AC. 10/02/2012
Item No. 4.42

UNIVERSITY OF MUMBAI

Syllabus for S.Y.B.Sc.
Program: S.Y.B.Sc.
Course: Computer Science

(Credit Based Semester and Grading System with effect from the academic year 2012–2013)

S.Y.B.Sc. Computer Science Syllabus

Restructured for Credit Based Semester and Grading System

To be implemented from the Academic year 2012-2013

Semester I

COURSE CODE	COURSE/PAPER	LECTURES /WEEK	CREDITS
USCS101	**Discrete Mathematics**	**3**	**2**
USCS102	**C++ Programming**	**3**	**2**
USCS103	**Data Base Management Systems -I**	**3**	**2**
USCSPG1	**Practical - I**	**9**	**3**

Practical- I: 3 lecture periods per course per week per batch. All three lecture periods of the practical shall be conducted in succession together on a single day.

SEMESTER I		
Course Code	**Title**	**Credits**
	THEORY	
USCS 101	**DISCRETE MATHEMATICS**	**2 Credits (45 lectures)**
Unit I	**UNIT I: Relations:** **Relations**: Definition and examples. Properties of relations, Partial Ordering sets, Linear ordering Hasse Daigrams , Maximum and Minimum elements, Lattices **Recurrence Relation**: Definition of recurrence relations, Formulating recurrence relations, Solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Particular solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generation functions, Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms.	15 lectures

Unit II	**Graphs and Trees:** **(a) Graphs :** Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs[R6.3] Warshall's algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms - searching in a graph; Insertion in a graph, Deleting from a graph, Traversing a graph- Breadth-First search and Depth-First search. **(b) Trees:** Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, representing binary trees in memory, traversing binary trees, binary search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree	15 lectures
Unit III	**Counting Principles, Languages and Finite State Machine:** **(a) Permutations and Combinations:** Partition and Distribution of objects, Permutation with distinct and indistinct objects, Binomial numbers, Combination with identities: Pascal Identity, Vandermonde's Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects. **(b) Counting Principles:** Sum and Product Rules, Tree diagram for solving counting problems, Pigeonhole Principle(without proof); Simple examples, Inclusion Exclusion Principle (Sieve formula) (Without proof). **(c) Languages, Grammars and Machines:** Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines.	15 lectures

References:

1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
2. Discrete Mathematics and its applications: Kenneth H. Rosen, Third Edition, McGraw- Hill Inc.
3. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
4. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.
5. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.

Additional References:

1. Norman L. Biggs: Discrete Mathematics, Revised Edition, Clarendon Press Oxford 1989.
2. K.D. Joshi: Foundations in Discrete Mathematics, New Age Publication , New Delhi.
3. Edward R. Scheinerman, Mathematics, Thompson learning

USCS 102	C++ Programming	2 Credits (45 lectures)
Unit I	**Concept of OOP and introduction to C++** **(a) Concept of OOP**: Procedure Oriented Programming, Object Oriented Programming (OOP), Basic Concept of OOP, Benefits of OOP. **(b) Data Types** : Basic data types, User defined data types, Derived data types. **(c) Variables**: Declaration of variables, Dynamic Initialization of variables, Reference variables. Comments. **(d) Introduction to C++** : Parts of C++ program, cout and cin objects, #include Directive, Variables and Constants, Comments. **(e) Operators**: Arithmetic, Relational & Logical Operators, Type cast Operator, Scope resolution operator, Memory Management Operators. **(f) Control Structures**: Loops: for, do…while, while. Decision: if…else, switch case. **(g) Functions**: main function, Function Prototyping, Call by reference, Return by reference, Inline functions, Default arguments, const arguments, Function overloading. **(h) Classes & Objects** : Defining a class, Defining member Functions, making an Outside function Inline, Nesting of Member Functions, Access Specifiers, Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays of Objects, Objects as Function Arguments, Friend functions. **(i) Constructors and Destructors** : Constructors, Parameterized constructors, Multiple constructors in a class, Constructors with default arguments, Dynamic initialization of Objects, Copy constructor, Dynamic Constructors, Constructing Two-dimensional Arrays, const Objects, Destructors.	15 lectures
Unit II	**Manipulators, Operator Overloading, Inheritance, Polymorphism:** **(a) Manipulators** : C++ Streams, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing O/P with Manipulators. **(b) Operator Overloading :** Introduction, Defining Operator Overloading, Overloading Unary and Binary Operators, Overloading Unary and Binary Operators using Friend functions, Rules for Overloading Operators. **(c) Inheritance** : Introduction, Defining Derived classes, Single, Multilevel, Multiple, Hybrid Inheritance, Virtual Base classes, Abstract class, Constructors in derived classes. **(d) Polymorphism**: Introduction, Pointer to Objects, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions.	15 lectures

Unit III	**File handling, Templates, Exceptions, Standard Template Library** **(a) File handling** : Introduction, Classes for File Stream Operations, Opening and Closing a File, Detecting End of File, More about Open(), File pointers and their Manipulations, Updating a file, Error Handling during File Operations, Command line arguments. **(b) Templates** : Introduction, Class Templates, Class Templates with multiple parameters**,** Function Templates, Function Templates with multiple parameters, Overloading of Template Functions, Member Function Templates, Non-Type Template Arguments. **(c) Exceptions**: Introduction, Basics of Exception Handling, Exception Handling Mechanism, Throwing Mechanism, Catching Mechanism, Re-throwing an Exception, Specifying Exceptions. **(d) Standard Template Library**: Introduction, Components of STL, Containers, Algorithms, Iterators, Applications of Container classes.	15 lectures

Main Reference:

1. Object Oriented Programming with C++, 2e, E. Balguruswamy, TMH.
2. Starting out with C++, 3e, T. Gaddis, Dreamtech.

Additional References:

1. Complete Reference C++, H. Schildt, TMH.
2. Object Oriented Programming with C++, Y. Kanitkar, BPB.

USCS 103	**Database Management Systems -I**	**2 Credits (45 lectures)**
Unit I	**Relational Model** **(a) Overview:** Overview of database management system, limitations of data processing environment, database approach, data independence, three level of abstraction, DBMS structure. **(b) Entity Relation Model:** Entity, attributes, keys, relations, cardinality, participation, weak entities, ER diagram, Generalization, Specialization and aggregation, conceptual design with ER model, entity versus attribute, entity versus relationship, binary versus ternary relationship, aggregate versus ternary relationship. **(c) Relational Structure:** Introduction to relational model, integrity constraints over relations. **(d) Schema refinement and Normal forms:** Functional dependencies, first, second, third, and BCNF normal forms	**15 Lectures**

	based on primary keys, lossless join decomposition.	
Unit II	**Query Languages** **(a) Relational Algebra :** select and projection, Set operations like union, intersection, difference, cross product, Joins – conditional, equi join and natural joins, division, examples. Overview of relational Calculus. **(b) Creating and altering tables:** Conversion of ER to relations with and without constraints; CREATE statement with constraints like KEY, CHECK, DEFAULT, ALTER and DROP statement. **(c) Handling data using SQL:** selecting data using SELECT statement, FROM clause, WHERE clause, HAVING clause, ORDER BY, GROUP BY,DISTINCT and ALL predicates, Adding data with INSERT statement, changing data with UPDATE statement, removing data with DELETE statement **(d) Functions:** Aggregate functions-AVG, SUM, MIN, MAX and COUNT, Date functions: DATEADD(), DATEDIFF(), GETDATE(), DATENAME()YEAR, MONTH, WEEK, DAY, String functions- LOWER(), UPPER(), TRIM(), RTRIM(), PATINDEX(), REPLICATE(), REVERSE(),RIGHT(), LEFT() **(e) Joining tables:** Inner, outer and cross joins, union. **(f) Sub queries:** sub queries with IN, EXISTS, sub queries restrictions, Nested sub queries, correlated sub queries, queries with modified comparison operations, SELECT INTO operation, UNION operation. Sub queries in the HAVING clause.	**15 Lectures**
Unit III	**Implementing Indexes, Views and procedures** **(a) File Organization and Indexing:** Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes using SQL. **(b) Views:** Meaning of view, Data independence provided by views, creating, altering dropping, renaming and manipulating views using SQL. **(c) Stored Procedures:** Types and benefits of stored procedures, creating stored procedures using SQL, executing stored procedures: Automatically executing stored procedures, altering stored procedures, viewing stored procedures. **(d) Triggers:** Concept of triggers, Implementing triggers in SQL: creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers.	**15 Lectures**

References:

1. Database Management Systems- Ramakrishnam, Gehrke , McGraw- Hill.
2. SQL, PL/SQL The Programming language of Oracle- Bayross, B.P.B. Publications.
3. Professional SQL server 2000 Programming – Rob Vieira, Wrox Press Ltd, Shroff.

Additional References:

1. SQL server 2000 black book- Patrick Dalton and Paul Whitehead, Dreamtech Press.
2. *Elsmasri and Navathe, "Fundamentals of Database Systems", Pearson Education.*
3. Peter Rob and Coronel, *"Database Systems, Design, Implementation and Management", Thomson Learning*
4. C.J.Date, Longman, "Introduction to database Systems", *Pearson Education.*
5. Jeffrey D. Ullman, Jennifer Widom, "A First Course in Database Systems", Pearson Education.
6. Martin Gruber, "Understanding SQL",B.P.B. Publications.

USCSPG1	**PRACTICAL**	**3 Credits**
Units I to III P	**(I) Practical List for Discrete Mathematics** **1.** Examples of partial ordering sets, Hasse diagram. **2.** Examples on Lattices. **3.** Formulate and solve recurrence relation. **4.** Develop Warshall's and Shortest Path algorithm and implement using C. **5.** Implement operations on graph in C. **6.** Develop Breadth and Depth First search algorithms and implement using C. **7.** Create a link list, inserting items into a link list, deleting an item from a link list **8.** Implement in C for searching and inserting in binary search trees and deleting in a binary search tree. **9.** Examples based on different counting principles. **10.** Examples based on Finite state Automata, and Finite state machines. **(II*) Practical* List for C++ Programming.** **1.** Write programs that illustrates the simple **C++** concepts without classes. **2.** Write C++ programs to illustrate the concepts pointers, functions and function overloading. **3.** Write C++ programs to illustrate the concepts: classes, friend functions. **4.** Write C++ programs to illustrate the concepts: constructors, constructor overloading and destructors. **5.** Write C++ programs to illustrate the concepts: operator overloading (both Unary and Binary). **6.** Write C++ programs to illustrate the concepts: Simple, Multiple, Multilevel inheritance. **7.** Write C++ programs to illustrate the concepts:	**45 Lectures**

	Polymorphism (Virtual functions, Pure Virtual functions). **8.** Write programs to illustrate the file handling in C++. **9.** Write programs to illustrate the templates in C++. **10.** Write programs to illustrate the exceptions in C++. **(III) Practical List for Database Management Systems -I** **1.** Creating a single table without constraints and firing queries. **2.** Queries containing aggregate, string and date functions fired on a single table. **3.** Creating single table with constraints and executing queries. **4.** Updating tables, altering table structure and deleting table Creating and altering a single table and executing queries. **5.** Joining tables and processing queries. **6.** For given scenario draw E-R diagram and convert entities and relationships to table. Write relational algebra queries and convert to SQL queries on these tables. **7.** Creating, dropping and maintaining indexes. **8.** Create and manage views and process queries on views. **9.** Creating stored procedures, executing procedures, deleting procedures. **10.** Creating with or without enforcing data integrity through triggers, nested triggers, viewing, modifying and deleting triggers.	

SEMESTER II

COURSE CODE	COURSE/PAPER	LECTURES /WEEK	CREDITS
USCS201	**Computer Graphics**	**3**	**2**
USCS202	**Java Programming**	**3**	**2**
USCS203	**Data Base Management Systems -I**	**3**	**2**
USCSPG2	**Practical - II**	**9**	**3**

Practical- II: 3 lecture periods per course per week per batch. All three lecture periods of the practical shall be conducted in succession together on a single day.

USCS 201	Computer Graphics	2 Credits (45 lectures)
UNIT I	**Introduction to Computer Graphics** **(a)** Introduction to Computer graphics and its applications, Elements of graphics Displays. **(b) Scan Conversion of lines:** Digital Differential Analyzer(DDA) algorithm, Bresenhams' Line drawing algorithm **(c) Scan Conversion of a circle:** Bresenhams' method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm. **(d)** Introduction to Computer Graphics libraries in C.	15 lectures
Unit II	**Design and Visualization** **(a) Viewing and Clipping** Introduction to Viewing and Clipping, Window to Viewport mapping **2D Clipping system:** Point clipping, Inside-Outside Test, Introduction to Line Clipping- Mid-Point Subdivision Clipping Algorithm, Cohen-Sutherland Clipping algorithm. **Introduction to Polygon Clipping:** Sutherland-Hodgeman Algorithm. Character Clipping **(b) Curves and Object design** Introduction to Modeling of object primitives, Space Curve representation Cubic Splines, Bezier curves, Properties of Bezier curves, B-Spline curves, comparison of Bezier curves and B-Spline curves **Surface Generation and Object Design**: Wire frame model, Surface of Revolution, Sweep surface design, Quadric Curved surfaces.	15 lectures

Unit III	**Advanced Computer Graphics** **(a) Object Rendering** **Visible and Hidden Surfaces**: Introduction to hidden lines and surfaces, Image and Object space algorithm, Floating Horizon Algorithm, Painters algorithm, Z-Buffer algorithm **Object Rendering Models:** Introduction to object rendering, Illumination Model, **Shading Techniques:** Gouraund Shading, Phong Shading. Transparency effect, Introduction to shadows, Texture mapping (**b) Animation and Virtual Reality** **Animation and Virtual reality:** Introduction to Computer Animation and Multimedia systems: Components of Animation system, Keyframing, Kinematics and Inverse Kinematics, Introduction to Morphing Introduction to Virtual Reality and Special Effects	15 lectures

References:

1. Procedural elements of Computer Graphics, David F. Rogers, Tata McGraw Hill.
2. Computer Graphics, Donald Hearn, M P. Baker, PHI.
3. Computer Graphics: A programming Approach, Steven Harrington, McGraw-Hill.

Additional References:

1. Computer Graphics: A programming Approach, Steven Harrington, McGraw-Hill.

2. Theory and Problems of Computer Graphics, Zhigang Xiang, Roy, plastock, Schaum's outline series, McGraw-Hill.

USCS 202	**Java Programming**	**2 Credits (45 Lectures)**
Unit I	**Introduction to Java Programming** **Introduction:** History of Java, Java features, different types of Java programs, Differentiate Java with C and C++, JVM, JIT and JRE. **Java Basics:** Variables and data types, declaring variables, literals: numeric, Boolean, character and string literals, keywords, type conversion and casting. Standard default values.	15 lectures

	Java Operators: Arithmetic, relational, logical, assignment, increment and decrement, conditional, bitwise, precedence and order of evaluation, statement and expressions, string arithmetic. **Loops and Controls:** Control statements for decision making: select statements (if statement, if …else… statement, if … else … if …statement, switch statement), goto statement, looping (while loop, do … while loop and for loop), nested loops, breaking out of loops (break and continue statements), labeled loops. **Arrays and Strings:** one and two -dimensional array, creating an array, strings, stringbuffer. **Introduction to Classes:** Defining a class, creating instance and class members: creating object of a class, accessing instance variables of a class, creating methods, naming methods of a class, accessing methods of a class, constructor, parameterized constructor, 'this' keyword, garbage collection, finalize() method, methods overloading, constructor overloading, nested and inner classes, static member. **Visibility control:** public access, friendly access, protected access, private access, private protected access.	
Unit II	**Inheritance, Interface and Packages** **Inheritance:** Various types of inheritance, super and subclasses, keywords- 'extends', 'super', constructor chaining, method overriding, final variables and methods, final classes, abstract method and classes, dynamic method dispatch. **Interface:** Defining interfaces, extending interfaces, implementing interfaces **Packages:** System packages, using system package, naming conventions, creating packages, accessing a package, using a package, adding a class to a package **Exception Handling:** Exception-handling fundamentals, Exception types, Uncaught exceptions, Using try and catch, Multiple catch clauses, Nested try statements, use of throw, throws and finally keywords, Java's Built-in exceptions, User defined exception, Chained Exception. **Streams and File I/O:** Concept of streams, stream classes, byte stream classes: InputStream, and OutputStream, character	15 lectures

	stream classes: Reader and Writer, Difference between byte stream classes and character stream classes, other I/O classes, File class, Reading/writing bytes/characters, random access file, serialization.	
Unit III	**Java Applets and Graphics Programming** **Applets:** Difference of applet and application, creating applets, applet life cycle, passing parameters to applets. **Graphics, Fonts and Color:** The graphics class, painting, repainting and updating an applet, sizing graphics. Font class, draw graphical figures - lines and rectangle, circle and ellipse, drawing arcs, drawing polygons. Working with Colors: Color methods, setting the paint mode. **AWT package:** Window fundamentals: Component, Container, Panel, Window, Frame, and Canvas. AWT Controls: labels, buttons, textfield, textarea, checkboxes, checkboxgroup, choice, and list. Layout Managers: FlowLayout, BorderLayout, GridLayout. **Event Handling:** The Delegation Event Model, Event classes (ActionEvent, FocusEvent, InputEvent, ItemEvent, KeyEvent, MouseEvent, MouseWheelEvent, TextEvent, WindowEvent) and various listener interfaces (ActionListener, FocusListener, ItemListener, KeyListener, MouseListener, MouseMotionListener, MouseWheelListener, TextListener, WindowFocusListener, WindowListener)	15 lectures

Main References:

(1) Java2: **The Complete Reference** – Tata McGraw Hill, Fifth edition
(2) **Programming with Java A primer**, by E. Balagurusamy 3rd Edition

Additional References:

(1) Programming in Java, Schaum Series.
(2) Java2 Programming – Black Book, Dreamtech Press.

USCS 203	Software Engineering	2 Credits (45 Lectures)
Unit I	**Introduction to Software Engineering** What is software? Types of software, Software Quality factors, What is software engineering? Introduction to Soft Eng & its objectives, The general systems approach to problem solving. The three approaches to software systems development - The Structured approach, the Object Oriented Approach and the Information Engineering Approach. **Software Process:** SDLC -Requirement Analysis, Software design, coding, testing, maintenance etc. **Software Development Life Cycle Models** - Waterfall Model, Prototyping Model, RAD Model, Incremental Model, Spiral Model, Component Based Model, Their features, strengths, weaknesses and differences between them, Fourth Generation Techniques Project Management Process, Role of metrics & models in project management **Project Feasibility Study -** Operational, Technical, Economic, Organizational and Cultural feasibility. Defining project costs and project benefits. Cost/Benefit Analysis for a project	**15 lectures**
Unit II	**System Analysis** **Investigating System Requirements** – Software Requirement Specification Document, Need of SRS, Characteristics & Components of SRS, Stakeholders, Identifying requirements using various techniques (such as Questionnaires, reviewing reports/forms, interviews, workflows etc), building prototypes, Structured Walkthroughs, **Modeling System Requirements** –Conceptual modeling **Data Modeling -** Data entities, Attributes, Relationships, Cardinality, ERD **Process Modeling** - Developing Data Flow Diagrams, Level of abstraction, Context diagram, Top level DFD, DFD fragments, Physical and Logical DFD, Data Dictionary, Events, Event Table **Logic Modeling**- Decision Tables, Decision Trees, Structured English & Pseudo-code **Object Oriented Modeling:** Object Model, Elements of Object Model, Basic Principles of OO Approach, Association, Generalization, Specialization, Aggregation **UML:** Basics of UML, Types of UML Diagrams, Use Case Diagram, Class Diagram, Object Diagram, Sequence diagram & Collaboration diagram, State Transition & State chart diagrams	**15 lectures**

Unit III	**System Design & Coding** **System Design** - Problem partitioning, Abstraction, Top-down & Bottom-up Design, Function-Oriented & Object-oriented Design, Problem Partitioning, Abstraction & its type(Data & Function), Modularity, Coupling, Cohesion, Drawing Structure Charts & Flow charts, UML Activity Diagram, Component Diagram, Package & Deployment Diagram **Designing Databases:** Converting ERD to Databases, Introduction to OO Databases, Object-Relational Databases, **User Interface Design -** Designing System Input, output, User Interface, Characteristics of good interfaces **Coding –** Top down VS Bottom up strategies, structured programming & object oriented programming, Information hiding, programming styles, Internal documentation **Verification & Validation:** What is V&V, Types of V&V activities such as inspection, review, walkthrough, V&V with respect to requirements, system analysis, system design & coding.	**15 lectures**

References:

1. Software Engineering – A Practitioner's Approach – 6th Edition,Roger Pressman, McGraw Hill.
2. Integrated Approach to Software Engineering (3rd Edition) - Pankaj Jalote (Narosa)
3. System Analysis & Design in a Changing World, Satzinger, Jackson, Burd – Thompson Learning
4. Instant UML, Pierre-Alain Muller, Wrox.

Additional References:

1. UML User's Guide – By Grady Booch, Ivar Jacobson, James Rambaugh
2. OO Modeling & Design with UML, IInd Edition, Blaha, Rambaugh, Pearson

USCSPG2	PRACTICAL	3 Credits

Unit I P, Unit II P and Unit III P	**(I) Practical List for Computer Graphics** **1.** Study and use of graphics library for drawing primitive images. **2.** Write a program to a line using following algorithms: (i) Digital Differential Analyzer (DDA), (ii) Bresenham's Line drawing method **3.** Write a program to draw circle using the following algorithms (I) Bresenham's circle drawing method, (ii) Midpoint circle drawing algorithm **4.** Write a program to demonstrate the following primitive 2D transformations on a unit square or a triangle: (i) scaling in X or Y or Both directions, (ii) translation in X or Y or Both directions, (iii) shear transformation, (iv) reflection about an axis, (v) rotation transformation **5.** Write a program to rotate a line about an arbitrary point (x,y). *[Use shift of origin]* ***6.*** Write a program to draw an origin centered 3D cube on the screen. (Use shift of origin and bring origin of coordinate at the center of the screen) **7**. Implement line clipping algorithm using (i) Mid-Point Subdivision Clipping Algorithm, (ii) Cohen-Sutherland Clipping algorithm, **8.** Write a program to generate a Bezier curve for the N input control points. (take n= 4, 5 and 6). **9.** Generate cylinder as surface of revolution by rotating a line around an axis. (use delay for better visualization) **10.** Write a program **(a)** To implement bouncing ball problem using two balls. **(b)** To generate an animating clock. **(c)** To draw animated scenery using primitive structures like points, lines, curves, circles, etc showing rising sun and a wind mill. Other objects in the scene include trees, house, roads etc. **(II) Practical List for Java Programming** **1.** Write a Java program to create a Java class: **(a)** without instance variables and methods, **(b)** with instance variables and without methods, **(c)** without instance variables and with methods. **(d)** with instance variables and methods. **2.** Write a Java program that illustrates the concepts of selection statement, looping, nested loops, breaking out of	45 periods

loop.

3. Write a Java program that illustrates the concepts of one, two dimension arrays and strings.
4. Write a Java program that illustrates the concepts of Java class that includes
   (a) constructor with and without parameters.
   (b) Overloading methods.
   (c) Overriding methods
5. Write a Java program to demonstrate inheritance by creating suitable classes.
6. Create a Java package, interface and implement in Java program.
7. Write a program that illustrates the error handling using exception handling.
8. Write a program that illustrates the concepts of stream classes.
9. Write a Java applet to demonstrate graphics, Font and Color classes.
10. Write a Java program to illustrate AWT package, Event classes and listeners.

(III) Practical List for Software Engineering

Following Case Studies are recommended: (To be solved in group of max of 3 to 4 students)

Developing the system for the following with SSAD and OOAD approach:

- **a.** Library System,
- **b.** Reservation System,
- **c.** Inventory System,
- **d.** Hospital management System
- **e.** Any other system

Note: - Multiple sessions may be required for completing a practical

1. Problem Definition - Identifying & Understanding the system, its functions, desired inputs, outputs etc.
2. Conducting Feasibility Study – Deciding S/W, H/W requirements, Type of system (Single-User/Multi-user etc), Limitations of current system, Benefits of the proposed system etc.
3. Requirement Analysis, Interviews, Questionnaire, Creating SRS
4. Drawing ERD & converting to tables
5. Drawing Context Diagram, DFDs for understanding process flow

	6. Drawing Use Case Diagram **7.** Drawing Class, Object Diagrams, **8.** Drawing Sequence & Collaboration Diagrams, **9.** Drawing State Transition, State chart diagrams, **10.** Drawing Activity, Component, Package Diagrams	

Scheme of Examination

The performance of the learners shall be evaluated into two parts. The learner's performance shall be assessed by Internal Assessment with 40% marks in the first part by conducting the Semester End Examinations with 60% marks in the second part. The allocation of marks for the Internal Assessment and Semester End Examinations are as shown below:-

[A] Internal assessment (i) Theory 40 %

Sr. No.	Evaluation Type	Marks
1.	Two Assignments/Case study/Project	20
2.	One class Test (multiple choice questions objective)	10
3.	Active participation in routine class instructional deliveries(case studies/ seminars//presentation)	05
4.	Overall conduct as a responsible student, mannerism and articulation and exhibit of leadership qualities in organizing related academic actives.	05

[A] Internal assessment (ii) Practical 40%

Sr. No.	Evaluation Type	Marks
1.	Two best practicals, test on practical, skill/demo experiments, error calculations, graph plotting.	10
2.	Journal	05
3.	Viva	05

- Each practical examination can be conducted out of 50 marks, 20 internal and 30 external.
- Minimum 75 % practical from each paper are required to be completed and written in the journal.

[B] External Theory examination 60 %

(a) Duration - These examinations shall be of 2 Hours duration for each paper.

(b) Theory Question Paper Pattern (For Paper I, II and III) :-

1) There shall be four questions each of 15 marks. On each unit there will be one question and the fourth one will be based on entire syllabus.
2) All questions shall be compulsory with internal choice within the questions. (Each question will be of 20 to 23 marks with options.)
3) Question may be subdivided into sub-questions a, b, c … and the allocation of marks depend on the weightage of the topic.

****