

**Shiksha Mandal's**  
**Bajaj College of Science, Wardha**  
(Formerly known as 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.

**SYLLABUS FOR B.Sc. COMPUTER SCIENCE**  
**SEMESTER I and II**

**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.

**Syllabus for B.Sc. SEMESTER I**  
(To be implemented from Academic Session 2021-22)

<b>Course Objectives</b>	<b>Course Outcomes</b>
Students will try to learn:	After successful completion of the course student will be able to
1. To understand the concepts of fundamentals of a computer system, number systems, generations of computer languages, and other system tools like translator programs.	1. Understand various number systems, generations of computer languages and translators.
2. To understand various types of memories, storage devices, input and output devices.	2. Know various types of memories, storage devices, input and output devices.
3. To get acquainted with concepts of networks and its associated terminologies, topologies, architecture, communication devices and recent technologies like Bluetooth and Wi-Fi.	3. Learn the concepts of networks, topologies, architecture, communication devices and recent technologies.
4. To understand the basics of 'C' programming. Also get knowledge of algorithm and flowcharts.	4. Learn to make basics programs and also able to draw algorithm and flowcharts of programs.
5. To know the concepts of control and looping structure. Also student get the idea about concepts of arrays and functions.	5. Perform programs using control and looping structure also understand the basic programs on arrays and functions.
6. To understand the basics of structures, pointers and file handling.	6. Implementation of structures, pointers and file handling concepts in programming.

**FUNDAMENTALS OF INFORMATION TECHNOLOGY and  
PROGRAMMING IN 'C'**

**Course Code: UG-CS (05)-S1-T**

**Course Outline**

**Unit I**

Evolution of Computers, Generations of Computer, Characteristics of a Computer, Computer Classification.

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 II**

Memory: Memory hierarchy, RAM, Static & dynamic RAM, Types of RAM (SDRAM, RDRAM, DDR), ROM, PROM, EPROM, EEPROM, Flash and Cache. Storage Devices: Hard Disk, Zip Disk and Optical Disk. Pen Drive, Blu Ray and SSD.

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 III**

Computer Network: Goals and applications - Business Application, Home Application, 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. TCP/IP Reference Model with functionality of each layer.

Internet services (Introduction only): WWW – Web browser, URL, Internet search engines, WWW development languages, Electronic mail – E-mail address, e-mail message format, e-mail services (application based e-mail,

webmail), how email works (client-server model), File Transfer Protocol – How FTP works (client-server model), Terminal network, Uses of Internet

#### **Unit IV**

Programming Languages and Tools, Machine language, Assembly language, High level languages. 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, character and string handling Functions.

#### **Unit V**

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. Macro: single-line and multi-line. Storage Classes. (Special emphasis on problem solving)

#### **Unit VI**

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.

**B.Sc. Semester I**  
**COMPUTER SCIENCE PRACTICALS**  
**Subject Code: UG-CS(05)-S1-P**

**Section A**

Minimum five study experiments based on unit I through unit III

**Section B**

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.

### 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/)



**Syllabus for B.Sc. SEMESTER II**  
(To be implemented from Academic Session 2021-22)

<b>Course Objectives</b>	<b>Course Outcomes</b>
<b>Students will try to learn:</b>	<b>After successful completion of the course student will be able to</b>
1. To introduce the basics of Object Oriented Programming like classes, objects, abstraction, encapsulation and polymorphism, etc.	1. Learn to write programs using Object Oriented Programming concepts like class and object.
2. To understand the concepts of constructor and destructor; also familiar with operator overloading and dynamic objects.	2. Implementation of creation, deletion of objects and to make dynamic objects.
3. To get an idea of inheritance, virtual functions and exception handling.	3. Understand the implementation of inheritance, virtual functions and exception handling using programming.
4. To know about components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques used for data collection and data analysis.	4. Understand components of Computerized Information Systems, role of Systems Administrator, various software development models; structured tools and techniques.
5. To understand concepts associated with input and output design, form design, activities related to implementation and change strategies related to a computer system.	5. Know concepts associated with input and output design, form design, implementation activities and change strategies.
6. To understand the principles of testing strategies, conversion methods, planning and scheduling techniques, SCM, ISO, software re-engineering and software maintenance costs.	6. Get acquainted with the principles of testing strategies, conversion methods, planning and scheduling techniques, and advanced concepts.

**OBJECT ORIENTED PROGRAMMING USING ‘C++’ and  
SYSTEM ANALYSIS AND DESIGN**

**Course Code: UG-CS (05)-S2-T**

**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, Dynamic Constructor, Destructor, Order of Construction and Destruction, Static data members with Constructor and Destructors, Virtual Destructor.

Operator Overloading: Definition, Overloadable Operators, Unary Operator Overloading, 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. Runtime polymorphism.

Exception Handling: Exception Handling Model, List of Exceptions, catch with ellipses catch( ) 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 model and Spiral model.

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.

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**

**Subject Code: UG-CS(05)-S2-P**

**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 member function 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 pure virtual 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.
14. Program to accept customer account information like Customer Name, Account Number, Account Type. Implement the following operations using inheritance:
  - a. Deposit
  - b. Withdraw
  - c. Display Balance
  - d. Display Account details.

### 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**  
**Subject Code: UG-CS(05)-S3-T**  
**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.

File Management: 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**  
**Subject Code: UG-CS(05)-S3-P**

**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. Donoven.
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**  
**Subject Code: UG-CS(05)-S4-T**  
**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**  
**Subject Code: UG-CS(05)-S4-P**  
**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)

**B.Sc. SEMESTER V**  
**VISUAL BASIC PROGRAMMING and**  
**DATABASE MANAGEMENT SYSTEM**  
**Subject Code: UG-CS(05)-S5-T**  
**Course Outline**

**Unit I**

**Working with Visual Basic Window Components:** Menu Bar, Tool Bar, Project Explorer Window, Form Layout Window, Properties Window, Toolbox, Code Editor Window **Working with Forms:** Properties, Events, Methods **Working with Basic Controls:** Label, Command Button, Text Box, Option Button, Frame, CheckBox, ListBox, ComboBox, Image, Scroll, Picture, Timer, DriveListBox, DirListBox, FileListBox and Shape Controls. **Basic Programming Fundamentals:** Variables, Data types, Constant, Conversion Function. **Scope of Variable:** Public, Private, Static. **Operators:** Logical, Arithmetic, Concatenation, Comparison. **Decision Structure:** If.. Then, If..Then..Else, Select Case.. End Case. **Loop Structure:** Do..While, While.. Wend, For.. Next, With.. EndWith. DoEvents ().

**Unit II**

**Arrays:** Dynamic Array, Preserve and Control arrays. **Procedure:** General procedure, General Methods for Passing Arguments to a Procedure, **Functions:** User-Interaction, String, Math, Date, Conversion Functions. **Modules:** Form, Standard.

**Menus:** Creating, Adding Menu Items, Creating Shortcut, Adding Separators Bars, Submenus, Code for Menus. **Creating Popup Menu:** System, Custom.

**Unit III**

**Database Handling:** Database Concepts, Creating and Accessing Database, Using Data Control. **Using DAO:** Creating Search Programs, Numeric Search and Complex Search Programs.

**Using ADO Data Control:** Data Link, ODBC Data Source name, Using Connection String, Creating Navigating buttons. **Working with Advanced Data Controls:** DataList Control, DataCombo Control, DataGrid Control and Msflexgrid Control. **Handling Errors:** Run Time, Trapping and Handling Error, ERR Object. Data Environment and Data Reports.

**Unit IV**

**DBMS:** Definition: Databases, DBMS, Problems with traditional file processing system, Objectives of the database systems, Three level architectures of DBMS, Components of DBMS, Database Administrator, Database Users, Data model, Different types of data models, Concepts of Hierarchical, Network Models.

**E-R Models:** Basic Concepts, Entity, Attributes, Relationship, Mapping, Keys, Weak and Strong Entity Set, Problems on E-R Diagrams.

## Unit V

**Extended E-R Features:** Specialization, Generalization, Aggregation, Problems on Reduction of an E-R Schema to Tables, Tabular representation of Strong, Weak entity Sets and Relationship Sets.

**Relational Model:** Structure, Relational Algebra, Fundamental Operations, Set – Intersection, Natural Join, Division and Assignment Operation. Extended Relational Algebra Operations, Aggregate Functions.

## Unit VI

**Functional Dependency:** Functional Dependency, Fully Functional Dependency, Partial Dependency, Transitive Dependency, Multi Valued Dependency. **Normalization:** Normal Forms (1NF, 2NF, 3NF, BCNF, 4NF, 5NF). Problems on Normal forms.



**B. Sc. Semester V**  
**COMPUTER SCIENCE PRACTICALS**  
**Subject Code: UG-CS(05)-S5-P**

**Section A**

1. A) Make an application in Visual Basic, which display a picture when
  - i. Command button with caption “click here” is clicked.
  - ii. After delay time 100ms (by using Timer Control).
 B) Design an application to find
  - i.  $\sin x = x - x^3/3! + x^5/5! - x^7/7! + \dots$
  - ii.  $\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$
  - iii. Find an even /odd number.
  - iv. Find Factorial of number.
  - v. Find if number is a Prime or Not Prime.
  
2. A) A book stall gives discount on the books as per following conditions,

No. of books purchased	Discount
$\leq 5$	Nil
$> 5$ and $\leq 10$	10%
$> 10$ and $\leq 15$	12%
$> 15$	20%

Create a form as follows to calculate the discount.

Unit Price	<input style="width: 60%;" type="text"/>
No. of Books	<input style="width: 60%;" type="text"/>
Total Price	<input style="width: 60%;" type="text"/>
Discount	<input style="width: 60%;" type="text"/>
Total Amount	<input style="width: 60%;" type="text"/>

- B) Design a form for speed control program using scrollbars.
  
3. A) Design an application, which display a text and given choice to change text to bold, italic, underline (any one or in combination) using check box controls.
- B) Write a VB code to give five choices of colors. Design an application to choose any one color using option button and change the forecolor of TextBox Caption.

4. A) Write a program in VB to build application to display sorted list of group A & group B students in two list boxes. Make provision to shift name (s) on either side using List Box control.  
 B) Write a program in VB to build application which display list of states. Make provision to display capital of the state and cities when a particular state is selected using combo box.
5. A) Write a VB Program to create an array of N elements. Find the frequency of elements and display the positions in an array.  
 B) Create an application, which counts no. of characters and no. of words entered in TextBox.
6. Create Menus using Menu Editor as follows.

File
New
Open
Save
Exit

Edit
Cut
Copy
Paste

And write appropriate code on click event of New and Exit menu options.

7. Create a database pay.mdb using Visual Database Manager with fields Code Number(n), Name of Employee(T), Designation(T), Address(T), Joining Date(d), Basic Pay(n), Status(T) [Permanent / Temporary], Department(T) [Purchase, Sales, Account]. Achieve connectivity using Data Control. Develop an application in VB to display data of employees using MSFlexgrid control as shown below with company's logo.

EMPLOYEE INFORMATION							<logo>
ID	NAME	ADDRESS	DESIGNATION	DEPT.	Date of Joining	Status	BASIC

8. Design a menu as shown below.

DATABASE	Record
Open	Append
Exit	Save

Create table pay.mdb using Visual Data Manager and connect it to VB

application using Data Control. Display/view records using DbGridControl. Create an application in VB to open database of employee pay.mdb by using menu option "OPEN". Use APPEND and SAVE options to add and save records in DATABASE.

9. Open database fees.mdb with fields:-

Name, Class, Fees using Visual Data Manager. Use Data control for connectivity. Develop a menu driven application in VB to:-

- I) Compute total fees class wise
- II) Search a record of a particular student belonging to a particular class.

DATABASE	Record
Open	Compute total fees
Exit	Append
	Save

10. Create a database College.mdb containing fields:-

Name, Class, Marks in Phy, Comp. Sc., Maths and Percentage using Visual Data Manager. Connect database to VB application using Data Control. Develop a Menu driven application in VB to

- I) Edit a record class wise.
- II) Calculate class wise passing percentage where a student is declared "pass", if he/she gets percentage  $\geq 45$ .

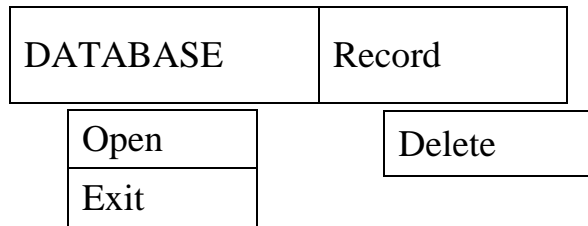
Display class wise list of students who are placed in I<sup>st</sup> Division. (Percentage  $\geq 60$ )

DATABASE	Edit
Create	B.Sc. - I
Display	B.Sc. - II
Exit	B.Sc. - III

11. Create a table book.mdb using VDM containing fields:-

Name of Book, Subject, Price, Date of Purchase, Name of Publication. Develop a menu driven application in VB to:-

- I) Open Database.
- II) Delete a record of book.
- III) Display subject wise list of book in proper format.



Use Data Control for connectivity.

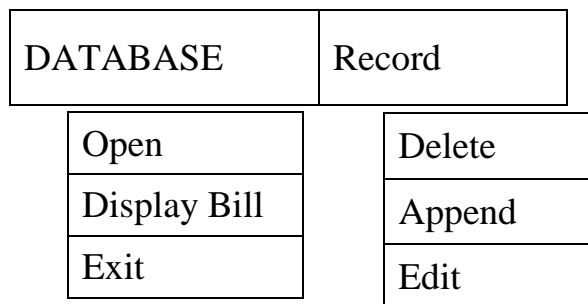
12. Create a table using VDM to store required information for computing Electricity Bill.

Charges are as follows:

- 1) 1.25 / units for 1<sup>st</sup> 30 units.
- 2) 2.90 / units from 31 to 300 units.
- 3) 4.0 / units from 301 to onwards.

Develop a menu driven application to:-

- I) Open Database
- II) Display Electricity Bill in proper format.
- III) Delete a record of book
- IV) Append a record.
- V) Edit record

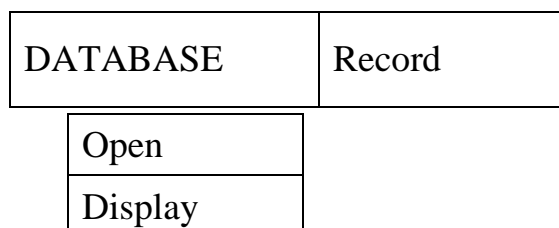


Use ADODC to achieve connectivity.

13. Create the Database emp.mdb having table emp\_biodata directly through MS-Access and display record using DataGrid Control.

Develop a menu driven application to:-

- I) Open Database
- II) Display records using DataGrid Control.
- III) Search a record.
- IV) Append a record.



Exit	Append
	Search

Use ADODC to achieve connectivity.

14. Create an application for employee salary sheet.

- If BASIC > 5000
  - HRA= 40% of BASIC
  - DA= 50% of BASIC
- If BASIC >= 2500
  - HRA= 35% of BASIC
  - DA= 45% of BASIC
- If BASIC >= 1500
  - HRA= 30% of BASIC
  - DA= 40% of BASIC
- Otherwise
  - HRA= 20% of BASIC
  - DA= 30% of BASIC
- TA=150
- PF= 15% of BASIC
- PT= 18% of BASIC

Develop a menu driven application in VB to:-

- I) Compute Total Salary.
- II) Search a record of a particular employee belonging to a particular department.

DATABASE	Record
Open	Compute Total Salary
Display	Search
Exit	

Use ADODC to achieve connectivity.

Note: -While designing menu, MDI forms are expected whenever required.

## Section B

Minimum five study experiments based on unit IV through Unit VI.

### REFERENCE BOOKS:

1. VISUAL BASIC – to Advance by Soma Dasgupta [BPB Publication]
2. Evangelos Petroustos, Mastering Visual Basic 6.0 BPB Publication.
3. VISUAL BASIC 6 COMPLETE REFERENCE (TMH PUB)
4. Visual Basic 6 Deitel & Deitel (Pearson Education)
5. Mastering VB 6.0 Black Book -Peter - Norton-Techmedia
6. Front End Development Using Visual Basic 4<sup>th</sup> Edition –Dr. S.B.Kishor, Das Ganu Prakashan.
7. Data Base System Concepts By A SilbersChatz By Henry Korth And S.Sudarshan
8. Introduction to Data Base Management by NAVEEN PRAKASH [Tata McGrawHill Ltd.]
9. Bipin C. Desai, An Introduction to Database Systems, Galgotia Publications.
10. Raghuram Ramakrishnan & Johannes Gerhrke, “Data Base Management Systems”.
11. Muzumdar, Introduction to Database Management Systems, TMH.
12. Database Management System – Dr. S.B.Kishor, 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/listtutorials/visual-basic/1](http://www.tutorialspoint.com/listtutorials/visual-basic/1)
2. [www.vbtutor.net/vbtutor.html](http://www.vbtutor.net/vbtutor.html)
3. [www.visual-basic-tutorials.com](http://www.visual-basic-tutorials.com)
4. [www.freetutes.com/learn-vb6](http://www.freetutes.com/learn-vb6)
5. [www.beginnersbook.com](http://www.beginnersbook.com)
6. [www.w3schools.in/dbms](http://www.w3schools.in/dbms)
7. [www.tutorialcup.com](http://www.tutorialcup.com)

**B. Sc. Semester VI**  
**COMPILER CONSTRUCTION, SQL and PL / SQL**  
**Subject Code: UG-CS(05)-S6-T**  
**Course Outline**

**Unit I**

**Compilers and Translators:** Need, The Structure of a Compiler, Lexical Analysis, Syntax Analysis, Intermediate Code Generation, Optimization, Code Generation, Book Keeping, Error Handling.

**Unit II**

**High Level Programming Languages:** Definitions of Programming Languages, The Lexical and Syntactic Structure of a Language, Data Elements, Structures, Operators, Assignment Statements, Data Environments, Parameter Transmission, Storage management.

**The Role of the Lexical Analyser:** Approach to the Design of Lexical Analyzer, Implementation of Lexical Analyzer, Context Free Grammars, Derivations and Parse Trees, Ambiguous Grammar.

**Unit III**

**Parsers:** Shift-Reduce Parsing, Operator Precedence Parsing, Top-Down Parsing, Predictive Parsers, Symbol Table,

**Code Optimization:** The Principal Source of Optimization, Loop Optimization, DAG Representation of Basic Blocks.

**Code Generation:** A Machine Model, A Simple Code Generator, Register Allocation and Assignment.

**Unit IV**

**CODD'S Rules,** Oracle Database Objects, Sub Languages of SQL, Data Types, Operators. **DDL Statement:** Creating Tables, Deriving Table from Existing Table, Altering, Dropping Tables. Integrity Constraints, Specifying Names for the Constraints, Viewing Integrity Constraints, Adding and Dropping Constraints. **DML Statements:** SELECT statement, Insert, Update, Delete, Working with Sequences and Synonyms. **Built-in functions:** Arithmetic, Date, Character, Conversion, Single row, Aggregate, Decode. Joins, Set Operators and Sub queries. **DCL and TCL Statements:** Grant, Revoke, Commit, Rollback and Savepoints.

**Unit V**

**VIEWS:** Creating Views, Dropping Views, Inserting, Updating and Deleting Data using Views, Types of Views. **PL/SQL Programming:** PL/SQL Data Types, Identifiers, Operators and Expressions, Iterative Statements, Conditional Statements, Emphasis on Problems.

**Exception Handling:** Predefined Exceptions, User Defined Exceptions. **Cursors:** Declaring Cursors, Opening and Retrieving Records, Closing Cursors. Attributes of Explicit and Implicit Cursors, Parameter Passing in Cursors, Problems on Exception Handling, Cursors.

**Unit VI**

**Procedures:** Create and Drop Procedure, Creating Procedures with Parameters, Calling Procedures, Granting the EXECUTE Permission, Problems on Procedures.

**Function:** Creating and Dropping Function, Purity Levels in Functions, Executing Functions. **Triggers:** Create Triggers, Type of Triggers, Creating BEFORE and AFTER Triggers, INSTEAD-OF Triggers, Trigger Predicates, Inserting, Updating and Deleting Triggers, Enabling, Disabling and Dropping Triggers. Problems on Functions and Triggers.

**B. Sc. Semester VI**  
**COMPUTER SCIENCE PRACTICALS**  
**Subject Code: UG-CS(05)-S6-P**

**Section A**

## 1. DDL Statements :

- i) **Create the table Invoice with fields** inv\_no(char, Primary Key), mv\_no(number), cust\_id(char), issue\_date(date), return\_date(date).

Insert data in the table Invoice as follows:

inv_no	mv_no	cust_id	issue_date	return_date
101	1	a01	21-Dec-06	25-Dec-06
102	3	a02	19-Oct-06	30-Oct-06
103	5	a03	14-Jul-06	1-Aug-06
104	5	a04	15-Sep-06	30-Sep-06
105	6	a05	19-Oct-06	21-Oct-06
106	7	a06	1-Jan-06	15-Jan-06
107	8	a07	15-Aug-06	30-Aug-06
108	2	a05	1-Oct-06	15-Oct-06
109	4	a01	12-Sep-06	19-Sep-06
110	9	a02	1-Dec-06	15-Dec-06
111	6	a05	2-Feb-06	14-Feb-06
112	3	a06	1-Jan-06	10-Jan-06



- ii) Modify Table Structure by adding new field price(number) and increase the size of inv\_no.
- iii) Add and drop constraints
  - i. Add Primary Key constraint.
  - ii. Drop Primary Key constraint.
  - iii. Add Unique Key constraint.
  - iv. Drop Unique Key constraint.
  - v. Add Check constraint.

2. DML Statements:

- i) Add a record in the above table.
- ii) i) Change inv\_no of I12 with 'I111'.  
ii) Change issue\_date of customer of cust\_id 'a01' to '22-Dec-06'.  
iii) Change movie no of cust\_id 'a05' to 11.
  
- iv) Display all movies with issue\_date between 01-Jan-06 to 01-Sep-06.
- v) Display information for cust\_id 'a01' & 'a02'.
- iii) i) Delete records with inv\_no 'I11' from invoice table.  
ii) Delete records having return date <='20-Jan-06'.

3. DCL and TCL Statements

- i) Grant insertion and deletion privileges on Invoice Table to another user XYZ.
- ii) Revoke the deletion privileges from user XYZ.
- iii) After insertion of 2 records in invoice table create Savepoint A.
- iv) Delete a record and create Savepoint B.
- v) Rollback to A.

4. Create the table Movie with fields mv\_no(number, Primary Key), title(char), type(char), star(char), price(number), releasedate(date)

- i) Add few records in above table.
- ii) Count total number of customer.
- iii) Calculate total price of all the movies.
- iv) Calculate average price of all the movies.
- v) Determine the maximum & minimum movies prices and rename the title as MAX-PRICE and MIN-PRICE.
- vi) Count the number of movies having price greater than or equal to 150.

- vii) Display total no. of movies released in the current year.
- viii) Display all the movie which start with letter 'K'.

5. i) **Create a View** of table Invoice with following columns inv\_no(char, Primary key), mv\_no(number), cust\_id(char).

- ii) Insert and delete records.
- iii) Drop view

6. **PL/SQL Programming:**

Write a PL/SQL block:

- i) To find largest of 2 numbers.
- ii) To reverse the number.

7. **Cursor:**

- i) Write a cursor to fetch all the rows of table Invoice.
- ii) Write a cursor to display all the movies with issue\_date between 01-Jan-06 to 01-Sep-06. If no record found then use Exception to display proper message.
- iii) Write a PL/SQL to accept the name and age of student and insert into table STUD after validating age not less than 18 otherwise raise exception.

8. **Create Procedure:**

- i) To swap two values.
- ii) To display the cust\_id, mv\_no and call the procedure through main program.

9. **Create Function:**

- i) To find gcd of two numbers.
- ii) To find the total price of all the movie starting with letter 'K' and call the function through main program.

10. **Create Trigger:**

- i) That restricts the user from performing a DML on movie table on 'Monday'.
- ii) Covert all the movie name into the upper case on insertion of each row.

### Section B

Minimum five study experiments based on unit I through Unit III.

#### REFERENCE BOOKS:

1. Principles of Compiler Design - A.V. Aho, J. D.Ullman: Pearson Education.
2. Modern Compiler Design- Dick Grune, Henry E. Bal, Cariel T. H. Jacobs, Wiley Dreamtech.
3. Engineering a Compiler-Cooper & Linda, Elsevier.
4. Compiler Construction, Louden, Thomson.
5. Compiler Construction - Dr. S.B.Kishor, Nishant Shastrakar, Das Ganu Prakashan.
6. Understanding ORACLE by Ivan Bayross [BPB Publication].
7. Database System Using Oracle: A Simplified Guide to SQL & PL-SQL: Nilesh Shah, PHI Publication.
8. Database Management Systems (Complete Practical Approach) by Sharad Maheshwari & Ruchin Jain, Firewall media.
9. Dr. P.S.Deshpande SQL & PL/SQL for Oracle 10g Black Book.
- 10.Scott Urman Programming PL/SQL TMH.
- 11.Oracle (SQL and PL/SQL Programming) 2<sup>nd</sup> Edition - Dr. S.B.Kishor, Das Ganu Prakashan.
- 12.Database Programming with Oracle - Dr. S.B.Kishor, 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](http://www.tutorialspoint.com)
2. [www.symbolaris.com](http://www.symbolaris.com)
3. [www.w3schools.com](http://www.w3schools.com)
4. [www.beginner-sql-tutorial.com/sql.htm](http://www.beginner-sql-tutorial.com/sql.htm)
5. [www.guru99.com/sql.html](http://www.guru99.com/sql.html)
6. [www.sqlcourse.com](http://www.sqlcourse.com)
7. [www.tutorialspoint.com/compiler\\_design/compiler\\_design\\_syntax\\_analysis.htm](http://www.tutorialspoint.com/compiler_design/compiler_design_syntax_analysis.htm)