copy; Call to Constructors – Implicit and Explicit; Constructors with Default arguments; Characteristics of Constructors; Destructors – Definition and Use; Characteristics.

## **Pre requisites**

- Concept about variable initialisation
- Idea about memory allocation

## Learning Activities

### 5.1 Discussion on need for constructors (40 Mins.)

- 5.1.1 Teacher gives example for variable declaration statement and variable initialisation statement for ordinary variable to realise the impact of variable initialisation statements.
- 5.1.2 Problem: *Define a class consisting of an integer variable as data member and a member function to print value of the data member.*
- 5.1.3 The students present the class definition.
- 5.1.4 Students are asked to predict the output, when the above mentioned member function is called and list the difficulty for the prediction.
- 5.1.5 Modify the class by defining another member function to assign user supplied values. Invite the responses from students about the draw backs of using the above function. The idea, that the value can be assigned only by calling the member function explicitly, must be elicited.
- 5.1.6 Teacher consolidates and introduces the term constructor and is defined for the class defined in the activity 5.1.3.

### 5.2 Discussion on characteristics of constructors (40 Mins.)

- 5.2.1 Students are asked to list the features of the constructor defined in activity 5.1.6
- 5.2.2 Through discussion, students list some features that they can identify and teacher consolidates them.

### 5.3 Discussion for default constructors (40 Mins.)

- 5.3.1 Teacher invites responses of students in a program which contains classes having no constructor defined.
- 5.3.2 A discussion is conducted for the following points:
  - (1) C++ compiler generates a constructor even if we do not define any constructor for a class.
  - (2) It initialises the data members by any garbage value.
- 5.3.3 Teacher consolidates the discussion by introducing the default constructor, students prepare notes.

**5.4 Discussion for parameterised constructor (40 Mins.)**

5.4.1 A discussion is conducted for the limitations of default constructors.

It include the following points

- (1) It assigns fixed values to the data members of all objects
- (2) It is not possible to pass initial values to a default constructors.

5.4.2 Students are asked to give some parameters to already created constructors, and predict the output for various values for this arguments. A general discussion is conducted.

5.4.3 Teacher consolidate the points and introduces parameterised constructor , students prepare notes

5.4.4 An assignment is given for writing program code for above mentioned constructors.

**5.5 Discussion for constructor with default arguments (40 Mins.)**

5.5.1 Problem: *Construct a parameterised constructor with 3 arguments for passing values to the object.*

5.5.2 Through discussion, students write the code.

5.5.3 A general discussion is conducted to predict the output of passing values to an object in the following cases

1. Declare an object1 with one argument
2. Declare an object2 with two arguments
3. Declare an object3 with three arguments

How these values are passed to the formal arguments?

5.5.4 A discussion is conducted and students identify the necessity of assigning values to some arguments in case1 and case2.

5.5.5 Teacher consolidates the points and introduces the constructor with default argument and prepare notes.

**5.6 Discussion on copy constructor (40 Mins.)**

5.6.1 Teacher asks students, how to take an exact copy of a variable in a program.

5.6.2 Through a general discussion students recall the idea.

5.6.3 Problem: *How to take an exact copy of an existing object?*

5.6.4 A discussion is conducted to modify the program in activity 5.5 to take the exact copy of an object.

5.6.5 After discussion, teacher consolidates the points and introduces the copy constructor and prepare notes

5.6.6 Give an assignment to write programs using different type of constructors in observation book and do it in the lab

**5.7 Discussion on different methods to call constructors (40 Mins.)**

5.7.1 A general discussion is conducted to collect and observe all the object creation statements used in above examples.

5.7.2 A discussion is conducted on such statements and realises that the constructor name is not provided in those statements.

5.7.3 Teacher consolidates and introduces the concept of implicit call .

5.7.4 Teacher arises a problem, how to invoke the constructor by specifying its name?

5.7.5 After general discussion, teacher consolidates and introduces the concept of explicit call

5.7.6 An assignment is given for writing a program to differentiate the implicit and explicit call

5.7.7 An assignment is given for preparing a chart showing the characteristics of constructors

**5.8 Discussion on the need for destructors and its characteristics (40 Mins.)**

5.8.1 Teacher illustrates the need for removal of memory allocation created by an object in a program

5.8.2 General discussion is conducted for destroying an object

5.8.3 Discussion is concluded and the concept of destructor is introduced.

5.8.4 Students point out the features of destructor

5.8.5 An assignment is given for preparing a chart showing the characteristics of destructors

**Sample TE Questions**

1. Explain the role of a default constructor? When is it considered equivalent to a parameterised constructor? Support your answer with examples.

(5 scores)

**Scoring Indicators:**

- |       |   |          |
|-------|---|----------|
| (i)   | concept about default constructor                         | 1 score  |
| (ii)  | Differentiation of default with parameterised constructor | 3 scores |
| (iii) | Explanation using example                                 | 1 score  |

2. Declare a class STUDENT having the data members name, roll number, grade and two member functions one to get the data and the other to display the data. Add suitable constructor and destructor.

(5 scores)

***Scoring Indicators:***

- |       |   |           |
|-------|---|-----------|
| (i)   | class declaration with data members         | 1 score   |
| (ii)  | member function declarations and definition | 1½ scores |
| (iii) | suitable constructor , destructor           | 1 score   |
| (iv)  | code for reading and displaying data        | 1½ scores |

**6**

# **Function Overloading and Inheritance**



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Product	Time (Periods)
Familiarise the concepts of function overloading.	Function Overloading	Observation, Communication.	Discussion Labwork	Notes Record Book	Notes Record Book	2
Identify the role of function over loading in implementing polymorphism	Compile time polymorphism.	Observing, Communicating.	Discussion Lab work	Text Book, Record Book	Notes, Record Book	2
Recognise the different types of constructors as constructor overloading	Constructor Overloading.	Observation, Comparison	Labwork Discussion Illustration	Text Book Record Book	Notes Record Book	2
Analyse functions with default arguments and compare them with function overloading	Default Arguments	Observation, Communication	Discussion Illustration	Text Book	Notes	1
Formulate the idea about different types of polymorphisms.	Compile Time & Run Time Polymorphism.	Observation, Communication	Discussion Illustration	Text Book	Notes	1

**UNIT PLAN**

<b>Curriculum Objectives</b>	<b>Concepts/Ideas</b>	<b>Process Skills</b>	<b>Activities</b>	<b>Learning Aids/ Materials</b>	<b>Evaluation/ Product</b>	<b>Time (Periods)</b>
Formulate the idea about inheritance	Inheritance	Observation, Communication.	Observation Discussion	Text Book Notes	Notes	2
Familiarise the impact of visibility modes in inheritance	Private Public Protected	Observation Communication.	Observation Discussion Labwork	Text Book Notes, Lab Diary	Notes, Lab Diary	1
Familiarise different type of inheritances	Single Inheritance Hybrid Inheritance Multiple Inheritance Multilevel Inheritance	Observation, Comparison Communication	Discussion Observation	Text Book Notes	Notes Charts	2
Acquire the idea about abstract and virtual base classes	Abstract Class Virtual Base Class	Observation Communication	Discussion Observation	Text Book Notes	Notes	2
Analyse inheritance, friendship and containership	Friendship, Containership.	Observation Comparison	Discussion Illustration	Text Book	Notes	1

## Introduction

The students are already familiar with user-defined functions, classes and objects. This chapter introduces function overloading and inheritance, two major concepts of OOP. Polymorphism is introduced through the use of a group of functions with the same name. The concept of reusability of an existing code is also introduced and implemented through inheritance. The concepts of base class and derived class, protected visibility mode and different types of inheritance are introduced. The ideas are to be conveyed through discussions, illustrations and lab work on various programming exercises.

## Curriculum Objectives

1. Familiarise the concepts of function overloading through function definitions, output prediction etc. and elicit the features of over loading.
2. Identify the role of function over loading in implementing polymorphism through discussion and prepare notes.
3. Recognise the different types of constructors as constructor overloading through observation, discussion etc. and prepare notes by comparing constructor overloading and function over loading.
4. Analyse functions with default arguments and compare them with function overloading and develop sample programs.
5. Formulate the idea about different types of polymorphisms through discussion and prepare notes.
6. Formulate the idea about inheritance through illustrating real life situations and prepare notes.
7. Familiarise the impact of visibility modes in inheritance through illustration, picturising, discussion etc. and prepare a table showing the features of various modes.
8. Familiarise different type of inheritances through illustration of real life situation, charting, discussion and develop the lay out, syntax of each.
9. Acquire the idea about abstract and virtual base classes through observation, discussion etc. and prepare notes.
10. Analyse inheritance, friendship and containership through group discussion and summarise similarities and dissimilarities among them.

## Content Details

Function Overloading – Declaration, Definition and Use; Calling Overloaded functions; Constructor overloading V/s Function overloading; Functions with Default arguments V/s Overloading; Types of Polymorphism – Static and Dynamic (only theoretical concept).

Inheritance – Concept; Base and Derived Classes; Defining Derived class; Visibility Modes; Types of Inheritance; Abstract Class and Virtual Base Class (only theoretical concept); Derivation V/s Friendship; Inheritance V/s Containership.

## Pre requisites

- Ability to write user defined functions
- Knowledge about the concept of class and object
- Idea about various visibility modes

## Learning Activities

### 6.1 Discussion on need for function overloading (30 Mins.)

6.1.1 Problem: Write two functions in C++ (Integer values are passed as arguments to the function)

a. To add two integer numbers

b. To add three integer numbers

6.1.2 Learners write the program individually

6.1.3 One or two selected students present their programs.

6.1.4 Learners may compare call statements and list out their differences and provide suggestions.

6.1.5 Learners list out their observations and teacher consolidates with the following points:

- (a) Discrepancy in giving different names to functions doing the same process.
- (b) Difficulty in remembering function names
- (c) Less readability etc.
- (d) Advantage in giving the same name to the functions.

6.1.6 Teacher concludes that in C++ more than one function can have the same name.

### 6.2 Discussion on rules to be followed in writing overloaded functions (90 Mins.)

6.2.1 Problem: Learners are asked to modify the program in activity 6.1 by giving the same name to both the functions and identify the problems in calling them.

6.2.2 Group discussion is conducted and learners note the points.

6.2.3 Each group presents their findings and teacher consolidates as follows:

It is difficult for the compiler to identify the function to be executed, as both functions carry same name.

6.2.4 Learners are asked to give suggestions to rectify this problem.

6.2.5 Learners present their observations and teacher concludes the most important points.

- (a) By looking at the parameters
- (b) No of parameters should be different
- (c) Type of parameters should be different
- (d) Order of parameters should be different

- 6.2.6 Idea of resolving the problem through the role of arguments is introduced.
- 6.2.7 Teacher consolidates the discussion by illustrating function overloading and how it relates to polymorphism. Constructor overloading is also illustrated.

**Note:** Function overloading is compile time polymorphism.

- 6.2.8 Give lab assignment on programs related to function overloading.

### **6.3 Discussion on Default Arguments (30 Mins.)**

6.3.1 Problem: *Is it possible to achieve the tasks performed by the two functions in activity 6.1.1 through a single function?*

6.3.2 Students discuss the situation and list the possibilities as follows:

1. Only one function is needed with maximum parameters.
2. Assign default values to all arguments

6.3.3 Teacher illustrates functions with default arguments and their use.

6.3.4 Students are given exercises to overload functions by using default arguments.

### **6.4 Discussion on Concepts of Inheritance (30 Mins.)**

6.4.1 Problem: *Both father and mother in a family have assets worth 1 lakh each. They have only one son. Find out the possible amount the son can inherit from each of his parents and the total amount the child can inherit from both his parents and the total amount the son can have.*

6.4.2 Students discuss the situation and list the possibilities.

6.4.3 Teacher consolidates the points

1. Parent will utilize some assets for their personal use.  
(Note: To demonstrate private visibility)
2. The remaining assets with the child will be what he got from the parents + what he made on his own.

6.4.4 Teacher introduces the term inheritance on the basis of this real life scenario.

6.4.5 Learners are asked to list out similar real life situations in which inheritance is involved.

### **6.5 Discussion on need for Inheritance (40 Mins.)**

6.5.1 Problem: *In the above problem father, mother and son are treated as three separate classes, each with private members (a & b) and member functions (which displays the asset with each). Assume that the child can access the members and member functions of the parent class. List out the possible advantages in doing so.*

6.5.2 Group discussion is conducted on the possible advantages and learners make note of the points observed.

6.5.3 Teacher consolidates the point – reuse of parent class members/ member functions without disturbing the existing class.

**6.6 Discussion on implementation of Inheritance (30 Mins.)**

6.6.1 Teacher demonstrates the implementation of inheritance through a sample program, which implements the activity 6.5. It should contain private (a & b) and protected variables (c & d).

6.6.2 The students identify the different levels of class in inheritance and through a discussion identify base class and derived class.

6.6.3 Teacher illustrates the difference between various visibility modes.

**6.7 Discussion on Friendship (40 Mins.)**

6.7.1 Problem: *Your friend is given the key to your house. Comment on this scenario.*

6.7.2 After a general discussion, learners list out their observations

1. The friend can access or use any thing inside the house whenever he needs.
2. In the absence of owner, the friend can open the house for putting items like a new gas cylinder.
3. Without getting the key (i.e. permission) from the owner the friend would not be able to enter the house, so this is not a dangerous scenario.

6.7.3 Teacher illustrates friend functions and friend classes in connection with the above scenario and concludes the discussion by illustrating advantages and use of friendship. Friend function / Friend class can access the private and protected members of class is stressed.

6.7.4 Learners understand friendship and prepare a table comparing friendship and derivation.

**6.8 Brain storming session on types of inheritance (40 Mins.)**

6.8.1 What are different ways in which the classes A, B, C, D, E can be arranged as base class and derived class? Each design can contain a minimum of two classes or more. Illustrate by drawing a diagram. Students create designs in groups.

6.8.2 A member from each group is asked to present their design on the black board, and the designs are discussed in the class.

6.8.3 Teacher consolidates by giving each design appropriate name and discusses the different types of inheritance and their advantages.

6.8.4 Students are given the assignment of preparing charts showing layout of classes in different types of inheritance.

**6.9 Discussion of Constructors & Destructors in Inheritance (40 Mins.)**

6.9.1 Problem: *Students are asked to modify the program in activity 6.7 by including constructors and destructors. The behavior of constructors and destructors during inheritance is discussed. (Note: Constructors and destructors may be defined to display a message.) Students are asked to predict the output during the creation and destruction of objects.*

6.9.2 The program is compiled and executed in the lab.

6.9.3 Observation is made on the precedence of execution of constructors and destructors.

6.9.4 Teacher illustrates the special cases such as

- a. Explicit invocation of constructors
- b. Constructors in multiple inheritance

**6.10 Lab Session & Illustration of Abstract Class (30 Mins.)**

6.10.1 Problem: *Students are asked to try creating objects of base class in activity 6.9. By putting the keyword 'abstract' before the keyword 'class' of the base class.*

6.10.2 Learners make observations and note them.

When abstract keyword is used C++ compiler does not allow making object of that class.

6.10.3 Teacher explains why it does not allow making instances of the class and consolidates by explaining use of abstract class.

**6.11 Lab Session and Demonstration of Virtual base class (30 Mins.)**

6.11.1 Learners are asked to execute the following code in the lab and make observations.

```
class A
{ public:
  int a;
};
class B : public A
{ public:
  int b;
};
class C : public A
{ public :
  int C;
};
class D : public B, public C
{ public :
  int D;
};
void main()
{ D ob;
  ob.a = 5; }
```



**7**

# Pointers



## UNIT PLAN

Curriculum Objectives	Concepts/Contents	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Period)
Familiarise the memory allocation for the execution of a program	Memory Allocation	Observation, Communication.	Observation Discussion	Text Book Notes Charts	Notes	1
Familiarise different types of memory allocation techniques	Static allocation Dynamic allocation	Observation Comparison	Observation Discussion	Text Book Notes,	Notes	1
Formulate the concept of pointers and understand its declaration and initialization	Pointer Creation	Observation	Observation Discussion Labwork	Text Book Notes Record Book	Notes Record Book	2
Familiarise pointer arithmetic	Pointer Arithmetic	Observation	Observation Discussion Labwork	Text Book Record Book	Notes Record Book	1
Understand the memory allocation operators	new delete	Observation Communication	Observation Discussion Labwork	Text Book Record Book	Notes Record Book	1

**UNIT PLAN**

<b>Curriculum Objectives</b>	<b>Concepts/Contents</b>	<b>Process Skills</b>	<b>Activities</b>	<b>Learning Aids/ Materials</b>	<b>Evaluation/ Products</b>	<b>Time (Period)</b>
Familiarise the memory leak problem	Memory leak	Observation, Communication.	Observation Discussion	Text Book	Notes	1
Familiarise the use of pointers in handling string data	Pointers to array of characters	Observation Comparison	Discussion Lab Work	Text Book Lab Diary	Notes Lab Diary	1
Examine the role of pointers as function arguments	Pointers as arguments	Observation	Discussion Labwork	Text Book Notes	Notes Lab Diary	1
Examine the role of structure pointers and self referencial structures	Pointer to structures, Self-referencial structures	Observation	Discussion Labwork	Text Book Lab Diary	Notes Lab Diary	1
Understand the role of pointers to objects	Pointers to Objects, this pointer	Observation Communication	Discussion Labwork	Text Book Lab Diary	Notes Lab Diary	1
Familiarise the concept of linked list	Linked lists, node, info, next	Observation Communication	Discussion	Notes	Notes	1

## **Introduction**

Pointer, a powerful tool in C++ is introduced in this chapter. Functionalities like dynamic allocation of memory, passing by reference etc. are discussed in detail. The relation of pointers with classes, objects, structures etc. are also discussed. The activities like discussion, observation, error correction, output prediction, lab work etc. are used to transact the concepts.

## **Curriculum Objectives**

1. Familiarise the memory allocation for the execution of a program through observation of chart and prepare notes and diagrams.
2. Familiarise different types of memory allocation techniques through discussion and distinguish them.
3. Formulate the concept of pointers and understand its declaration and initialization through observation, discussion etc. and create and initialise pointers.
4. Familiarise pointer arithmetic through discussion, output prediction etc. and prepare notes on pointer arithmetic.
5. Understand the memory allocation operators through discussion and list out its use.
6. Familiarise the memory leak problem through illustration and list out the situation that result in memory leak.
7. Familiarise the use of pointers in handling string data through illustration by an example and develop programs for string manipulation.
8. Examine the role of pointers as function arguments through the discussion, output prediction etc. and develop programs.
9. Examine the role of structure pointers and self referential structures through illustration, discussion etc. and prepare notes covering definitions of self referential structures.
10. Understand the role of pointers to objects through illustrations, discussion and develop sample program.
11. Familiarise the concept of linked list through the illustration of real life situations and list down its advantages.

## **Content Details**

Memory mapping for a Program; Dynamic and Static Memory allocation; Declaration and Initialization of Pointers; Pointer Arithmetic; Dynamic Allocation/De-allocation Operators; Pointers and Arrays – One dimensional and Two dimensional; Memory leak; Pointers and String; Pointers as Function arguments; Pointer to a Structure; Self-referential structures; Pointers to objects; Concept of Linked Lists.

## Pre requisites

- Knowledge of different data types and their declaration.
- Ability to write user defined function.
- Idea about the declaration and usage of structures and class.

## Learning Activities

### 7.1 Discussion on allocation of memory and need for pointers (30 Mins.)

7.1.1 Problem: A C++ program stores the marks of a few students in an array. The number of students can vary from 50 to 100. Discuss the efficiency of this program in relation to memory allocation.

7.1.2 Through discussion students make notes and list out their observations. The discussion may include the following:

- (a) If the array size is 100 and fewer elements are stored, then wastage of memory occurs.
- (b) If the array size is 50 at runtime the program will fail if marks of more number of students are to be stored.
- (c) They identify, data type whose length can be varied or can be allocated at runtime is needed to rectify this problem.

7.1.3 Teacher consolidates and introduces the ideas of static and dynamic allocation. Consolidation includes the following points.

- (a) Dynamic allocation is done through the pointers
- (b) Memory is allocated from the heap memory (free store).

### 7.2 Discussion on address, name and value relationship (30 Mins.)

7.2.1 Problem: *What happens when the statement `int a;` is executed?*

7.2.2 A general discussion is conducted to elicit the following points:

- (a) Each data type has a specific predefined size and memory space is allocated to a variable depending on the size of the variables data type (2 bytes here).
- (b) The variable has a name.
- (c) Each location in memory will have an address.

7.2.3 Teacher consolidates the discussion and introduces the use of '&' operator for getting the address of a variable.

7.2.4 The teacher illustrates a sample program, which displays the address of a variable.

### 7.3 Discussion on declaration and initialisation of pointers (15 Mins.)

7.3.1 Teacher demonstrates the declaration and initialisation of pointers through a sample program, which uses pointer variables

7.3.2 Learners observe and make notes.

**7.4 Problem solving on pointers (15 Mins.)**

7.4.1 Teacher illustrates the following program and students are asked to write the output.

```
#include <iostream.h>
void main()
{
    int a,b;
    int *x, *y;
    a = 10;
    b = 5;
    x = &a;
    y = x;
    *y = b;
    cout << a << b << *x << *y;
}
(output : 5555)
```

7.4.2 Students find out the output and present it

7.4.3 Teacher illustrates the correct output and provides justification if necessary

**7.5 Discussion on pointer arithmetic (15 Mins.)**

7.5.1 Teacher illustrates a sample program. The problem can contain declaration of various types of pointers, use of increment and decrement operators. Learners are requested to note the value of each pointer variable (not the value of the location to which it pointing to) after each increment and decrement operation.

7.5.2 Students list their outputs and present them.

7.5.3 Teacher can make suggestions if required.

Only addition or subtraction operators can be used with pointer variables.

**7.6 Discussion on relational operations on pointers (15 Mins.)**

7.6.1 Problem: *Is there any relevance in using relational operators on pointers?*

7.6.2 After a general discussion the learners note points.

7.6.3 Teacher consolidates their findings as follows:

Relational operations other than = and != are not needed for pointers as no address is smaller or larger than the other.

**7.7 Illustration of dynamic memory allocation (15 Mins.)**

7.7.1 Teacher illustrates the 'new' operator through a sample code.

7.7.2 Students make observations and prepare notes.

7.7.3 Learners are given an assignment to draw a chart showing the pictorial representation of what happens when:

- Pointer variable is declared
- Pointer variable is initialised using 'new' operator.

**7.8 Discussion on memory leak (30 Mins.)**

7.8.1 Problem: *A Person takes a room in a hotel. The receptionist issues the key for the room. After staying in the room for few hours, the person vacates the room without informing the hotel receptionist or leaving the key. What may be the problems that may arise during the situation.*

7.8.2 Learners discuss in group and make note of observation and presents them.

(a) Since receptionist does not know that the room is vacant, he won't allot it to another person.

(b) Room unnecessarily remains vacant.

7.8.3 Teacher illustrates memory leak and the point, allocated memory should be deallocated after its use is introduced and learners prepares notes.

**Note:** Assigning the result of 'new' statement to a pointer that was already pointing to an allocated object is also illustrated.

7.8.4 Students are given assignment to draw a chart showing pictorial representation of how a memory leak occurs.

**7.9 Discussion on memory deallocation (15 Mins.)**

7.9.1 Problem: *What happens if allocated memory is not deallocated after its use?*

7.9.2 After a discussion points are consolidated by the teacher

(a) Other programs will not be able to use these memory.

(b) Unnecessary memory wastage.

7.9.3 Teacher consolidates the discussion by illustrating the importance of de-allocation and introduces the 'delete' operator.

**7.10 Illustration and discussion on pointers and arrays (40 Mins.)**

7.10.1 Teacher illustrates how the array name holds the address of the first element in the array.

7.10.2 Learners may be asked to list out the output if array 'a' declared as int a[10]; and if it holds 10 values.

```
(a) for (i=0; i <=10; i++)
    {cout << *a;
      a++;
    }
```

```
(b) int *p;
    p = a;
    for (i=0; i <=10; i++)
    {
      cout << *p;
      p++;
    }
```

```
(c) for (i=0; i <=10; i++)  
    {  
    cout << *(a+i);  
    }
```

7.10.3 The learners present the outputs and through a general discussion they learn the relationship between arrays and pointers.

(a) The name of the array holds the starting location of the array.

(b) No change can be made to this value. (Problem 'a' will generate error)

(c) When addition is done on pointer variables, it increments by the size of data type. (Problem (b) and (c) will print values in the array)

(d) Consecutive values in the array are stored in consecutive locations in the memory.

7.10.4 Teacher illustrates dynamic arrays. Students make note of points.

### **7.11 Lab session on pointers (60 Mins.)**

7.11.1 Learners in the lab execute problem in activity 7.4, 7.5 & 7.6.

7.11.2 Learners make observations and note them in their Lab Diary.

### **7.12 Illustration of call by reference using pointers (30 Mins.)**

7.12.1 Teacher writes on the board and illustrates a function with pointer variables as parameters

(swapping two values etc.)

7.12.2 The teacher demonstrates how the address is passed.

7.12.3 Illustration of how function returns reference is done through a sample program.

7.12.4 Learners make observations and note them.

### **7.13 Illustration of pointers to structures (40 Mins.)**

7.13.1 Teacher illustrates the declaration and initialization of a structure, through the use of a pointer to that structure, with the help of a sample code.

7.13.2 Learner make observations and take notes

7.13.3 Student groups should list out and demonstrate difference between passing a structure by reference and by value.

7.13.4 Based on the demonstration teacher illustrates the advantages.

### **7.14 Discussion on self-referential structures (30 Mins.)**

7.14.1 Discussion is conducted on self-referential structure.

7.14.2 Learners make observations.

7.14.3 Teacher provides the code for self-referential structure and illustrates.

7.14.4 Learners make observations.

7.14.5 Teacher illustrates the use of self-referential structure and provides the idea of linked list.

7.14.6 Learners produce a memory representation diagram for a three element linked list.

**Note :** Self-referential structures are used for the creation of dynamic data structures like linked list, stacks, queues, trees etc.

**7.15 Illustration of pointers to objects (40 Mins.)**

7.15.1 Teacher illustrates the declaration and initialization of an object, through the use of a pointer to that object, with the help of a sample code.

7.15.2 Learner make observations and take notes

7.15.3 Student groups should list out and demonstrate difference between passing an object by reference and by value.

7.15.4 Based on the demonstration, teacher illustrates the advantages.

7.15.5 Learners understand that objects can be created dynamically only through the use of pointers to objects.

**7.16 Demonstration of *this* Pointer (45 Mins.)**

7.16.1 Teacher illustrates a program involving *this* pointer in the lab.

7.16.2 Learners execute the program and make observations

7.16.3 Teacher consolidates the idea on *this pointer*.

7.16.4 Learners learn that *this* pointer is the pointer storing the address of the object currently invoking a member function.

**Sample TE Questions**

1. A program which implements dynamic allocation is run continuously. Each time after the execution of the program it was found that the available space in the heap memory, for dynamic allocation got reduced drastically. What may be the reason for this? Suggest a suitable reason?

***Evaluation Indicators:***

Identify Deallocation Not Done	1 score
Use Delete Command	2 scores

2. An integer pointer p is pointing to location 5000. To which location p will point after executing the statement  $p = p + 1$

**Evaluation Indicators:**

1. To the location 5002 1 score
3. This is a program to create an object dynamically. Is the implementation correct? Rectify the errors if any in the following program?

```
#include<iostream.h>
class A
{
private :
    int a;
public :
    void getdata()
    {
        cin >> a;
    }
};
void main()
{
    A *t1;
    t1->getdata();
}
```

**Evaluation Indicators:**

- Identify Object Not Created using new operator 1 score
- Writing statement for Object Creation  
(t1 = new A;) 2 scores

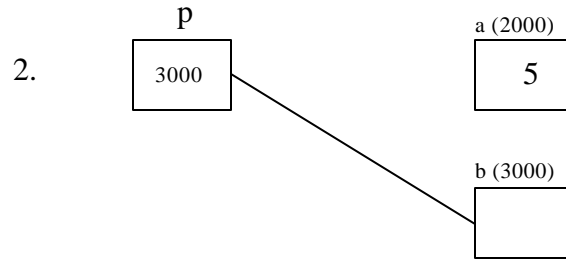
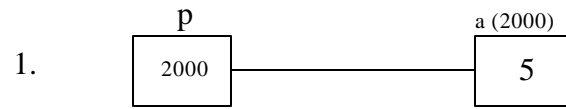
4. Find the error(s) in the following program, if any

```
#include <iostream.h>
struct student
{
    int rno;
    char name[20];
};
void main()
{
    student *s;
    s = new student;
    cin >> s.rno;
    cin >> s.name;
    cout << "Name : " << s.name;
    cout << "Reg No. :" << s.rno;
}
```

**Evaluation Indicators:**

- Identify Arrow Operator is not used 1 score  
(s->rno;)

5. Following two pictures are trying to explain a scenario in pointers. Can you identify the scenario.



***Evaluation Indicators:***

- |  |          |
|--|----------|
| 1. Memory leak   | 2 scores |
| 2. Pointer pointing to a new location without deallocating the first location. | 1 score  |

**8**

# Files in C++



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Acquire the knowledge about streams and familiarise stream objects	istream, ostream and ostream and ostream	Classification, Communication, Inference	Discussion, Illustration	Text Book	Notes and charts on stream objects	2
Familiarise the operators and functions involved in the file operations	eof(), bof(), open(), close(), read() and write()	Observation, Experimentation, Making operational definition	Illustration, Discussion, Lab Work	Text Book, Record Book	Notes on operators and functions in file operations and program code	4
Understand the concept of file modes	Different file modes	Observation, Classification, Experimentation	Discussion, Illustration, Table preparation	Text Book, Record Book	Notes on file modes, Record Book, Output for a code segment, Error corrections	2
Familiarise the random access and error handling function in file operations	seekg(), tellg(), seekp(), tellp(), good(), bad(), fail(), clear() etc.	Observation, Classification, Communication, Making operational definition	Discussion, Illustration, Lab Work	Text Book, Record Book	Notes, Program code in the Record Book, Output for a code segment, Error corrections etc	4

## Introduction

In a C++ program, files provide a method for permanent storage of data. Files are inevitable for applications that need data storage. Students may need file operations in complex programs intended for purposes such as assignments, project etc. The header file `fstream.h` contains operators and functions for file operations.

This chapter deals with file handling in C++. The basic concepts in file operations are explored through discussions, illustrations and lab work

## Curriculum Objectives

1. Acquire the knowledge about stream and familiarise stream object through discussion and prepare notes.
2. Familiarise the operators and functions involved in the file operation through illustration and develop sample program.
3. Understand the concept of file modes through discussion, illustration, output prediction etc.
4. Familiarise the idea about random access and error handling function through problem solving, discussion etc. and prepare a list of functions in each category.

## Content Details

Use of header file `fstream.h`; Types of stream objects – `fstream`, `ifstream`, `ofstream`; Reading and writing characters from/to disk; Detecting end of file; `open()` and `close()` functions; Reading and writing objects from/to disk using `read()` and `write()` functions; File mode Constants; File pointers for Random Access – `seekg()`, `tellg()`, `seekp()`, `tellp()`; Error handling functions.

## Pre requisites

- Ability to write programs using structure, class and functions
- Knowledge about concepts of secondary memory
- Console I/O operations in C++

## Learning Activities

### 8.1 Discussion on need of files in a C++ program (40 Mins.)

8.1.1 Problem: *Program to calculate the grade for the S.S.L.C. examination for the entire state*

8.1.2 Discuss the steps in implementing the program

8.1.3 A group discussion is conducted to elicit the following problems

- Handling of huge amount of data
- Need of permanent storage of data

8.1.4 Teacher consolidates the discussion by pointing out the need of files in the program

**8.2 Discussion on stream objects (40 Mins.)**

- 8.2.1 Problem: *List out the components involved in I/O operations in C++*
- 8.2.2 Students list the components of console I/O operations in C++, such as header file, operators, functions etc.
- 8.2.3 A discussion is conducted to analyse the stream objects involved in I/O operations, giving some sample statements. Students suggest the changes needed to perform I/O operations with files
- 8.2.4 Teacher illustrates the changes needed in the file I/O operations comparing with console I/O operations.
- 8.2.5 Discuss the concept of file stream objects and students prepare a chart showing the layout of stream classes

**8.3 Illustration of basic file operations (30 Mins.)**

- 8.3.1 Problem: *Understand the concept of open, input, output and close operations for a file*
- 8.3.2 A selected student is provided with a text book of computer science and a note book
- 8.3.3 He/she demonstrates the process of copying a program code from the text book to the note book
- 8.3.4 Through a discussion, elicit the following operations involved in the process
  - Open the books
  - Read the content from the text book
  - Write the content to the note book
  - Close the books after the operation is over
- 8.3.5 Teacher consolidates the discussion by comparing the above operations with the file operations

**8.4 Familiarisation of the operators and functions in file operations (60 Mins.)**

- 8.4.1 Problem: *Program to write roll number and marks of a set of students in a file. The data is given as input to the program. Read the data back from the file and display it.*
- 8.4.2 Discuss the different methods to open and close a file to a stream object. Point out the necessity of closing a file after the use.
- 8.4.3 Through a discussion, familiarise the functions and operators for the file I/O operations, by comparing it with console I/O operations.
- 8.4.4 Work out the program given in the step 8.4.1
- 8.4.5 Give lab work on programs involving file operations

**8.5 Discussion on binary functions for the file operations (40 Mins.)**

8.5.1 **Problem** :Write a program with options to

(a) Write roll number, name and address of a group of students to a file.

(b) Output the details of students stored in the file

Use structure variable to handle the students data

8.5.2 Students write the program individually

8.5.3 Discuss the easiness in coding, if it is possible to read and write the structure values as whole, directly to the file, instead of operating the individual element values.

8.5.4 Familiarise the read and write functions as a method for operating data in structure variables and objects directly to a file

8.5.5 Rewrite the program using read and write functions and implement the program in the lab

8.5.6 Discuss the difference between text and binary functions. List out the advantages in using binary functions to handle numeric data.

**8.6 Illustration on file modes in C++ (40 Mins.)**

8.6.1 **Problem**: Understand different file modes in C++

8.6.2 Teacher illustrates some situations that need to control of operations allowed to a file. (For example, a file containing student's mark list may be opened in read only mode in order to prepare the promotion list.)

8.6.3 A discussion is conducted to suggest possible methods for controlling file operations

8.6.4 Teacher consolidates the discussion and familiarise the concept of file modes

8.6.5 Discuss the features of each mode and the file mode constants

8.6.6 Prepare notes on file mode constants

**8.7 Illustration of random access operations (25 Mins.)**

8.7.1 A selected student is provided with a text book with index and another without index. Content of both the books should be on same subject

8.7.2 Ask the student to find the page explaining a particular topic from both the books

8.7.3 Discuss the role of index in locating the page directly, resulting in faster operation

8.7.4 Teacher consolidates the discussion by illustrating the role of random access operations for faster file operations by comparing it with the above process

**8.8 Familiarisation of random access functions (40 Mins.)**

8.8.1 Through a discussion, familiarise the random access functions for file operations

8.8.2 Discuss the role of random access functions to display the details of a student whose roll number is given as input to the program given in the activity 8.5



**Scoring Indicators:**

- (a) The mode specification ios::in 1 score
  - (b) if is a keyword 1 score
  - (c) Wrong arguments 1 score
7. Briefly discuss the following
- (a) Necessity of files in a C++ program
  - (b) Method for creating a new file
  - (c) Any two functions to operate text mode data from a file
  - (d) Binary functions for file operation
  - (e) Methods to determine start and end of data in a file

**Scoring Indicators:**

- (a) Points relating to permanent storage of data 1 score
  - (b) Open the file in write mode 1 score
  - (c) Explanation of any two text mode functions 1 score
  - (d) Explanation of read and write functions 1 score
  - (e) Concept of BOF and EOF 1 score
8. Is it possible to input and output data members of an object directly to a file? Illustrate your argument through a program to write the details of an employee, such as id, name and department to a file *emp.dat*

**Scoring Indicators:**

- (a) Concept of binary functions 1 score
  - (b) Class declaration 1 score
  - (c) Code for file operations 2 scores
  - (d) Correct logic 1 score
9. Consider a C++ program that store following details of employees: employee number, name, department and salary in a file *emp.dat*.
- (a) Discuss the functions needed to change the salary of an employee stored in the file
  - (b) Write a C++ program to display the details of an employee whose employee number is given as input

**Scoring Indicators:**

- (a) Points relating to random access functions 2 scores
  - (b) Use of random access functions 1 score
  - (c) Correct syntax of the program 1 score
  - (d) Correct logic 1 score
10. Discuss two modes that open a file by deleting the existing content

**Scoring Indicators:**

Explanation of two file mode constants such as ios::out alone and ios::trunc.

1 score each

9

# Boolean Algebra



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time Periods
Familiarisation of basic logic operators	OR, AND, NOT Operators, Truth table	Observation Classification	Discussion Preparation	Text Book	Notes	1
Acquire knowledge about basic logic gates.	OR, AND and NOT gates	Observation Classification	Discussion, Illustration, Designing	Text Book	Circuit diagram Chart	2
Understand basic postulates, theorems	Basic postulates, DeMorgan's theorems & other theorems	Communication	Discussion Chart creation	Text Book Chart	Notes Chart	2
Familiarise duality principles	Duality principles	Observation, Realisation	Discussion, Illustration	Text Book	Notes	2
Familiarise methods for evaluation of Boolean expression	Truth table method and algebraic method	Observation Communication	Illustration Problem solving Designing	Text Book	Notes Problems	2
Familiarise advanced gates	NAND, NOR, XOR, XNOR	Classification	Discussion Chart preparation	Text Book	Notes Charts	3
Develop skills on implementations of basic gates using universal gates	Implementation of basic logic gates	Understanding	Illustration Design of circuit diagram	Text Book	Notes	2
Acquire knowledge about application of logic gates.	Adders, Encoder, Decoder	Understanding Classification	Illustration Design	Text Book	Notes Problems	2

## Introduction

Boolean Algebra is a set of rules which are extremely suitable for digital circuits. It was discovered by an English Mathematician George Boole (1815-1864). The principle behind Boolean Algebra is that the answers to its equations are either true or false. The digit 1 or letter T is used to represent a true solution and digit 0 or letter F for a false solution. Boolean Algebra is an ideal tool for the design and analysis of logic circuits used in computers. This chapter deals with the theorems, methods for evaluation of Boolean expressions, and the concept of different logic gates. Learning strategies like discussion, illustration, problem solving methods, chart preparation, framing of circuit diagram etc. are used.

## Curriculum Objectives

1. Acquire the knowledge about basic gates through discussion, illustration and prepare circuit diagrams for Boolean expressions.
2. Understand the basic postulates and theorems in Boolean Algebra through discussion, illustration etc. and prepare a chart showing them.
3. Familiarise the concepts of principles of duality through observation, discussion etc. and find the dual of given expressions.
4. Familiarise evaluation of Boolean expression through illustration, discussion etc. and work out some examples.
5. Familiarise advanced gates through table preparation, discussion etc. and prepare chart showing them.
6. Develop skills for implementing basic logic gates using universal gates through illustration truth table creation etc. and draw corresponding circuit diagram.
7. Acquire knowledge about applications of logic gates through discussion, illustration etc. and develop circuit diagrams for sample applications.

## Content Details

Basic Logic Operations – OR, AND, NOT; Truth Tables; Basic Gates – OR, AND, NOT; Basic Postulates of Boolean algebra; Basic theorems of Boolean algebra; De Morgan's theorems; Principle of Duality; Evaluation of Boolean expressions (using Truth table and Algebraic method); Advanced Gates – NAND, NOR, XOR, XNOR; Implementation of Basic Logic Gates using NAND and NOR Gates; Application of Logic Gates – Half adder, Full adder, Encoder, Decoder (using Basic Logic Gates).

## Pre requisites

- Idea about switching circuits
- Concept of binary addition

## Learning Activities

### 9.1 Discussion on basic logic operators and truth table (20 Mins.)

9.1.1 Some statements as follows may be given to the students and asked to combine these for the specified purpose.

1.  $\text{age} < 20$
2. pass in SSLC

*Instruction:* The above set of statements are to be combined for +1 admission.

1. Member of NSS
2. Member of NCC

*Instruction:* Bonus marks will be awarded for such students in +1 admission

1. Kerala is a state in Pakistan

*Instruction:* Correct this statement without changing any word in the statement (but can add a word).

9.1.2 Through a discussion, learners identify the difference between OR, AND and NOT.

9.1.3 The teacher introduces the same words as operators in boolean algebra and consolidates by distinguishing these operators.

9.1.4 Exercises are given to present statements related to real life situations for using these operators

### 9.2 Introduce the concept of Truth Table (20 Mins.)

9.2.1 Problem: Consider a switching circuit that has two switches  $S_1$  and  $S_2$  connected in series with a battery and one electric bulb. Find out the switch states (ON, OFF) for which current will flow through the bulb. Prepare a table which contains input and output values for the above problem.

9.2.2 Consider the same circuit with switches  $S_1$  and  $S_2$  connected in parallel. Find out the switch states for which the current will flow through the bulb. Prepare the table which contains input/output values

9.2.3 Through a group discussion, students list out the combination of all possible inputs and their effect on the outputs using a table

9.2.4 Teacher consolidates the discussion and introduces the name "Truth Table".

### 9.3 Discussion on basic gates and preparation of circuit diagrams (80 Mins.)

9.3.1 Teacher displays a chart showing AND, OR and NOT gates and illustrates the operations performed by the gates.

9.3.2 Problem: Write truth table for each of the given gates.

9.3.3 Students write individually, discuss in groups and present the tables.

9.3.4 A brainstorming session is conducted to elicit the symbols used for the basic Boolean operations by bringing the attention to the mathematical operators.

(If A and B are the inputs in the tables designed above, ask the students to formulate an expression for the output column using the mathematical operator)

9.3.5 Teacher consolidates and introduces the symbols for the basic Boolean operations.

9.3.6 Teacher draws some simple Boolean expressions on the board and students draw the corresponding circuit diagrams.

#### **9.4 Discussion on basic postulates (25 Mins.)**

9.4.1 Teacher gives the postulates to each group for developing the idea about it with suitable values.

9.4.2 Each group presents the postulates after group discussion.

9.4.3 An assignment is given to prepare a chart showing basic postulates

#### **9.5 Discussion on basic theorems (25 Mins.)**

9.5.1 Teacher gives the theorems to each group for developing the idea by proving them with possible values.

9.5.2 Each group presents the theorems after group discussion.

9.5.3 Teacher consolidates and students prepare notes on the proof of each theorem

9.5.4 An assignment is given to prepare a chart showing basic theorems.

#### **9.6 Discussion on the concept about duality principle (80 Mins.)**

9.6.1 Teacher displays a chart showing binary addition and multiplication as follows

$$0+0=0 \qquad 1.1=1$$

$$0+1=1 \qquad 1.0=0$$

$$1+0=1 \qquad 0.1=0$$

$$1+1=1 \qquad 0.0=0$$

Through general discussion, find out the relationship between these tables.

9.6.2 After a discussion, students find the following points

1. changing values from 0 to 1 and from 1 to 0

2. changing operators '+' to '.' and from '.' to '+'

9.6.3 Teacher consolidates and introduces the concept of duality principle.

9.6.4 An assignment is given for writing the duality of all theorems they have studied.

**9.7 Discussion on De Morgan's theorems (30 Mins.)**

- 9.7.1 Through a general discussion, De Morgan's First law is illustrated by the teacher.  
(Students' involvement may be ensured by eliciting the laws/postulates used in each step in the derivation of the theorem)
- 9.7.2 Teacher gives the statement of De Morgan's second law to each group for finding the derivation. After group discussions students present the proof before the class. Teacher consolidates it and the students write the proof.
- 9.7.3 Assignment:- Prove De Morgan's theorems using truth table.

**9.8 Discussion on evaluation of Boolean expression using truth table method (40 Mins.)**

- 9.8.1 Problem: *Given some Boolean expressions to each group and asks them to evaluate the expression using the truth table.*
- 9.8.2 Through a discussion, each group evaluates the expression using truth table and presents it.
- 9.8.3 Teacher consolidates them and gives an assignment to evaluate some Boolean expressions using the truth table method.

**9.9 Discussion on evaluation of Boolean Expression using Algebraic method (40 Mins.)**

- 9.9.1 Problem: *Given some Boolean expressions to each group and asks them to evaluate the expression using the basic postulates and theorems already studied.*
- 9.9.2 Through a discussion, each group evaluates the expression and presents it .
- 9.9.3 Teacher consolidates them and introduces the Algebraic method
- 9.9.4 An assignment is given to evaluate the Boolean expression given in activity 9.8.1 using algebraic method.

**9.10 Discussion on advanced gates (80 Mins.)**

- 9.10.1 Problem: *Given the circuit diagram for AND gate and NAND gate ,OR gate and NOR gate to each group and asks them to prepare the Truth Table and compare them*
- 9.10.2 Through group discussion, students compare AND gate with NAND gate, OR gate with NOR gate and prepare a chart showing the details of each gate and present them.
- 9.10.3 Teacher shows a chart containing circuit diagram of XOR gate and its expression and asks the students to prepare the truth table for the expression.
- 9.10.4 After its consolidation, the teacher modifies the circuit diagram of XOR gate by adding NOT gate to the output line of XOR and asks the students to prepare the truth table.
- 9.10.5 Through group discussion, students prepare the truth table and the teacher introduces the modified circuit as XNOR gate and draws the simplified circuit.

**9.11 Discussion on Universal gates (40 Mins.)**

- 9.11.1 Problem: *Draw the logic circuit of AND gate using NAND gate only.*
- 9.11.2 Through discussion, the logic for expressing AND using NAND is derived and ask the students to implement an AND gate using NAND.
- 9.11.3 Students try individually, discuss in groups and present the circuit which may be consolidated by the teacher.
- 9.11.4 Problem: *Draw the logic circuit of an AND gate using NOR gate, OR gate using NOR gate, OR gate using NAND gate alone.*
- 9.11.5 Proceed as specified in 9.11.2 and 9.11.3
- 9.11.6 An assignment is given
  - 1. To draw the circuit diagram using NAND gates only for given Boolean expressions.
  - 2. To draw the circuit diagram using NOR gates only for given Boolean expressions.

**9.12 Discussion on Half Adder (40 Mins.)**

- 9.12.1 Problem: *List down the rules for adding two bits and prepare a table showing the inputs and outputs.*
- 9.12.2 Through general discussion, students prepare the table and present them.
- 9.12.3 Teacher consolidates by completing the table by filling the carry column.
- 9.12.4 Teacher brings the attention of the students to the two input columns and the sum column and asks them to identify Boolean operation that has the same type of truth table.
- 9.12.5 Through general discussion, the XOR gate is identified for the sum column of the above table.
- 9.12.6 Similarly the gate for carry output is derived.
- 9.12.7 Teacher consolidates the presentations and introduces half adder and ask the students to draw its circuit diagram.
- 9.12.8 Students try individually, discuss in groups and present.
- 9.12.9 Teacher consolidates the presentations.

**9.13 Discussion on Full Adder (40 Mins.)**

- 9.13.1 By modifying the activity detailed in 9.12, the full adder can be introduced.

**9.14 Discussion on Encoder (40 Mins.)**

- 9.14.1 Problem: *Prepare a table showing the binary equivalent of the digits of octal number system (0 - 7)*

- 9.14.2 Students prepare table individually, compares them within the groups and present them.
- 9.14.3 Teacher illustrates the logic circuit of an octal to binary encoder and ask the students to prepare a table showing the outputs of the encoder for various inputs.
- 9.14.4 Students compares both the tables and realises that both the tables are same.
- 9.14.5 Teacher introduces the logic circuit as encoder for octal to binary conversion.
- 9.14.6 An assignment is given to design the encoders for decimal to binary and hexa to binary along with truth tables.

**9.15 Discussion on Decoder****(40 Mins.)**

- 9.15.1 By modifying the activity detailed in 9.14, the decoder can be introduced.

**Simple TE Questions**

1. (a) Differentiate between  $A'B+AB'$  and  $A'B'+AB$ . 1 score
- (b) Draw the logic circuit for each expression. 2 scores
- (c) Construct the same circuits using NAND gate only 2 scores

**Scoring Indicators:**

- (a) Identification of XOR and XNOR  $\frac{1}{2} + \frac{1}{2} = 1$  score
- (b) Circuit diagram for each  $1 + 1 = 2$  scores
- (c) Logic circuit using NAND gate alone  $1 + 1 = 2$  scores

**(Total Scores: 5)**

2. (a) Why NAND and NOR gates are more popular than the basic AND,OR and NOT gates? (1)
- (b) Draw the truth table and circuit diagram for 4-input AND gate. (2)
- (c) Modify this circuit using NAND gate only (2)

**Scoring Indicators:**

- (a) Idea about universal gates 1 score
- (b) (i) Truth table for 4-input AND gate 1 score
- (ii) Circuit diagram for 4-input AND gate 1 score
- (c) Diagram using NANDgate 2 scores

**(Total Score: 5)**

3. (a) If  $a+b = a+c$  and  $a.b=a.c$ , prove that  $b=c$  1 score
- (b) Check the validity of the equation:  
 $(X+Y).(Y+Z).(Z+X)=(X'+Y').(Y'+Z').(Z'+X')$  1½ scores
- (c) Verify the De Morgan's laws with the help of truth table. 2½ scores

**Scoring Indicators:**

- (a) Steps for getting  $b=c$  1 score
- (b) Validity checking 1½ scores
- (c) Proof using truth table 2½ scores

(Total Score: 5)

**10**

# **Systems Analysis and Design**



## UNIT PLAN

Curriculum Objectives	Concepts/ Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Understand the concept of system.	Concept of system	Identification, Comparison, Communication	Discussion	Text Book Magazines	Note	1
Establish the importance System analysis and design.	Importance of System Analysis and Design.	Identification, Comparison Interpretation	Discussion Seminar	Text Book Magazines	Seminar Document	2
Understand the phases of system development life cycle.	Phases of system development life cycle.	Identification, Comparison Inference	Discussion	Text Book	Notes	1
Familiarize activities involved in system analysis and design.	Activities involved in system analysis and design.	Identification, Comparison	Discussion Seminar	Text Book, Web document	Seminar Document	8
Acquire knowledge in system implementation, maintenance and review .	System implementation, maintenance and review .	Identification, Comparison.	Discussion	Text Book Magazines	Notes	1

## Introduction

Students are familiar with the concept of a system, concept of office automation and use of different computer softwares. System analysis and design is a new concept that aims at developing a new system or modifying an existing system. It also helps to give the right direction to students in preparing and designing softwares for future needs. After the completion of this topic the learner is expected to have acquired ideas about various stages in developing a software project. These ideas are to be conveyed through group discussion and by preparing assignments.

## Curriculum Objectives

1. Understand the primary characteristics of a system.
2. Identify the need and importance of system analysis and design in real life.
3. Understand the phases of System Development Life Cycle (SDLC) and various activities involved in System Analysis and Design.
4. Familiarise the various aspects of system design.
5. Identify the major activities involved in system implementation, maintenance and review.

## Content Details

Definition of a system, Why system analysis and design, system development life cycle- system study, system analysis (Investigation and fact recording), Interviewing, Questionnaire, On-site Observation, System Design (Input, output, files, procedure), Implementation and maintenance.

## Pre requisites

- Knowledge of system concept and computer as a system.
- Idea about office automation.

## Learning Activities

### 10.1 Brainstorming on computer as a system and its characteristics (40 Mins.)

- 10.1.1 Problem: *Prepare a list of real life systems and mention their elements.*
- 10.1.2 Students list the systems and its elements in groups after discussion.  
(human system, school system, transportation system etc.)
- 10.1.3 The groups present their idea of a system and discuss in the class.
- 10.1.4 The teacher consolidates the discussions and presents the concept of a system.
- 10.1.5 Problem: *What are the general features of the systems mentioned?*
- 10.1.6 Students prepare a list of characteristics of systems in the groups and present in the class for further discussion.

10.1.7 Consolidation: The teacher consolidates the characteristics of system, students prepare notes on it.

**10.2 Discussion on concept of system analysis (40 Mins.)**

10.2.1 Teacher asks the students in groups to prepare a list of daily activities in their school office and to identify the requirements and problems if any.

10.2.2 Through a group discussion, and with the help of the teacher they list a series of activities such as school admission, rank list, preparation of admission, fee collections, preparation of reports, maintaining the details of staff and students etc.

10.2.3 The teacher asks the groups to identify the related problems and suggestions for improving the functioning of the office.

10.2.4 The groups present their findings in the class.

**Note:** The method of determining the problems as well as solution is termed system analysis. (identifying problem, examining the strength and weaknesses of the old system).

10.2.5 The teacher consolidates the discussion by introducing the concept of system analysis.

**10.3 Discussion on the concept of system design (40 Mins.)**

10.3.1 Teacher asks the groups to list out the steps and a method to improve the efficiency of the school office.

10.3.2 The groups discuss the matter and list out their findings.

10.3.3 Two or three students are selected at random and they present their findings in the class.

10.3.4 Finally the teacher consolidates the discussions and presentations by introducing the concept of system design.

**TIP:** The efficiency of school office can be improved by introducing computers, printers, barcode readers etc. and relevant softwares.

**10.4 Discussion on the activities involved in System Development Life Cycle (40 Mins.)**

10.4.1 Suppose you are in charge of introducing a new system design in your school office. List the set of activities to be performed for accomplishing this task.

10.4.2 Students in groups list their suggestions of activities.

10.4.3 Groups are asked to present their findings.

10.4.4 The teacher consolidates the activities as listed below and explains SDLC diagram.

Preliminary survey,	Feasibility study,	Investigation and fact recording,
System Analysis,	System Design,	Implementation,
Maintenance and Review.		

**10.5 Interview with an end user as a part of Preliminary Survey (90 Mins.)**

10.5.1 Problem: *A new system for the school office is to be developed. As a preliminary study, we need to consult the staff concerned. List out the important points to be noted for collecting information.*

10.5.2 Groups are asked to discuss on the list of queries based on issues they face and their suggested solutions.

10.5.3 Students in groups present their findings and discuss in the class.

**Tip:** Office staff should be informed at least one day before the interview date

10.5.4 The teacher consolidates the discussions.

10.5.5 Interview: An interview is conducted in the class with the office staff using the queries prepared and collect the responses

**Tip:** One or two office staff members may be invited

10.5.6 The students summarise the results obtained from the interview.

10.5.7 The groups present their summarised results in the class and discuss.

10.5.8 The teacher consolidates the issues and suggestions about the present system considering the volume of work, time taken to process data etc.

**10.6 Brainstorming on feasibility study (40 Mins.)**

10.6.1 Discussion is conducted on the summarised facts of the interview to fulfill the requirements of the user.

10.6.2 Students discuss in groups the issues faced and suggest a variety of solutions.

10.6.3 Each group presents its findings and other groups debate upon its feasibility.

10.6.4 The Teacher consolidates the discussions on the cost-benefit analysis and introduces the process known as feasibility study and its different types. The most feasible solution is selected for implementing.

**10.7 Discussion on investigation and fact recording (40 Mins.)**

10.7.1 Consider that the solution selected in the above activity is granted approval for implementation. What are the methods and techniques that can be adopted to determine the exact requirements of the user?

10.7.2 Students discuss in groups and prepares some suggestions.

10.7.3 Groups present their ideas in the class and discuss them.

10.7.4 Teacher consolidates the suggestions keeping in mind the different tools such as interviews, questionnaire, on-site observation etc.

10.7.5 Students are asked to prepare notes on each method with the help of text book and present it.

**10.8 General Discussion on the tools used in system analysis (40 Mins.)**

10.8.1 Problem: *The Teacher asks the groups to draw a diagram to represent the different stages of admission procedure for Plus-I class.*

**Tip:** Expected ideas may be like: *Admission Notification → Issuing Application forms → Receiving Application forms → Preparing ranklist → Interview → Selection and Admission*

10.8.2 Each group presents its views through diagrams.

10.8.3 Some students may be asked to present it on the black board and discuss it.

10.8.4 Consolidation: The Teacher consolidates the discussion through various tools such as Flow Chart, Data Flow Diagram, Decision tree, Decision Table, Data Dictionary etc.

10.8.5 Teacher asks the students to prepare an assignment on different tools.

10.8.6 At this stage the teacher introduces a document called 'Requirement Specification' that is prepared using the above tools and consists of specifications completely and accurately.

**10.9 Discussion on qualities and skills of system analyst (40 Mins.)**

10.9.1 Imagine that you are appointed as a system analyst in a company. From the above discussions, list out the desirable qualities that you need to become a good system analyst.

10.9.2 Each student individually lists out his/her views and suggestions.

10.9.3 A few students are asked to present their suggestions and others supplement it.

10.9.4 The teacher may consolidate the attributes and skills such as communication skill, business and technical knowledge etc.

10.9.5 Ask students to prepare notes with the help of text book or other materials.

**10.10 Discussion on activities involved in system design (40 Mins.)**

10.10.1 Problem: *Once the system specifications are determined, how will you design the system according to the user's needs?*

10.10.2 Ask the students to discuss their views in groups.

10.10.3 Groups present their views in the class and discuss.

**Tip:** The teacher is expected to guide the discussion highlighting the activities such as output design, input design, file design, program design etc.

10.10.4 Teacher consolidates the discussion and the students to prepare notes as home assignment.

**10.11 Discussion on system implementation, system maintenance and review (40 Mins.)**

- 10.11.1 Problem: *Once the system is designed, what are the subsequent steps for its implementation?*
- 10.11.2 Guide the groups to go through the activities such as program module preparation, file conversation, system change over, user training etc.
- 10.11.3 The teacher introduces the need of system maintenance, reviews and its importance.
- 10.11.4 Teacher consolidates the discussion and students prepare notes.

**Sample TE Questions**

- 1. An organization is facing difficulty to run their daily activities due to the incapability of the existing system. It decides to go for an alternative arrangement.
  - (a) What is the the solution of this problem? (1 score)
  - (b) What are the steps involved in identifying the exact nature of difficulty and requirements of the organization. (2 scores)
  - (c) Suggest the methods which will solve the problem in an effective way? (2 scores)

**Scoring Indicators :**

- (a) Designing a new system. 1 score
- (b) Fact finding tools and explanation 2 scores
- (c) System analysis and design -explanation 2 scores

- 2. Due to the lack of time and the large number of applicants for HSE admission, it is difficult to prepare an admission list manually. As a computer student the Principal seeks your help for introducing a new system design. What type of activities do you select before going to introducing a new system? (5 scores)

**Scoring Indicators :**

- System development life cycle 1 score
- Steps 2 scores
- Explanation 2 scores

**11**

# Concepts of DBMS



## UNIT PLAN

Curriculum Objectives	Concepts/Ideas	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Identify the advantages of DBMS over conventional record keeping system and recognize the components of DBMS	Conventional record keeping system, DBMS	Observation, Comparison, Inference.	Discussion	Text book, Magazines.	Notes	1
Understand the detailed structure of DBMS	Databases, DDL, DML, users.	Observation	Discussion, Drawing chart.	Text Book, Books on DBMS	Notes Chart	2
Understand the various levels of database abstraction and the concept of data independence.	Data abstraction, Physical, Logical and View level, Data independence	Observation Comparison Inference	Discussion	Text Book, Magazines	Notes	2
Familiarize the different data model.	Heirarchical, Network and Relational models,	Inference, Observation, Comparison	Discussion	Text Book, Magazines.	Notes, Assignment	2
Acquire the concept of RDBMS and its terminologies	Relation, Domain, Tuple, Attributes, Cardinality, Degree, Views, Keys	Observation, Comparison, Inference	Discussion	Text Book, Magazines.	Notes	1
Understand the various relational operations	Select, Project, Cartesian product, Union, Intersection, Set difference.	Observation Problem solving.	Discussion	Text Book, Magazines.	Notes, Assignment	2

## Introduction

This chapter aims at introducing the concept of DBMS as opposed to the conventional file keeping system. This covers different data models, levels of data abstraction and data independence in brief. Also the chapter discusses the relational model or the RDBMS. RDBMS is now being used in numerous applications and has established itself as a primary data model for commercial data processing applications.

## Curricular Objectives

1. Identify the advantages of DBMS over conventional record keeping system and recognise the components of DBMS through discussion and real life examples.
2. Understand the detailed structure of DBMS through illustration and prepare a chart.
3. Understand the various levels of database abstraction and the concept of data independence.
4. Familiarize the different data models through discussion, comparison etc. and form a table showing the features of each model.
5. Acquire the concept of RDBMS and its terminologies through general discussion and prepare a glossary for RDBMS.
6. Understand the various relational operations through problem solving, illustrations, discussion, and write statements for relational operations.

## Content Details

Data bases, advantages over conventional file system, DBMS and its structure, components of DBMS-Databases - DDL, DML, users, Database abstraction - Various levels, data independence, types of users of database, Data models - Relational model, network, heirarchical; Elementary terminologies of RDBMS, Relation, Domain, Tuple, Attributes, Cardinality, Degree, Views, Keys (Primary, Candidate, Alternate, Super, Foriegn) Relational operators - select, project, cartesian product, union, intersection, set difference.

## Pre requisites

- Idea of data and data processing concepts.
- General idea about the relationship between data and information.

## Learning Activities

### 11.1 Quiz on various aspects of data processing (40 Mins.)

- 11.1.1 Planning: Students are asked to prepare questions from chapter 1 (Principles of Data Processing) of standard XI Computer Science.

11.1.2 Questions prepared by each group are consolidated and finalised within the group and teacher adds questions if required.

11.1.3 Group quiz is conducted and the current level of the students in the topic is assessed. Remedial measures are taken if necessary to assure the awareness of the students about the terminologies and concepts related to data processing, its components and the advantages of electronic data processing over manual data processing.

**11.2 Discussion on centralised data processing (40 Mins.)**

11.2.1 Problem: *List down the data handled and processed in a school office and the information required for various purposes.*

11.2.2 Students individually identify the data processed and information to be generated, consolidate within the groups and present the findings.

11.2.3 Teacher consolidates the findings covering the following aspects:

- Data are to be collected and processed during admission, examination, fee collection etc.
- Details of staff related to teaching, salary etc. are to be maintained.
- Information reflecting the roll list, progress report, list of students having fee concession, amount collected as fee from various classes, list of fee defaulters etc. may be required.

11.2.4 Through an open discussion, teacher lists out the registers or files used for keeping the data and the activities carried out for processing them to generate desired information.

11.2.5 Through proper questions students are asked to identify the difficulties to be faced while performing the above activities.

11.2.6 The responses from the students are to be consolidated as follows:

- Related data are to be kept separately in suitable formats.
- The data as well as information should be maintained at one place, but they should be available to the concerned.
- While making data and information sharable, proper security must be ensured in order to avoid any misuse or malfunctioning.

**11.3 Discussion on the need for database and DBMS (40 Mins.)**

11.3.1 Teacher highlights the consolidation points of the above activity (11.2) and introduces the idea of database and DBMS.

11.3.2 Students may be asked to read the relevant topics from the text book or they may be provided with handouts containing the details of database and DBMS.

11.3.3 A discussion based on the contents provided is conducted within the group and the idea formulated is presented.

11.3.4 Teacher consolidates the features and advantages of database and DBMS; and students prepare notes.

**11.4 Discussion on the components of DBMS (40 Mins.)**

11.4.1 Problem: *Consider a familiar database management system from real-life (for example, telephone inquiry system of Telephone Department). Identify and list down various components (or, requirements for the working) of such a system.*

11.4.2 Through group discussions, students present their findings.

11.4.3 Teacher consolidates the findings as:

- Details of telephone subscribers
- Computer and software
- Users or operators

11.4.4 Teacher introduces the components of DBMS by correlating with the above findings as follows:

- Details of telephone subscribers  
to database
- Methods for creating files and storing the details  
to storage manager (DDL)
- Retrieving required information  
to query processor (DML)
- Users like public, operators and managers  
to users

11.4.5 Students prepare detailed notes and identify the components of similar systems.

**11.5 Discussion on database abstraction (20 Mins.)**

11.5.1 Teacher introduces some real life entities like electric switch board, dash board of car etc. and asks the students to list down the reason for the simplicity, usability and compactness of these.

11.5.2 The discussion is to be directed to get the idea of abstraction by raising suitable questions.

11.5.3 Students present their findings and teacher consolidates as follows:

The complex circuits are kept hidden by providing necessary switches, levers, knobs in switch boards and dashboards.

11.5.4 Students prepare notes on database abstraction.

**11.6 Discussion on levels of data abstraction and data independence (30 Mins.)**

11.6.1 A case is introduced and problems are posed as follows:

A computer system is made up of physical and logical components. The peripheral devices like monitor, system unit (CPU), keyboard, mouse etc. may be considered as the physical components and operating system and other software as logical components.

*How do these components interact each other? How are they dependent one another? What is the role of software? What are the different types users of the computer system?*

11.6.2 Through group discussion, students present their findings.

11.6.3 Teacher guides the discussion in such a way that the idea of levels of abstraction, independancy of peripheral units and the types of users working with computer etc. are elicited.

11.6.4 Teacher consolidates the discussion by correlating the concepts of levels of abstraction and data independence with the findings of the above case as follows:

- At the user's level, he/she is provided with switches, keys, buttons etc. to operate the machine.
- Software is in-charge of interpreting and activating the computer to work for the user.
- The electronic circuits associated with the hardware are responsible for carrying out the tasks instructed.
- The three levels of abstraction may be introduced by setting analogy to the above three cases.
- Similarly by setting analogies, the concepts of data independence and types of users may be introduced.
- The keyboard, mouse etc. are independent of the OS and the OS is independent to hardware circuitary of the machine.
- The different types of data base users may be introduced analogous to ordinary users, software developers, hardware engineers of computers.

11.6.5 Students prepare notes on levels of data abstraction, data independence and types of users with the help of suitable diagrams.

## **11.7 Discussion on different data models (40 Mins.)**

11.7.1 A task is given to the learners to collect the details like roll no., name, age, course and grade of Computer Science in an examination of the group members and organize them in a most appropriate format and present it.

11.7.2 Through group discussion, the data organised in an appropriate form is presented.

11.7.3 Teacher consolidates the format into tabular form and introduces the relational data model.

11.7.4 Students are provided with the concept of network data model and hierarchical data model through the text book or handout.

11.7.5 The idea formulated within the group is presented and teacher introduces the other two data models ask the students to convert the relation framed by them into network and hierarchical data models.

**11.8 Group discussion on RDBMS (40 Mins.)**

11.8.1 Students are provided with the terminologies related to RDBMS through text book or handout and ask them to identify and mark the terminologies in the 'student relation' prepared in activity 11.7.3.

11.8.2 During the presentation of groups, other groups and teacher interfere when and where required and the RDBMS terminologies are familiarised to the students.

11.8.3 More relations from real life are provided to identify the terminologies and evaluated in peer.

**11.9 Group discussion on various relational operations (40 Mins.)**

11.9.1 Using the activity similar to 11.8, the relational operations are introduced and evaluated by giving exercises.

**Sample TE Questions**

1. Gopal argued that using the conventional system of record keeping is better than using DBMS. Do you agree with him? Justify by giving proper reasons. (3 scores)

**Scoring Indicators:**

Features of conventional system	1 score
Features of DBMS	1 score
Comparative study with examples	1 score

2. When the rank list for Plus one admission is prepared manually, some records are found repeated twice or thrice. Can you suggest a good remedy to avoid such repetitions?

**Scoring Indicators:**

Concept of data redundancy	1 score
Remedy by using DBMS with proper explanation	1 score

3. What is relational algebra? Explain various relational operators with examples.

**Scoring Indicators:**

Definition of relational algebra	1 score
Explanation of relational operators with examples	4 scores

**12**

# Structured Query Language



## UNIT PLAN

Curriculum Objectives	Concepts/Contents	Process Skills	Learning Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Understand the different processing capabilities of SQL and its application	Database softwares like MS-Access or Oracle or SQL Server	Observation	Discussion	Text Book, Web documents	Notes	2
Identify the various data types available in SQL	SQL data types (Numeric, Character, Date, Time)	Observation, Comparison.	Discussion, Assignment, Lab work, Experimentation	Text Book, Web documents, Record Book	List, Notes, Assignments	1
Familiarize different commands used in SQL and identify their functions	SQL commands - DDL and DML	Observation Classification Comparison	Discussion, Lab work, Assignment	Text Book, Record Book	Query results, Assignments	7
Familiarize column constraints and clauses in SQL	Column constraints, Clauses used in SQL commands	Observation, Classification	Discussion, Illustration, Lab work, Assignment	Text Book, Record Book	Query results, Assignments	4
Familiarize different SQL functions used with SQL commands	Built-in functions in SQL - AVG, COUNT, MAX, MIN, SUM Creating tables, data entry, retrieval etc.	Observation, Classification, Comparison Realization	Discussion, Illustration, Assignment, Output Prediction, Lab work	Text Book, Record Book	Query results, Assignments	2

## **Introduction**

In the last two chapters students have learned about various aspects of Data Base Management System. To work with a Relational Database a suitable and powerful tool is required. Structured Query Language is such a tool using which one can effectively handle relational databases. It is an elegant and almost machine independent language which supports popular relational database technology. We generally follow ANSI (American National Standard Institute) standard for SQL. After learning this chapter the learner is expected to work with database operation like creation/ modification/ deletion of database files, entering data, editing, updating, retrieval, deletion etc. using suitable queries. Irrespective of the database software used, the learner should be able to use SQL in various database operations. The content can be transacted to the learners by way of discussion, illustration, lab work, assignments etc.

### **Assignment as CE item**

*An assignment can be given to acquire competency in handling database related problems using SQL queries, and it may be taken as a CE item.*

## **Curriculum Objectives**

1. Understand the different processing capabilities of SQL and its application through discussion and list them.
2. Identify the various data types available in SQL through group discussion and prepare a table describing the SQL data types.
3. Familiarize different commands used in SQL and write an assignment to apply them for table creation and manipulation for various real-life situations..
4. Familiarize column constraints and clauses in SQL through illustration, output prediction etc. and list out them.
5. Familiarize different SQL functions used with SQL commands through illustration and examples; learners solve related problems.

## **Content Details**

Processing capabilities; DDL, DML - retrieving, Manipulating, Updating tables; SQL data type - Numeric, Character, Date, Time, SQL commands- CREATE TABLE, CREATE VIEW, DROP TABLE, ALTER TABLE, SELECT, INSERT, DELETE, UPDATE, COLUMN constraints UNIQUE, PRIMARY KEY, NOT NULL, DEFAULT, CHECK; clauses with SQL commands - DISTINCT, INTO, FROM, WHERE, LIKE, GROUP BY, ORDER BY, HAVING, SET; SQL functions- AVG, COUNT, MAX, MIN, SUM.

## Pre requisites

- Awareness about the need for computer programming languages.
- Knowledge in Database Management Systems.
- Idea about different types of data.
- Thorough knowledge in Relational Operations.
- Concept of Table/Attribute/Tuple/Primary key.

## Learning Activities

### 12.1 Team Quiz to refresh the knowledge in DBMS (20 Mins.)

12.1.1 Planning: The whole class is divided into two teams, say Team A and Team B. Students are asked to prepare questions based on chapter 11 (concept of DBMS) Teacher explains the rules and regulations to conduct a team quiz.

Example: Each team can ask around 10 questions. Each learner should be given chance to ask questions and answer questions. Each and every student should prepare questions and questions should be presented in teams and select apt questions for the quiz.

*Tip: The quiz programme should not take more than 20 minutes. Both the teams should participate in the quiz without losing the real spirit.*

12.1.2 The teacher evaluates the quiz, and assesses the knowledge level of students before going to the details of the current topic.

*Tip: If needed teacher can take remedial measures so as to clarify certain points. Planning session may be conducted one day before the quiz.*

### 12.2 Discussion on working with a real life database (20 Mins.)

12.2.1 Problem: *Suppose you have an address book which contains the details of your friends. What are the activities you may do with the content of the book?*

12.2.2 Learners discuss in groups and list out their findings.

12.2.3 At random groups are allowed to present, other groups contribute.

*Tip: Commonly performed activities like adding a new address, taking an existing one, changing a phone number, need for arrangement in the alphabetic order etc. are expected from the learners. Teacher may take care to elicit such points from the student side.*

12.2.4 Consolidation: The teacher consolidates the discussions by highlighting the utility of an address book or similar databases, like writing a new address into the book, searching and picking out a particular address of your friend when needed, removing an address or phone number which is invalid, from the book etc.

**12.3 Discussion on functionality of SQL (40 Mins.)**

- 12.3.1 Problem: *You need to store the details of students in your class in a database file and manipulate it as and when needed. What are the capabilities or qualities expected from SQL as an efficient tool to manage a database?*
- 12.3.2 Students form groups and discuss among themselves. Let the learners come up with the need for creating a database file, entering data, editing data etc.
- 12.3.3 A group can be selected at random for the presentation of their findings. Other groups supplement their ideas and list the points.
- 12.3.4 Consolidation: Teacher consolidates the discussion by highlighting the DDL / DML features of SQL. Examples can be cited to clarify the above functions.
- 12.3.5 Students prepare short notes on the processing capabilities of SQL

**12.4 Discussion on various data types in SQL (40 Mins.)**

- 12.4.1 Problem : *Teacher writes the following data on the black board and asks the students to identify the categories*

SURESH KUMAR

T.C 11/1053

NEW STREET

EAST FORT

TRIVANDRUM

PIN 695001

10/04/2006 10:25:38

- 12.4.2 Students discuss the problem in groups and try to categorize the data items using their prior knowledge.
- 12.4.3 Teacher invites one or two groups to the present their findings.
- 12.4.4 Group representatives come to the stage write their classification on the board. Other groups contribute to complete the task.
- 12.4.5 Consolidation : Teacher consolidates the discussion and presentation by explaining the different data types available in SQL and helps the students to categorize the given data under suitable data types  
*Tip: Teacher is expected to explain the various standard data types available in SQL.*
- 12.4.6 A sentence can be given to the students as below  
“Rahul is a boy of age 17 born on 18/9/2004, 9:00:00 am at Ernakulam .....”
- 12.4.7 Students differentiate one data type from another and prepare a categorized list of given data items.

**12.5 Discussion on SQL commands (225 Mins.)**

- 12.5.1 Problem: *Design the structure of a table to store the details of students in the class. How do you create the table? Is it possible to modify its structure? How do you delete such a table, if needed?*
- 12.5.2 Students discuss in groups and write down the structure of a suitable table. Let them feel the need for a set of commands to do various operations on a table.
- 12.5.3 Teacher introduces the solution by introducing SQL-DDL commands for creating a table, modifying, deleting etc..
- 12.5.4 Teacher illustrates the commands using certain examples on the board.  
*Tip: Here the teacher is expected to explain all the commands like CREATE TABLE, ALTER TABLE and DROP TABLE with its syntax through suitable illustrations.*
- 12.5.5 Teacher poses another problem: How do we enter the details of students into the table which is created? Is it possible to correct a mistake in the name of a student after entering into the table? A student has transferred to some other school, can we remove his / her record from the table? etc.
- 12.5.6 Discussion is conducted and the learners may come out with some solutions without using SQL commands ( eg. enter records directly into the table using the respective database software) .Guide the discussions in such a way that the students feel the need for a set of SQL commands to perform the above tasks.
- 12.5.7 Consolidation: Teacher consolidates the discussions by explaining the SQL-DML commands to perform the tasks through illustrations.
- 12.5.8 The teacher gives a problem which include DDL/DML command operations as an assignment  
(This can be taken as a CE item for evaluation)  
*Tip: The teacher is supposed to explain the SQL commands like SELECT, INSERT, DELETE and UPDATE with its syntax through suitable illustrations. Class assignments can be given*

**12.6 Discussion on Views (40 Mins.)**

- 12.6.1 Problem: *The details of all the items in a Margin free shop are kept in a database table. The manager of the shop may ask several queries about the data and the computer operator has to supply it. But it is not always advisable to use the table which contains the data because a slight mistake may cause problems to the database. How can we solve the above problem?*

12.6.2 Students form groups and discuss. The need for a duplicate table for performing the query operations is to be elicited from the students. Students may come out with some solutions (taking copy the data table, taking printout of the table contents etc. are expected).

12.6.3 Consolidation: The teacher consolidates the discussions by introducing the concept of Virtual table or Views through illustrations. The need and working of views are explained.

12.6.4 Learners solve certain related problems and prepare notes.

*Tip: The teacher is expected to explain the CREATE VIEW command to create a view and give suitable problems to solve.*

## **12.7 Discussion on column constraints (90 Mins.)**

12.7.1 Problem: *How do we restrict certain data from not entering into the table? Is there any advantage?*

Eg: In the case of student table, same Register number should not be given for more than one student.

Age should not be above 20, etc.

12.7.2 Students discuss and list out the problems with faulty data entry. At random students are asked to present their ideas. Guide the discussions further in such way that the learners realize the need for imposing some pre-defined standard on data to be entered.

12.7.3 Consolidation: Teacher consolidates by demonstrating constraints through illustrations on the board.

*Tip: The teacher is expected to give suitable problems to the students so as to understand the applications of column constraints in SQL like UNIQUE, PRIMARY KEY, NOT NULL etc.*

12.7.4 Learners are given class assignments to familiarize the usage of different column constraints in SQL

## **12.8 Lab work on solving SQL problems (90 Mins.)**

12.8.1 A problem is given to the students.

Eg: The details of employees in company are to be stored in a data base file (table)

The details include, Employcode, Employname, Address, Basicpay, Dateof\_join etc.

Certain constraints can be given like Employcode should not repeat, Employname should not be blank, Basicpay should not be zero etc.

Sample records are given for data entry.

Certain queries can be given for retrieving the data based on conditions

Eg: How many employees are having basic pay more than 5000

How many employees are going to retire this year etc..

12.8.2 Let the learners design and write queries to create the table, and retrieve data.

12.8.3 Written queries are experimented at the lab.

12.8.4 Learners note down the observations and results of the queries in their Lab Diary.

## **12.9 Problem solving and group discussion on clauses in SQL (90 Mins)**

12.9.1 Problem: *Suppose the details of all the plus-2 students in our school are kept in a table named 'EXAM'. The table is having fields like RegNo, Name, Batch, Term, Sub1, Sub2, Sub3, Sub4, Sub5, Sub6, Total and Passed (logical field). The Principal wants some information as given below. How can you retrieve the details from the database using SQL commands?*

The queries can be like the following:

(i) Prepare a list with RegNo, Name and Total of all the students studying in the SCIENCE batch.

(ii) Prepare a list with the details of students studying in the COMMERCE batch arranged in the alphabetic order of their names.

(iii) Prepare a list of students with their names arranged in the alphabetical order.

12.9.2 Students discuss the problem in groups. They write the queries for displaying all the students in the school and the names of all the students in the unsorted order.

Let them feel the difficulty in filtering the records for a particular batch and the alphabetic arrangement of names.

12.9.3 Students at random are invited to present the issues or difficulties they face during solving the problem.

12.9.4 Teacher introduces the certain clauses which can be used along with SQL commands and their need through illustrations.

*Tip: Teacher is expected to explain the clauses like FROM, WHERE and ORDER BY used to solve the above problem in detail in this context. Also it is to be noted that the teacher explains the various operators like Relational, Boolean, Special operators etc. and give problems involving that.*

12.9.5 Let the learners write the required queries using the clauses explained by the teacher.

12.9.6 Consolidation: Teacher consolidates the discussions by familiarizing other clauses also through illustrations, class assignments etc.

*Tip: Teacher is expected introduce and explain all the clauses in SQL like DISTINCT, INTO, GROUP BY, SET, LIKE etc using suitable examples and illustrations.*

12.9.7 Learners are given class assignments in the form of problems which require application of various clauses used in SQL statements.

### **12.10 Discussion on SQL aggregate functions (90 Mins.)**

12.10.1 Problem: A company having more than 500 employees. The details are kept in a database table. How do you find the total salary paid to all the employees? How do you find the average salary of an employee?

12.10.2 The students form groups and discuss the problem

12.10.3 At random groups are allowed to present their finding and difficulties faced. Other groups contribute (manual methods to add the individual salaries of all the employees, calculate the average etc. are expected from student side).

12.10.4 Consolidation: teacher consolidates the discussions by introducing the built in aggregate functions of SQL to simplify the types of tasks through illustrations.

12.10.5 Teacher gives similar problems and asks the learners to write solutions which involve aggregate functions.

12.10.6 Learners prepare notes or tables showing the syntax and usage of the aggregate functions.

12.10.7 Learners are given class assignments in the form of problems which require application of various aggregate functions in SQL.

*Tip: The teachers are supposed to explain the aggregate functions in SQL like AVG, COUNT, MAX, MIN and SUM using suitable examples*

### **Sample TE Questions**

1. As a part of the Project work, the details of products in a shop are to be stored in a database table. The following tasks are to be done
  - i. Make a table with fields ProdID, ProdName, UP, Stock
  - ii. Enter records
  - iii. Add one more field to the table as Remarks
  - iv. Remove a record.

- (a) Name a commonly used database software using which we can store the product details. (1 score)
- (b) Write suitable SQL commands to perform the above tasks (1 score)
- (c) Write the syntax of the above commands using example (4 scores)

**Scoring Indicators:**

- (a) MS-ACCESS, Oracle, SQL-Server etc 1 score
- (b) Correct commands ( CREATE, INSERT INTO, ALTER TABLE, DELETE ) 1 score
- (c) Correct syntax of the above commands 4 scores

2. The details of applicants of plus1 admission in a school are to be entered in a database table. The following are some conventions adopted during the design of the table
- i. Each applicant should be given a unique number.
  - ii. The age of the applicant should be less than or equal to 20
  - iii. The name should not be omitted
  - iv. The content for the SEX field should appear as 'Male' and allows editing.
- (a) How do we impose some validations while data entry? (1 score)
- (b) Identify the suitable keywords in SQL and write. (2 scores)
- (c) Explain shortly on each of the above used keywords (2 scores)

**Scoring Indicators:**

- (a) Column constraints 1 score
- (b) Identification of correct constraint 2 scores
- (c) Correct explanation of the usage of each constraint 2 scores

3. The following are some situations where a SQL command or a clause or an operator is to be used
- i. From the STUDENT table list out the Names of all the students starting with the letter 'A'
  - ii. List the Names of students in the above table studying in the SCIENCE and COMMERCE batches.
  - iii. From the EMPLOY table display the Empcode and Name of all the employees having Basicpay in the range of 5000 to 8000
  - iv. Find the total number of employees working in the ACCOUNTS department in the above table.
  - v. Delete a View having name 'Sample'

(a) Identify and write the command or clause or operator appropriate for the above cases (2½ scores)

(b) Write the SQL query statement for each of the above (2½ scores)

**Scoring Indicators:**

(a) LIKE, IN or OR, BETWEEN, COUNT, DROP VIEW  $\frac{1}{2} \times 5 = 2\frac{1}{2}$  scores

(b) Correct statement  $\frac{1}{2} \times 5 = 2\frac{1}{2}$  scores

**13**

**Data  
Communications**



## UNIT PLAN

Curriculum Objectives	Concepts/Contents	Process Skills	Activities	Learning Aids/ Materials	Evaluation/ Products	Time (Periods)
Understand the concept of data communication and compare different modes of transmission	Networking and data communication, Serial, Parallel, Digital, Analog, Synchronous, Asynchronous	Observation, Comparison	Quiz, Discussion etc.	Text Books, Web documents etc.	Notes, Chart	2
Understand the various types of communication	Simplex, Half duplex, Full duplex	Comparison, Classification	Discussion, Illustration etc.	Text Books, Web documents etc.	Notes, List	3
Familiarize various communication channels	Twisted pair, Coaxial, Optical cables, radiowaves, microwaves and satellites	Identification, Classification, Comparison	Discussion, Exhibition etc.	Text Books, Web documents, IT magazines etc.	Notes, Chart	3
Acquire knowledge about protocols and understand the various types of protocols	TCP/IP, FTP, HTTP	Observation, Classification, Comparison	Group discussion, Panel discussion etc.	Text Books, Magazines.	Notes, List	1
Familiarize different data access methods	CSMA/CD, Token Passing	Visualization, Classification, Comparison	Discussion, Demonstration, Role Play etc.	Text Books, Magazines, Web Resources.	Notes, Chart, List	2
Understand the features and applications of various data communication devices	Bridge, Router, Repeater, Gateway, Backbone	Observation, Classification, Comparison	Discussion, Showcasing etc.	Text Books, Magazines, Web documents etc.	Notes, List, Charts	3
Familiarize the various data communication terminologies related to Internet	WWW, Websites, Webpages, HTML, URL	Observation, Realization	Lab demonstration	Computer magazines, Text Books etc.	Notes	2

## **Introduction**

Learners are already familiar with computer networks. One of the important applications of networking is data communication. Data Communication technology is improving day by day with the help of rapidly evolving computer technology. Even though the process of communication originated from non-verbal / verbal communication, now it is advanced to bluetooth wireless technology.

At the end of this chapter, the learner is expected to be well versed with modes and types of communication, communication protocols and devices, Internet, mobile technology etc. The whole topic can be transacted through classroom activities like discussion, quiz, demonstration, panel discussion, showcasing etc.

## **Curriculum Objectives**

1. Understand the concept of data communication and compare different modes of transmission through discussion, illustration etc. and also prepare notes by comparing different modes.
2. Understand the various types of communication through discussion and illustration; prepare charts with proper examples.
3. Familiarize various communication channels through observation, comparison, diagrammatic representation etc. and prepare detailed notes.
4. Acquire knowledge about protocols and understand the various types of protocols through discussion.
5. Familiarize different data access methods through discussion, role play etc. and prepare notes on each of the access methods.
6. Understand the features and applications of various data communication devices through general discussion and prepare notes on each device.
7. Familiarize the various data communication terminologies related to Internet through observation, experimentation etc. and prepare notes.

## **Content Details**

Concept of Data Communication, Modes of Transmission -Digital Vs Analog, Serial Vs Parallel, Synchronous Vs Asynchronous; Type of Communication - simplex, Half Duplex, Full Duplex; Modem; Communication channels - Twisted pair cables, Coaxial cables, Optical Fiber, Radio Waves, Satellites; Communication Protocols- FTP, HTTP, TCP/IP; Access Methods - CSMA/CD and Token Passing; Data Communication Devices- Repeater, Bridge, Router, Gateway, Back one Network; Data Communication Terminologies in Internet – WWW, Website, Web page, HTML, URL.

## Pre requisites

- Overall idea about computer networking
- Knowledge of terms like Data, Information, Signal etc.
- General awareness about Internet, e-mail
- Idea about traditional methods of communication.

## Learning Activities

### 13.1 Quiz to test prior knowledge on networking (20 Mins.)

- 13.1.1 Teacher places a small box on the table, which contains some folded bits of paper. Each paper bit has a keyword written on it.  
eg: Data, Information, Network, Internet, E-mail etc.
- 13.1.2 Teacher invites the learners in random order to come and pickup a paper bit.
- 13.1.3 The learner explains / defines the keyword written on that paper.  
Chances can be given to other students to complete the explanation if one fails.  
*Tip : Teacher can elaborate certain points if really needed*
- 13.1.4 Teacher evaluates the performance / responses of the students and assesses the current knowledge level in the topic to be discussed.

### 13.2 Discussion on the concept of data communication (25 Mins.)

- 13.2.1 Problem: *Teacher calls one of the students, say Raju, sitting in the first row and tells him a sentence. Eg: "Tomorrow is a Holiday"*
- 13.2.2 The learner is asked to communicate this message to his friend, say Anil, sitting in the last row.  
The problem is opened to all the students in the class
- 13.2.3 The students list out various methods for communicating such a message from one point to another. Each one has to identify the sender, target and medium of communication.
- 13.2.4 One or two students present their ideas. Teacher points out the merits and demerits.
- 13.2.5 Consolidation: Teacher consolidates the discussions, by explaining the terms like source, target, channel or medium of communication. Related to the above problem the source and target are two students and the medium may be by air (verbal) or by way of jesters (non-verbal) or by sight (showing a banner) or by passing the message from pupil to pupil or any other method.

### 13.3 Role-play to illustrate the modes of transmission (40 Mins.)

- 13.3.1 Teacher asks six students in the class to come up to the platform. Each of these students has to act as a data bit. The problem is, these students have to move

from one end of the class room to the other end. Teacher wants all the students to watch the play and comment on it.

First of all, teacher asks the learners to move one after the other in a queue fashion (serial).

As another method, teacher allows the learners to move in two or three rows (parallel).

13.3.2 Learners are asked to discuss the two modes of data transmission in terms of their speed, time taken, merits or demerits etc.

13.3.3 Each group presents its opinion about the tried out modes.

13.3.4 Consolidation: Teacher consolidates the discussions and presentations by introducing the Serial and Parallel modes of communication.

13.3.5 Teacher introduces and familiarizes the other modes of transmission like Synchronous / Asynchronous and Digital / Analog.

*Tip :The teacher is expected to explain the different modes of transmission in detail with suitable illustrations*

13.3.6 The teacher illustrates the different modes of transmission with the help of diagrams.

13.3.7 Students are asked to prepare notes and comparison charts showing the different modes of transmission.

#### **13.4 Discussion on various types of communication (90 Mins.)**

13.4.1. Problem: *The teacher presents three cases in front of the learners.*

*The first case is a person watching a Television programme.*

*The second case is a police man talking over wireless equipment (Walkie-talkie).*

*The third case is a person conducting conversation over telephone or mobile phone.*

*(This can be shown with the help of charts or diagrams on the board etc.)*

13.4.2 The teacher asks the students to compare the above three cases and comment on each, on account of their nature, properties, advantages, disadvantages etc., in groups through discussions.

13.4.3 Let the group leaders present their findings and try out differentiating the above three cases.

13.4.4 Consolidation: Teacher consolidates the discussions and presentations by introducing the three types of communication.

*Tip: The teacher is expected to explain the types of communications like Simplex, Half-Duplex and Full duplex. He / She should correlate these with the cases he/she presented earlier.*

13.4.5 The learners are asked to prepare notes and draw block diagram on each type of communication and present in the class.

**13.5 Discussion and chart making on communication path (40 Mins.)**

13.5.1 Problem: *How do the data travel from the sender computer to the receiver computer? Is there any need for converting data from one form to another? eg. An e-mail communication*

13.5.2 The learners discuss in groups and present their beliefs on the path of data from source to target.

13.5.3 Consolidation: Teacher consolidates the discussions by introducing the device, Modem, and its functions through illustration.

13.5.4 Learners are asked to prepare a chart (as a group activity) in the class room itself which shows the diagrammatic representation of the process and path of communication including devices like Computers, Modem etc.

*Tip: Teacher can ask the groups to bring chart paper beforehand. Make sure that the whole work takes not more than 45 minutes.*

**13.6 Discussion on different types of communication media (135 Mins.)**

13.6.1 Problem: *What are the different types of communication cables you are familiar with in your daily life? List out them.*

13.6.2 Students discuss and list out the various types of cables they are familiar with.

*Tip : The ordinary wires used in telephone, flat and round cables used for cable-TV etc are expected here. Teacher should guide the learners to come up with such items from their day to day life.*

13.6.3 Consolidation: Teacher consolidates the discussion by explaining the teachers of guided and unguided media for communication.

*Tip: The teacher is expected to explain different types of communication channels in detail here. Home assignments can be given. Let each group present more details on each channel.*

13.6.4 Teacher conducts a demonstration of different types of communication channels like Twisted pair, Co-axial cables, Fiber optic cables etc.

*Tip: Teacher is expected take steps to collect the samples of different communication channels through reliable sources.*

13.6.5. Students observe and notedown points on each medium comparing their merits and demerits.

**Supplementary Activity**

**13.6(a) Album preparation and exhibition (40 Mins.)**

Planning: Each communication channel is assigned to each batch. An album is to be arranged. It can be a big sized note book or a bundle of papers filed or something

similar which serve the purpose. The learners are asked to collect maximum details on each item assigned to them. Sources may be specified by the facilitator such as library reference, IT magazines, Telecommunication journals, web browsing, text books etc. Time may be pre-defined and measures to be taken for participation of all the students. Students can consult the teacher and review the matter before the preparation of album

- Each group collects, classifies and prepare detailed description with diagrams or samples.
- An album is developed in the class room by the learners. Each group contribute to the work.
- The album is completed exhibited in the class. Each group is allowed to present their work, others listen and clarify doubts.
- Learners complete their notes by referring to the album.

**13.7 Discussion on communication protocols (40 Mins.)**

13.7.1 Problem: *What are the traffic rules that you obey when walk through a road? Are they really needed? Why?*

13.7.2 Students form group and discuss the need for rules on road.

13.7.3 Ask the learners to suggest the logic used in formulating traffic rules.

13.7.4 Teacher guides the students to correlate this with a set of rules and regulations in the data communication system.

13.7.5. Consolidation: Teacher consolidates the discussions by explaining the different types of communication protocols.

*Tip: The teacher is expected to define the term protocol and explain the different communication protocols like TCP/IP, FTP etc.*

13.7.6 Students prepare a comparison table on different types of protocols.

**13.8 Roleplay to simulate Access Methods (45 Mins.)**

13.8.1 Planning : Teacher invites 8 students to come on to the platform and explains the role to be done by each participant. Each student plays the role of a node in a network. Other students in the class must observe the play and comment on it.

13.8.2 Students form a circle as a simulated human network. Teacher gives two small paper boxes or a paper packets to two different students in the network. The paper box/ packet is supposed to be the data to be sent.

13.8.3 Students pass the boxes in opposite directions so as to reach a particular destination. Before reaching the respective target, the boxes meet at some point and faces a collision. Teacher poses a question to the class. How can we avoid such a situation of data collision ?

Tip: The play can be repeated if needed.

13.8.4 Students form groups and discuss, list out solutions. Best possible solution is discussed in groups and consolidated.

13.8.5 One or two students present their findings in the class, and the play is conducted once again as per the solution.

13.8.6 Teacher allow the students to highlight the merits and demerits

Tip: One or more methods of sending ata may be expected from the students.

13.8.7 As a final solution teacher following the most appropriate techniques accessing data in a network.

Technique 1: Suppose student at position 1 wants to send the paper box tothe student at position 7. Before sending the box, the student at position 1 has to listen and make sure that no other box is passing the network.

Technique 2: Apart from the paper bpxes an additional red coloured small box is required.This red coloured box is acting as a token and it is to be circulated always in the network. When a student wants to send his paper box to some destination, he must first collect the red box from the network, and then send his box, thus avoiding the chances of collission.

13.8.8 Students perform the role play as per the above two techniques and realise the alogrithm.

13.8.9 Teacher consolidates the duscussion and simulations by presenting the two data access methods in a network as CSMA/ CD and token passing.

13.8.10 Students note down points by finalizing the comparision table

13.8.11 Students prepare notes on each method.

13.8.12 A class assignment is given to prepare a chart showing the two access methods using block diagrams and comparison table

### **13.9 Discussion on access methods (45 Mins.)**

13.9.1 As a continuation to the activity the teacher initiate a group discussion about access methods in the class with the help of hand outs or text books

Problem : *Compar the two access methods namely CSMA/ CD and Token passing then list down the features of each.*

13.9.2 Students discuss the two different methods in groups and list out the findings

13.9.3 One or two groups present relevant product in the class, relevant points ar noted on the balck board.

13.9.4 Teacher consolidates the discussion by finalising the comparison table.

13.9.5 Students prepare notes on each access methods.

**13.10 Discussion on communication devices (90 Mins.)**

13.10.1 Problem: *We send data from a computer placed in India to a computer in USA. We need to connect one network to another. How is this possible? We know that when the cable-TV connection is extended to many users the signal strength becomes weak and the last ones get poor reception. What is the remedy for this?*

*Tip: Discussion can be started by posing certain questions like this.*

13.10.2 Students form groups and discuss. Teacher the guide the students to think and list out solutions. Let some of the students present their ideas and beliefs.

13.10.3 Consolidation: Teacher consolidates the discussions by presenting the details on communication devices.

Features of each device are discussed and the learners are asked to note down points.

*Tip: The teacher is expected to explain the devices like Bridge, Router, Gateway, and Repeater etc. with their need and application in detail*

13.10.4 Let the learners to clarify their doubts if any and they prepare separate short notes on each devices.

**13.11 Showcasing still models of communication devices (40 Mins.)**

13.11.1 Planning Session :

The whole class is divided into 4 to 5 groups and each communication device assigned to each batch. A still model of a device is to be developed and presented with appropriate explanations. Thermocol or cardboard can be used to make the models. The teacher can allow the students to do this work at home. A time limit is to be given.

*Tip: Make sure that the students are not taking much time on the beautification of the models.*

13.11.2 Each batch develops the model of the devices assigned to them with a detailed description in the form a chart.

13.11.3 Each group is allowed to observe and understand the details collected by other groups.

13.11.4 Each batch is allowed to prepare notes on the model developed by the other batches.

13.11.5 Teacher evaluates the performance of each group and gives credit accordingly.

*Tip: Teacher can ask the learners to display their models in the class itself for some days so that the one can clearly distinguish each device from other.*

**13.12 Panel discussion and note making (40 Mins.)**

13.12.1 Planning: Students are asked to prepare questions based on topics, which need more explanations in the closing chapter.

13.12.2 The questions are pooled and categorized in groups. The selected questions are to be kept ready well in advance.

*Tip: The date and time of conducting the panel discussion can be announced earlier. The computer science, computer application teachers of the same school or from nearby schools or experts from outside can be the panelists.*

13.12.3 Students ask selected questions to the expert panel members and note down points.

13.12.4 Students prepare detailed notes on each and present in the class.

**13.13 Discussion on Internet terminologies (40 Mins.)**

13.13.1 Problem: *What is Internet? Can you say some of the Internet services? What is a homepage?*

13.13.2 A group discussion can be started by posing such problems. Students discuss about Internet and related terms in groups.

13.13.3 The discussion can be guided in such way that the teacher can elicit the terms and concepts related to Internet from the student side to the maximum.

13.13.4 Teacher ask some of the students from different groups to speak out, others contribute.

13.13.5 Consolidation: Teacher consolidates the discussions by familiarizing the terminologies Internet like WWW, Website, Webpage, Web browser, Homepage, URL etc.

13.13.6 Students prepare notes on each item.

***Supplementary Activity***

**13.13 Live demonstration of browsing Internet (40 Mins.)**

13.13.1 After imparting overall knowledge about Internet the students can be directed to the lab for a live demo of web browsing.

13.13.2 Task: The students are asked to collect some details about ‘TSUNAMI’ from the Internet and prepare a note.

*Tip: Hints can be given to the students about the idea of search engines, names of popular search engines or teacher can demonstrate as per the situation demands.*

13.13.3 Learners search the Internet using Google, Yahoo etc search engines and collect details.

13.13.4 The teacher ask the learners to identify the Homepage, URL, WWW etc while browsing.

13.13.5 Learners write their findings in the observation book and prepare notes on required items.

*Tip: Hints can be given to the students about the idea of search engines, names of popular search engines or teacher can demonstrate as per the situation demands.*

13.13.6 Learners search the Internet using Google, Yahoo etc search engines and collect details.

13.13.7 The teacher ask the learners to identify the Homepage, URL, WWW etc while browsing.

13.13.8 Learners write their findings in their observation book and prepare notes on required items.

### Sample TE Questions

1. In a city the following traffic rules are implemented for reducing the traffic block.  
*In road A, the vehicles can go from East to West only; in road B, the vehicles can pass only in one direction at a time; in road C, the vehicles are free to move in both the directions at the same time.*
- (a) How do you correlate the above type of traffic system with that of data communication?  
(b) Explain each with examples and block diagrams. (2 + 3 = 5)

#### **Scoring Indicators:**

- (a) Types of communication , Simplex, Half duplex and Full duplex 2 scores  
(b) Explanation of the above three types with block diagram 3 scores
2. Telecommunication department has to setup some network connections in a metro city. The following are the tasks to be done
- Connect Server computer in the ground floor of a building to the Nodes placed in the first floor.
  - Send huge volume of data from the central exchange to local exchange situated about 5 kms away.
  - Send data from the central office to the zonal office situated about 50 Kms away.
  - Send data to the network station located abroad.

- (a) Suggest the most suitable communication channel that can be used for each of the above situations. (1 score)
- (b) Justify your selection by listing out the features of each. (4 scores)

**Scoring Indicators:**

- (a) Coaxial cable, Optical fiber, Microwave and Sattellite communication respectively  
1 score
- (b) Listing out features of each media  
4 scores

3. Analyse the following situations related to data communication

- i. The data from a computer is to be send through telephone wires
- ii. While sending data to distant places the signal strength becomes weak
- iii. Connect a LAN with Internet
- iv. Browse a website while travelling in a train or bus

- (a) Suggest the most appropriate communication device that can be used for the above purposes (1 score)
- (b) Justify your selection with proper explanations (4 scores)

**Scoring Indicators:**

- (a) (i) Modem, (ii) Repeater, (iii) Gateway (iv) Mobile phone with sufficient technology  
1 score
- (b) Explanation of each of the above items  
4 scores

4. The following are certain terms related to data communication:

Gateway, Computer, Router, Modem, Internet, Cable

- (a) Which of the above items is not a device? (1 score)
- (b) Arrange the above given devices in the most suitable way so that the e-mail goes from sender to the receiver. (2 scores)
- (c) Simply describe the reason for such a connectivity (2 scores)

**Scoring Indicators:**

- (a) Internet  
1 score
- (b) Computer-Modem-Cable-Router-Gateway-Internet  
2 scores
- (c) Digital data from the computer is converted analog and send through the cable, go via router and gateway to get into the Internet and follows the similar path to reach the destination.  
2 scores

# References

1. *E. Balagurusami, Object Oriented Programming with C++*  
(Tata McGraw Hill Publishing Company)
2. *Robert Lafore, Object Oriented Programming in Turbo C++*  
(Galgotia Publications)
3. *Yashwant Kanetkar, Let Us C++*  
(BPB Publications)
4. *Eric Nagler, Learning C++ - A Hands-on Approach*  
(Jaico Books Company)
5. *John Hubbard, Programming with C++ - Schaum's Series*  
(McGraw Hill Publishers)
6. *Stanley B. Lippman & Josee Lajoie, C++ Premier*  
(Addison Wesley Longman Inc., Singapore)
7. *K. R. Venugopal, Raj Kumar & T. Ravishankar, Mastering C++*  
(Tata McGraw Hill Publishing Company)
8. *Albert Paul Malvino & Donald P. Leach, Digital Principles and Applications*  
(McGraw Hill Publishers)
9. *Kendall & Kendall, Systems Analysis and Design*  
(Printice Hall India)
10. *Henry F. Korth, Introduction to DBMS*  
(McGraw Hill Publishers)
11. **Understanding SQL**  
(BPB Publications)
12. *Achyut S. Gobble, Data Communication and Networks*  
(Tata McGraw Hill Publishing Company)