

**Shiksha Mandal's  
Jankidevi Bajaj College of Science, Wardha**

**SCHEME and SYLLABUS FOR  
UNDERGRADUATE PROGRAMME (B. Sc.) IN COMPUTER SCIENCE**

**Programme Objective:**

There are two main objectives to the B.Sc. Computer Science Programme.

- a) The primary goal of the degree program in Computer Science is to provide students the foundations for the future work and careers in computation-based courses. These foundations support both a successful career path in computing as well as provide appropriate qualifications for further degree work in computation related disciplines. Our degree emphasizes development of analytical skills, acquisition of knowledge and understanding of systems, languages and tools required for effective computation-based problem solving.
- b) A Computer Science major will provide you with knowledge of programming, operating systems, compiler design and principles of programming language. These skills will prepare graduates to enter technological fields such as systems programming, technical support, research and teaching.

On completion of the B.Sc Computer Science Programme, the students:

- a) Will have sound knowledge of the Programming languages such as C, C++, Java, VB and SQL. Also they will have knowledge about Information Technology, Operating System, System Analysis and Design and Development.
- b) Can participate in off-campus internships, independent study projects, part-time computer employment, lectures by guest speakers, Society of Computing Students programming and on campus employment as lab assistants, lab supervisors and Helpdesk assistants.
- c) Be in a position to develop industrial and entrepreneur applications.

**PROPOSED SYLLABUS FOR B.Sc. COMPUTER SCIENCE**  
**SEMESTER I, II & III**

**Aim of the Course**

To understand the basics of computer hardware, software, network and algorithms, flowcharts and programming and also to provide sound knowledge of object oriented programming concepts and their understanding, development, working process of software.

**Objectives of the Course**

1. To learn the basics of Computer components,
2. To learn the basics of Software and Hardware,
3. To learn the Algorithms and flowcharts,
4. To learn basics of programming language that is C.
5. To learn the concepts of Object Oriented Programming.
6. To learn about networks and network topologies.

## Scheme of Teaching, Evaluation and Examination

### Credit Based System (CBS)

Name of the Department: **Computer Science**

No. of Papers: **One**

Max. Marks per Paper: **100** (Internal Assessment: **20**, End Semester Exam: **80**)

No. of Practical: **One**

Max. Marks for Pract.: **50** (Internal Assessment: **20**, End Semester Exam: **30**)

No. of Lectures, Tutorials and Practicals (per week):

B. Sc. I : 6Th, 1Tu/Batch, 6Pr/Batch

B. Sc. II & III : 6Th, 2Tu/Batch, 6Pr/Batch

Th.: Theory, Tu.: Tutorial, Pr.: Practical

#### Scheme:

Subject	Paper	Internal Assessment (Max. Marks)	End Semester Exam. (Max. Marks)	Total Marks	Minimum Passing Marks	Credits	No. of Hours	No. of Lectures
Computer Science	Theory	20	80	100	40	4	60	80
	Practical	20	30	50	20	2	60	80

#### Internal Assessment:

##### A. For Theory:

Sr. No	Evaluation type	Marks
1	Assignments	8
2	Class Test(s)	6
3	Instructional deliveries (case studies/ seminars/ presentation) Overall conduct as a responsible learner, manners, sincerity, skill in articulation, leadership qualities demonstrated through organizing, co-curricular activities, Active participation in routine class, etc.	6

##### B. For Practical:

Sr. No	Evaluation type	Marks
1	Additional Practicals, Lab based assignments, Projects, Survey, Case study, etc.	16
2	Viva	4

**Grade Awards:**

Seven point rating scale is used for the evaluation of the performance of the students to provide letter grade for each course.

Range of percentage of Marks obtained	Grade Points	Grade	Remark (Not to be displayed on the transcripts)
90 - 100	10	O	Outstanding
80 – 89.99	9	A+	Excellent
70 – 79.99	8	A	Very Good
60 – 69.99	7	B+	Good
55 – 59.99	6	B	Fair
50 - 54.99	5	C+	Average
40 - 49.99	4	C	Below Average
Below 40	0	F	Fail
Absent	0	AB	Fail

*(Note: In case, the marks scored by the student fall in multiple grades, higher grade will be considered in the interest of the student)*

**Computation of Semester Grade Point Average (SGPA) for each semester:**

$$SGPA = \frac{\sum \text{Credits in the subject} \times \text{Grade Points obtained}}{\text{Total Credits in the Semester}}$$

**Computation of Cumulative Grade Point Average (CGPA) after completion of Program:**

Sem-I		Sem-II		Sem-III		Sem-IV		Sem-V		Sem-VI	
Credits	SGPA	Credits	SGPA	Credits	SGPA	Credits	SGPA	Credits	SGPA	Credits	SGPA

$$CGPA = \frac{\sum (\text{Credits} \times \text{SGPA})}{\text{Total Credits in the Program}}$$

**B.Sc. SEMESTER I**  
**PROGRAMMING IN 'C' and**  
**FUNDAMENTALS OF INFORMATION TECHNOLOGY**  
**Course Outline**

**Unit I**

Programming Structure: Sequence, Selection, Iteration and Modular. Problem Solving techniques: Development Tools: Algorithm, Flowcharts and Pseudo code (Definition and its characteristics) Developing Algorithm and Drawing flowcharts.

C character set, Tokens, Identifiers, Keywords, Variables, Data types, Qualifiers. Operators and Expressions: Arithmetic, Relational, Logical, Bit-Wise, Increment, Decrement, Conditional and Special operators. typedef, Type Conversion, Constants, Declaring Symbolic Constants, Character Strings, Enumerated Data Types, Operator Precedence and Associativity. Library functions: Maths, string handling Functions.

**Unit II**

Control Structure: Compound statement, Selection statement: if, if-else, nested if, switch. Iteration statement: for, while, do..while, Nested loops, Jump statement: break, continue, goto. (Special emphasis on problem solving).

Arrays: Need, Types: Single and Two Dimensional Array. Strings: Strings Manipulation, Arrays of Strings, Evaluation order

Function: Function Components, Return Data type, Parameter Passing, Return by Reference, Default Arguments, Recursive Functions, Arrays with Functions, Storage Classes.(Special emphasis on problem solving).

**Unit III**

Structure: Declaration, Definition, Accessing structure members, Initialization, Nesting of Structures. Union: Unions, Differences between Structure and Union

**Pointer:** Introduction, Address Operator (&), Pointer variables, void pointers, Pointer Arithmetic, Pointers to Pointers.

File handling: Hierarchy of File Stream Classes, Opening & closing a file, Testing for errors, File Modes, File pointers and their manipulations, Sequential Access, Random Access, Command Line arguments.

**Graphics:** Initializing graphics, Drawing basic shapes, Graphics Color system, Types of pens and brushes.

#### **Unit IV**

Basic Components of Digital Computers: Block Diagram. CPU: Functions of Each Unit: Primary Memory, ALU and CU, Instruction format. Bus: Data, Control and Address Bus. Number Systems: Binary, Octal, Decimal, Hexadecimal, Their Conversions, Binary Arithmetic. ASCII, BCD, EBCDIC.

Language Evolution: Generation of Languages: Machine, Assembly, High Level Languages. Characteristics of Good Language Translators: Compiler, Interpreter and Assembler. Source and object Program.

#### **Unit V**

Memory: Static & dynamic, RAM, ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blue Ray.

Input Devices: Keyboard, Mouse, Light Pen, Touch Screen, Voice Input, MICR, OCR, OMR, Barcode Reader and Flatbed Scanner. Output Devices: VDU, Printers: Dot Matrix, Laser and Inkjet. Plotters: Drum, Flat-Bed and Inkjet.

#### **Unit VI**

Network: Network terminology, Topologies: Linear, Circular, Tree and Mesh. Types of Networks: LAN, WAN, MAN. Repeaters, Bridge, Routers, Brouters and Gateway. Modem for Communication between PC's, Wi-Fi network, Introduction of Bluetooth and Infrared devices. Network protocols. Architecture: Peer-to-Peer, Client/Server.

**B.Sc. Semester I**  
**COMPUTER SCIENCE PRACTICALS**

**Section A**

1. Program to compute Fibonacci series.
2. Program to find if a given number is prime or not.
3. Program to accept number and display it in words.
4. Program to find sum of digits of any entered no.
5. Program to reverse the digit.
6. Program to find frequency of occurrence of a given number from an array of N elements.
7. Program to reverse an array.
8. Program to Insert an element in one dimensional array at a given position.
9. Program to Delete an element from one dimensional array.
10. Program to Arrange string data (name of students) in alphabetical order using bubble sort.
11. Program to search the element in an array of N elements using Linear search method & Binary search method.
12. Program to
  - a) Multiply two dimensional Array's (3X3 matrix)
  - b) Find largest element in two dimensional Array (3X3matrix)
13. Program to
  - a) Check if given string is Palindrome or not
  - b) Calculate number of blanks, vowels and words
14. Program to compute:
  - a) Cosine series:  $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$
  - b) Sine series:  $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$
15. Program to find factorial of a number using recursive function.
16. Program using function to find sum of two numbers
  - a) With no argument & no return values,
  - b) With argument & no return values,
  - c) With argument & return values.
17. Program to demonstrate passing structure to functions by using
  - a) Call by Value and
  - b) Call by referenceFields names are empno, name, and basic
18. Program to swap values of two variables by passing pointers.

19. Read two integers and determine bigger of the two with the help of function `big()` returning an integer pointer.
20. Create a sequential file and perform following operation:
  - a. Add records
  - b. Process & Display output.Field names are Roll, Name, M1, M2, and M3.
21. Program to draw basic shapes like lines, rectangles, ellipse, pie etc. using graphics.
22. Drawing concentric circles and rectangles with various colors.
23. Program to use various pens and brushes using graphics.
24. Creating a text file and copy contents to another file and print it.
25. Create a binary file, store n integers. Read the file and divide it into two different files containing odd and even values separately.
26. Create a menu driven program to add, modify, and delete a record an any entity.

### **Section B**

Minimum five study experiments based on unit IV through Unit VI

#### **REFERENCE BOOKS:**

1. The Art of programming through flowcharts & algorithm by Anil B. Chaudhari, Firewall Media, Laxmi Publication, New Publication.
2. Programming in C by E. Balagurusamy.
3. C Programming – Kernighan & Ritchie
4. C Programming – Dr. Vishal M. Lichade Wiley-Dreamtech Publication
5. Let us C – Y. Kanetkar.
6. C Programming – Holzner, PHI Publication.
7. Programming in C – Ravichandran.
8. Information Technology Concepts by Dr. Madhulika Jain, Shashank & Satish Jain, [BPB Publication, New Delhi.]
9. Fundamentals of Information Technology By Alexis And Mathews Leon [Leon Press, Chennai & Vikas Publishing House Pvt. Ltd, New Delhi]

#### **Web Resources:**

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. [www.tutorialspoint.com/cprogramming/](http://www.tutorialspoint.com/cprogramming/)
2. [www.programiz.com/c-programming](http://www.programiz.com/c-programming)
3. [www.w3schools.com](http://www.w3schools.com)
4. <https://ittutorials.net/>
5. [http://www.tutorialspoint.com/computer\\_fundamentals/](http://www.tutorialspoint.com/computer_fundamentals/)



## B.Sc. SEMESTER II

### OBJECT ORIENTED PROGRAMMING USING 'C++' and SYSTEM ANALYSIS AND DESIGN

#### Course Outline

#### Unit I

**Object Oriented Methodology:** Elements of Object Oriented programming, Objects, Classes, OOPs features.

Classes & Objects: Specifying a Class, Creating Objects, Accessing Class members, Defining member function, Outside Member Functions as inline, Accessing Member Functions within the class, Static data member, Access Specifiers: Private, Protected and Public Members.

#### Unit II

Constructors & Destructors: Introduction, Parameterized Constructors, Constructor Overloading, Constructors with Default Arguments, Copy Constructor, Destructor, Order of Construction and Destruction, Static data members with Constructor and Destructors.

Operator Overloading: Definition, Overloadable Operators, Unary Operator Overloading, Unary & Binary overloading, Rules for Operator Overloading.

**Dynamic Objects:** Pointers to Objects, Creating and Deleting Dynamic Objects: New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, this Pointer.

#### Unit III

**Inheritance:** Definition, Abstract classes, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Constructor and Destructor in Derived Classes.

Virtual Functions: Need for Virtual Functions, definition, Pure Virtual Functions, Abstract Classes, Rules for Virtual Function.

Exception Handling: Exception Handling Model, List of Exceptions, Handling Uncaught Exceptions, Fault Tolerant Design Techniques, Memory Allocation Failure Exception, Rules for Handling Exception Successfully.

#### **Unit IV**

Introduction: System, Subsystems, Components of Computerized Information System, Systems Analysts, SDLC, Prototyping.

Feasibility Study and Analysis: Identifying Problems, Organizing Feasibility Analysis: Economic, Financial, Organizational and Technological. Feasibility Decision, Choice of a solution.

**Data Collection:** Interviews, Brain Storming, Questionnaires, Document Search, Observation.

Structured tools and techniques of Data analysis: Structured English, Process Charts, SOP, Decision Tables and Decision Trees, Data Flow Diagram, Data Dictionary (Special emphasis on problem solving).

#### **Unit V**

System Design: Input design: Input Validation, Human factor Consideration, Messages, System Tolerance.

Output design: Categories of output, Design Principles, Control of Output. Forms: Principles of Form Design, Ways to ensure Quality Forms.

Codes: Types, Physical Representation of Codes, Principle of Code Design. Implementation: Training, Operational Training and Related Activities, Planning to Implement Change, Change Strategies.

#### **Unit VI**

**Testing:** Preparation for Testing, Test Execution: Levels of Testing, Component, Function, Subsystem, System, Test Evaluation, Acceptance.

Conversion: Cold Turkey, Parallel, Pilot, Modular and Sequential Methods. Conversion Period Length. System Evaluation Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Scheduling: Work Breakdown Structure, Activity Networks and CPM, Gantt Charts, PERT Charts, Project Monitoring and Control. Risk Management.

Software Configuration Management: Necessity, Configuring Management Activities Software Reliability and Quality Management: Software Reliability, Software Quality, ISO 9000. Software Maintenance: Characteristics of Software Maintenance, Maintenance Process Models, Estimation of Maintenance Cost. Software Reuse: What can be reused, Why no reuse so far, Basic Issues.

---

**B. Sc. Semester II**  
**COMPUTER SCIENCE PRACTICALS**

**Section A**

1. Demonstrate the usage of Constructor and Destructor. Define a class data with data member acct\_no, balance containing constructor data to initialize data member and a member function display () for output.
2. Program to demonstrate usage of a constructor and Destructor function. Declare a class with public data member count. The class containing one constructor and destructor to maintain updated information about active objects i.e. i) No of objects created. ii) No of objects Destroyed.
3. Program to accept the distance between city 1st & 2nd, city 2nd & 3rd. Calculate the distance between city 1st & 3rd. Define a class road with private data member km, m, d1, d2, d3 containing member function getdata () to accept values of d1, d2 and calculate for calculating distance.
4. Demonstrate the use of operators overloading (string manipulation: + for concatenation and relational operators for alphabetical comparison).
5. In a bank N depositor deposit the amount. Write a program to find total amount deposited in the bank. Declare a class deposit with private data member Rupee and Paisa containing member function getdata(), putdata(). Use array of objects, use operator '+' for overloading.
6. Declare class event and accept time of first event and second event and find the difference between 1st and 2nd event containing public member function getdata() and display with private data member hour, minute, second and total. Use operator '-' for overloading.
7. Program to demonstrate Single Inheritance. Declare a class B and derive publically class D from B.
  - a. The class B contains private data member a, public data member b with member function get\_ab(), get\_a(), show\_a().
  - b. The derived class D contains data member c with member function mul() and display().
8. Program to demonstrate Multiple Inheritances. Declare class M and N and derive publically class P from M and N.
  - a. Declare a class M with protected data member m and public member function get\_m().
  - b. Declare a class N with protected data member n containing member function get\_n().
  - c. Declare class P containing member function display ()).

9. Program to demonstrate Multilevel Inheritance. Declare a class student and derive publically a class test and derive publically class result from class test.
  - a. The class student contains protected data member roll\_number with public member functions get\_number() and put\_number().
  - b. The class test containing protected data member sub1, sub2 with public member function get\_marks() and put\_marks().
  - c. The class result contains data member total and public memberfunction display ().
10. Program to demonstrate Hierarchical Inheritance. Declare a class Side and derive publically class Square from base class Side and also derive publically class Cube from base class Side.
  - a. Class Side contains protected data member L with a member function set\_values().
  - b. Class Square contains member function sq().
  - c. Class Cube contains member function cub().
11. Program to demonstrate usage of normal virtual function and purevirtual function with abstract class.
12. Program to determine whether the input is +ve or –ve through exception.
13. Program to raise exception if an attempt is made to perform divide by zero.

## **Section B**

Minimum five study experiments based on unit IV through Unit VI

### **REFERENCE BOOKS:**

1. Mastering C++ by K R Venugopal Tata McGraw-Hill, New Delhi.
2. The C++ Programming Language –Bjarne Stroustrup
3. Programming with C++ - Ravichandran
4. Programming with C++ - Robert Lafore
5. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill
6. Information Systems Analysis, Design and Implementation By K. M. Hussain, Donna Hussain [Tata McGraw-Hill Publishing Company Ltd, New Delhi]
7. Fundamentals of Software Engineering by Rajib Mall [PHI Publication]

8. Workbook on Systems Analysis & Design by V. Garg [PHI Publication]
9. System Analysis and Design- Don Yeates, Shiebls, Helmy (M).
10. System Analysis & Design - Edward –TMH
11. System Analysis and Design – Satzinger, Robert Jackson and Stephen Burd, Thomson Learning

**Web Resources:**

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. [www.tutorialspoint.com/cplusplus](http://www.tutorialspoint.com/cplusplus)
2. [www.cplusplus.com/doc/tutorial](http://www.cplusplus.com/doc/tutorial)
3. [www.cprogramming.com/tutorial](http://www.cprogramming.com/tutorial)
4. [www.subjectcoach.com/tutorials](http://www.subjectcoach.com/tutorials)
5. [www.tutorialspoint.com/sdlc](http://www.tutorialspoint.com/sdlc)

**B. Sc. Semester III**  
**DATA STRUCTURES and OPERATING SYSTEMS**

**Course Outline**

**Unit I**

**Linked List:** Linked List, Representation of Single, Double, Header, Circular Single and Double Linked list, All possible operations on Single and Double Linked List using Dynamic representation, Polynomial Representation and its Manipulation.

**Unit II**

**Stacks:** Stacks terminology, Representation of Stacks in Memory, Operation on Stacks, Polish Notations, Translation of infix to postfix & prefix expression, Infix to Postfix Conversion, Evaluation of Postfix Expression, Recursion, Problems on Recursion, Quick Sort and Tower of Hanoi Problem.

**Queue:** Representation of Queues in Memory, Circular Queue. Dequeue and Priority Queue, Operations of above Structure using Array and Linked Representation.

**Unit III**

**Sorting and Searching:** Selection Sort, Insertion Sort, Merge Sort, Efficiency of Sorting Methods, Big-O Notations. Hash Tables, Hashing Technique, Collision Resolution Technique.

**Trees:** Basic Terminologies, Representation of Binary Trees in Memory, Traversing of Binary tree, Binary Search Tree, Operation on Binary Search Tree, Heap Tree, Operation on Heap Tree, Heap Sort Method.

**Graphs:** Basic Terminologies, Definition and Representation of Graphs in Memory: Linked List and Matrix Representation. Traversing graphs: BFS, DFS Method.

#### **Unit IV**

Structure of Operating System, Operating System functions, Characteristics of Modern OS. Process Management: Process states, Creation, Termination, Operations on Process, Concurrent processes, Process Threads, Multithreading, Micro Kernels.

**CPU Scheduling:** Schedulers, Scheduling Methodology, CPU Scheduling Algorithms: FCFS, SJF, RR, Priority Scheduling.

#### **Unit V**

Performance Comparison: Deterministic Modeling, Queuing analysis, Simulators. Deadlock and Starvation: Resource Allocation Graph, Conditions for Dead Lock, Dead Lock Prevention, Dead Lock Detection, Recovery from Deadlock.

Memory Management: Logical Vs. Physical Address Space, Swapping, Memory Management Requirement, Dynamic Loading and Dynamic Linking, Memory Allocation Method: Single Partition allocation, Multiple Partitions, Compaction, paging, segmentation, segmentation with paging, protection.

#### **Unit VI**

I/O Management: I/O hardware, I/O Buffering, Disk I/O, Raid, Disk Cache.

FileManagement: File Management system, File Accessing Methods, File Directories, File Allocation Methods, File Space Management, Disk Space Management, Record blocking.

Protection Mechanisms: Cryptography, Digital Signature, User Authentication.

**B. Sc. Semester III**  
**COMPUTER SCIENCE PRACTICALS**

**Section A**

1. Program to insert a node at the beginning, at the end and in the middle of the given linked list.
2. Program to delete a node at the beginning, at the end and in the middle of the given linked list.
3. Program to create a linked list of customer names and their telephone numbers. (Using Menu Driven and include features of adding a new Customer and deleting an existing Customer.)
4. Program to reverse a linked list.
5. Program to search a value in the given linked list.
6. Program to insert a node at the beginning, at the end or in the middle of a given doubly linked list.
7. Program to delete a node from the beginning, at the end or in the middle of a given doubly linked list.
8. Program to create, insert and delete a node in Circular linked list.
9. Program to push and pop an element into / from a stack implemented using linked list.
10. Program to push and pop an element into / from a stack implemented using Array.
11. Program to evaluate postfix expression.
12. Program to sort an array using quick sort.
13. Program to solve Towers of Hanoi problems using recursion.
14. Program to perform insertion and deletion operation in linear queues.
15. Program to perform insertion and deletion operation on circular queues.
16. Program to sort an array using: Insertion sort.
17. Program to sort an array using: Selection sort.
18. Program to insert an element in a binary search tree.
19. Program to traverse inorder of a binary tree.
20. Program to traverse preorder of a binary tree.
21. Program to traverse postorder of a binary tree.

**Section B**

Minimum five study experiments based on unit IV through Unit VI



## REFERENCE BOOKS:

1. Classical Data Structures: D. Samantha. PHI, New Delhi.
2. DATA STRUCTURE : LIPSCHUTZ SCHAUM OUTLINE SERIES
3. Data Structures Using C++ : Y. Kanetkar
4. Data Structures Using C++: Tanenbaum
5. Data Structures by Tremblay Sorenson
6. Operating Systems by P. Balakrishna Prasad [Scitech Publication]
7. Operating System Concepts:Silberschatz (Addision Education)
8. Operating Systems – H.M. Deitel–Addision Wesley.
9. Operating Systems– John J. Donovan.
- 10.Operating System : A.S.Godbole (TMH)
- 11.Modern Operating Systems : Tanenbaum (Pearson Education)
- 12.Operating Systems : Peterson

## Web Resources:

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. [www.tutorialspoint.com/data\\_structures\\_algorithms](http://www.tutorialspoint.com/data_structures_algorithms)
2. [www.onlinetutorialspoint.com/data-structures/what-is-data-structures](http://www.onlinetutorialspoint.com/data-structures/what-is-data-structures)
3. [www.topcoder.com/community/data-science/data-science-tutorials](http://www.topcoder.com/community/data-science/data-science-tutorials)
4. [www.tutorialspoint.com/operating\\_system](http://www.tutorialspoint.com/operating_system)
5. [www.researchgate.net/publication/3786159\\_Tutorial](http://www.researchgate.net/publication/3786159_Tutorial)

**B.Sc. SEMESTER IV**  
**JAVA PROGRAMMING and**  
**LINUX OPERATING SYSTEM**

**Course Outline**

**Unit I**

**Introduction to Java:** History of Java, features of Java, getting started with Java. **Java programs:** Introduction of Application & Applets. **Variables:** Variable naming, variable initialization, assign values, Rules of variables, Scope of variable. **Operators:** Arithmetic, Assignment, Unary, Comparison, Shift, Bit-wise, Logical, Conditional, new, Special, Relational. **Data types:** Integers, Char, String, Float etc., Typecasting. **Tokens:** Java tokens, Order of precedence of operators, **Streams:** Input and output. **Creating a class & subclass:** Declaring a class, Naming a class, Rules to assign Class & Subclass, Creating a new object, Class of an object. **Data members:** Declaring data member, Naming variables, using class members. **Methods:** Using data members, Invoke a method, passing arguments to a method, calling method. **Access Specifiers & Modifiers:** Public, Private, Protected, Static & Final. **Overloading:** Method overloading, Constructor overloading. **Java class library:** Different types of classes.

**Unit II**

**Decision making & loops:** If-then-else, Switch, ?: operator, While-loop, do-while loop, for loop. **Arrays:** Creating an array, one-dimensional array, two dimensional array. **String:** String array, string methods. **Inheritance:** Single & multiple inheritances. **Interfaces:** Defining interfaces, extending interfaces, implementing interfaces.

**Packages:** Java API packages, creating packages, accessing packages, adding a class to packages. **Import statement:** Introduction & implementation of import statement. **Applets:** Introduction to Applets & Application, how applets and application are different, creating an applet. Applets life cycle, designing a web page, creating an executable applet, running the applet, applet tags, passing a parameter to applet, HTML tag, Converting applet to application. **Threads:** Overview of threads, single & multiple threads, life cycle of threads, stopping & blocking threads, working with threads, priority to thread, synchronization. **Exceptions & Errors:** Introduction, types of error, exception, syntax of exception, handling techniques, exception for Debugging.

**Unit III**

**Event:** Event driven programming, handling an (AWT) events. **Graphics class:** Introduction, the Graphics classes, drawing & filling of lines, rectangle, circle & ellipse, arcs, polygons, text & fonts, creating a font class, font objects, text, coloring object. **Streams:** Introduction, Abstract stream classes, file input & output. **AWT Applications:** Creating a GUI using AWT toolkit, using component class, frames. **Components & Control:** Textfield, Textarea class, label, button, choice, list, checkbox, class, and combo. **Menus:** Creating a popup menus. **Image:** Type of image, Properties of an image, Displaying an image. **Layouts:** Using Window Listener interface, Different types of Layout, Layout manager, Flow manager, Grid manager. **Container:** Different types of container (Frame, Dialog, Panel).

**Unit IV**

Logging In and Logging Out, Anatomy of Linux OS, Directory Structure, /usr Directory, **File Types:** User data files, System data files, Executable files. Naming files and directories, Spawning Processes. **Shell:** Creating User Account, Shell Program, bash shell, Changing shell prompt. **Commands:** Basic Syntax for a command, Exploring the Home Directory, ls, mkdir, rmdir, stat, cat, rm, mv, cp. **Editor:** Vi editor. **Hooking up Hardware Devices:** Formatting a Floppy Disk, Gathering important system information.

**Unit V**

**Backing Up and restoring the File System:** Simple Backup, gzip, gunzip, tar. **Printing files:** Print Spool directory, Sending files to Printer. **Sharing Files with other Users:** Maintaining User Accounts, Changing Password, Creating Group Accounts, Granting Access to files, Changing File Ownership, Protecting Files, Making a File Read-Only. **Working with Processes:** Types of processes, ps Command, Creating process, killing process, free command and top utility.

**Unit VI**

**Managing Disk Space:** df, du commands, Creating Additional Free Disk Space, Locating Unused Files, Setting System Clock. **Communication Utilities:** who, who am i, finger, mesg, write, wall, talk, Creating a message of the day. X Window System, **Graphical User Interfaces:** KDE and GNOME Desktop Environment.

**B.Sc. Semester IV**  
**COMPUTER SCIENCE PRACTICALS**

**Section A**

1. A) Write a program that declares a class, object and also it accesses the data member of it's class.  
B) Write an applet that accepts a value from the user and display it.
2. A) Write a program that accept marks of 5 subject, calculate total, percentage and display the grade according to their percentage.  
B) Write a program that will print the multiplication table from 1 to 10.
3. A) Write a program to accept a set of values from the user into an array, display the values as well as their average.  
B) Accept string into a textfield, sort the characters in the string and display the sorted string into another textfield.
4. A) Write a program to demonstrate the overloading of constructor.  
B) Write an applet that accepts two numbers from user and display all the numbers between them.
5. A) Write a program to demonstrate the single inheritance.  
B) Write an applet to accept 10 numbers into an array, sort the array and display the sorted array. Accept the 10 numbers into the 10 different textfields.  
C) Write a program to create a multiple selection list and also display the list of items selected by the user.
6. A) Write an applet to demonstrate the user menu bar.  
B) Write a sample program that will convert the applet to application.  
C) Write a program to demonstrate the interfaces.
7. A) Write a program for exception handling that accept two numbers as textfields, the values are added and their sum is displayed. The code traps the error when user could enter text instead of numbers.  
B) Write a program that would accept input for the user and store it in a file called Test.java.
8. A) Write a program to implement Graphics class draw (line, rectangle, fill rectangle, circle, oval).

- B) Write an applet that display a choice of menu of three buttons (Add, Modify, Delete) selecting a choice from the menu should display the appropriate button. Use the show ( ) method of the Layout Manager.
9. A) Write a Java program to read and display the information from the file ABC.dat.
- B) Write a program to implement the concept of loading and displaying of images.
- C) Write a program to demonstrate the Animation in Java.

### **Section B**

Minimum five study experiments based on unit IV through Unit VI

### **REFERENCE BOOKS:**

1. The Art of Programming through Flowcharts & Algorithms by Anil B.Chaudhari, Firewall Media, Laxmi Publication, New Publication.
2. Programming with Java a Primer II edition: E Balagurusamy (TMH).
3. Java Programming (For Absolute Beginners) - Russell, PHI
4. Black Book on Java
5. Java-Complete References
6. Programming in Java I<sup>st</sup> Edition – Dr. S.B.Kishor, Rajani Singh, Das Ganu Prakashan.
7. SAMS Teach Yourself Linux-Craig and Coletta Witherspoon [Techmedia]
8. LINUX Complete Reference by Richard Peterson
9. Linux Operating System I<sup>st</sup> Edition - Dr. S.B.Kishor, Suhashini Chaurasia, Das Ganu Prakashan.

### **Web Resources:**

Students are advised to make use of the resources available on the Internet. Some useful links related to computer science are given below.

1. [www.tutorialspoint.com/cprogramming/](http://www.tutorialspoint.com/cprogramming/)
2. [www.tutorialspoint.com](http://www.tutorialspoint.com)
3. [www.javatpoint.com/java-tutorial](http://www.javatpoint.com/java-tutorial)
4. [www.tutorialspointexamples.com](http://www.tutorialspointexamples.com)
5. <http://www.javatpoint.com/linux-tutorial>
6. <http://www.guru99.com/unix-linux-tutorial.html>
7. [www.w3schools.com](http://www.w3schools.com)