Semester III

1. Logic and Discrete Mathematics
2. Computer Graphics
3. Advanced SQL
4. Object Oriented Programming with C++
5. Modern Operating Systems

Semester IV

1. Software Engineering
2. Multimedia
3. Java and Data Structures
4. Quantitative Techniques
5. Embedded Systems

CLASS: B. Sc (Information technology)		Semester – III	
COURSE: Logic and Discrete Mathematics			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Set Theory:** Fundamentals - Sets and subsets, Venn Diagrams, Operations on sets, Laws of Set Theory, Power Sets and Products, Partition of sets, The principle of Inclusion-Exclusion. **Logic:** Propositions and Logical operations, Truth tables, Equivalence, Implications, Laws of Logic, Normal forms, Predicates and quantifiers, Mathematical Induction
Unit-II	**Relations, diagraphs and lattices:** – Product sets and partitions, relations and digraphs, paths in relations and digraphs, properties of relations, equivalence and partially ordered relations, computer representation of relations and digraphs, manipulation of relations, Transitive closure and Warshall's algorithm, Posets and Hasse Diagrams, Lattice.
Unit-III	**Functions and Pigeon Hole Principle:** Definitions and types of functions: injective, surjective and bijective, Composition, identity and inverse, Pigeon hole principle.
Unit-IV	**Graphs and Trees:** Graphs, Euler paths and circuits, Hamiltonian paths and circuits, Planer graphs, coloring graphs, Isomorphism of Graphs. **Trees:** Trees, rooted trees and path length in rooted trees, Spanning tree and Minimal Spanning tree, Isomorphism of trees, Weighted trees and Prefix Codes.
Unit-V	**Algebraic Structures:** Algebraic structures with one binary operation – semi groups, monoids and groups, Product and quotient of algebraic structures, Isomorphism, homomorphism, automorphism, Cyclic groups, Normal sub group, codes and group codes, Algebraic structures with two binary operations – rings, integral domains and fields. Ring homomorphism and Isomorphism.
Unit-VI	**Generating Functions and Recurrence relations:** Series and Sequences, Generating Functions, Recurrence relations, Applications, Solving difference equations, Fibonacci.

Books:

Discrete mathematical structures by B Kolman RC Busby, S Ross PHI Pvt. Ltd.
Discrete mathematical structures by RM somasundaram (PHI) EEE edition

References:

Discrete structures by Liu, TATAMCGRAW-HILL
Digital Logic John M Yarbrough Brooks/cole, Thompson Learning
*Discrete Mathematics and its Applications,*Kenneth H. Rosen, TATAMCGRAW-HILL
Discrete Mathematics for computer scientists and Mathematicians, Joe L.Mott, Abraham Kandel Theodore P. Baker, Prentice-Hall of India Pvt. Ltd.
Discrete Mathematics With Applications, Susanna S. Epp, Books/Cole Publishing Company
Discrete Mathematilcs, Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson, TATAMCGRAW-HILL

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.
Tutorial: At least **t**hree tutorials based on above syllabus must be conducted.

CLASS: B. Sc (Information technology)		Semester – III	
COURSE: Computer Graphics			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Introduction Computer Graphics and Primitive Algorithms:** Introduction to Image and Objects, Image Representation, Basic Graphics Pipeline, Bitmap and Vector-Based Graphics, Applications of Computer Graphics, Display Devices, Cathode Ray Tubes, Raster-Scan Display, Random-Scan Display, Flat Panel Display, Input Technology, Coordinate System Overview, **Scan-Conversion of graphics primitives:** Scan-Conversion of a Lines (Digital Differential Analyzer Algorithm, Bresenham's Line-Drawing Algorithm, Scan-Conversion of Circle and Ellipse (Bresenham's Method of Circle Drawing, Midpoint Circle Algorithm), Drawing Ellipses and Other Conics.
Unit-II	**Two Dimensional Transformation:** Introduction to transformations, Transformation Matrix, Types of Transformations in Two-Dimensional Graphics: Identity Transformation, Scaling, Reflection, Shear Transformations, Rotation, Translation, Rotation about an Arbitrary Point, Combined Transformation, Homogeneous Coordinates, 2D Transformations using Homogeneous Coordinates
Unit-III	**Three-dimensional transformations**, Objects in Homogeneous Coordinates, Three-Dimensional Transformations: Scaling, Translation, Rotation, Shear Transformations, Reflection, World Coordinates and Viewing Coordinates, Projection, Parallel Projection, Perspective Projection.
Unit-IV	**Viewing and Solid Area Scan-Conversion:** Introduction to viewing and clipping, Viewing Transformation in Two Dimensions, Introduction to Clipping, Two-Dimensional Clipping, Point Clipping, Line Clipping, Introduction to a Polygon Clipping, Viewing and Clipping in Three Dimensions, Three-Dimensional Viewing Transformations, Text Clipping **Introduction to Solid Area Scan-Conversion**, Inside–Outside Test, Winding Number Method and Coherence Property, Polygon Filling, Seed Fill Algorithm, Scan-Line Algorithm, Priority Algorithm, Scan Conversion of Character, Aliasing, Anti-Aliasing, Halftoning, Thresholding and Dithering
Unit-V	**Introduction to curves**, Curve Continuity, Conic Curves, Piecewise Curve Design, Parametric Curve Design, Spline Curve Representation, Bezier Curves, B-Spline Curves, Fractals and its applications. **Surface Design**: Bilinear Surfaces, Ruled Surfaces, Developable Surfaces, Coons Patch, Sweep Surfaces, Surface of Revolution, Quadric Surfaces, Constructive Solid Geometry, Bezier Surfaces, B-Spline Surfaces, Subdivision Surfaces **Visible Surfaces:** Introduction to visible and hidden surfaces, Coherence for visibility, Extents and Bounding Volumes, Back Face Culling, Painter's Algorithm, Z-Buffer Algorithm, Floating Horizon Algorithm, Roberts Algorithm.
Unit-VI	**Object Rendering:** Introduction Object-Rendering, Light Modeling Techniques, Illumination Model, Shading, Flat Shading, Polygon Mesh Shading, Gaurand Shading Model, Phong Shading, Transparency Effect, Shadows, Texture and Object Representation, Ray Tracing, Ray Casting, Radiosity, Color Models. **Introduction to animation**, Key-Frame Animation, Construction of an Animation Sequence, Motion Control Methods, Procedural Animation, Key-Frame Animation vs. Procedural Animation, Introduction to Morphing, Three-Dimensional Morphing

Books:

Computer Graphics, R. K. Maurya, John Wiley.

Mathematical elements of Computer Graphics, David F. Rogers, J. Alan Adams, TATAMCGRAW-HILL

Procedural elements of Computer Graphics, David F. Rogers, Tata McGraw-Hill.

Reference:

Computer Graphics, Donald Hearn and M. Pauline Baker, Prentice Hall of India

Computer Graphics, Steven Harrington, McGraw-Hill

Computer Graphics Principles and Practice, J.D. Foley, A. Van Dam, S.K. Feiner and R.L. Phillips, Addision Wesley

Principles of Interactive Computer Graphics, Willaim M. Newman, Robert F. Sproull, TATAMCGRAW-HILL

Introduction to Computer Graphics, J.D. Foley, A. Van Dam, S.K. Feiner, J.F. Hughes and R.L. Phillips, Addision Wesley

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Suggested Practical:

Should contain at least 10 programs development in C++. Some Sample practical are listed below**.**

1. Write a program with menu option to input the line coordinates from the user to generate a line using Bresenham's method and DDA algorithm. Compare the lines for their values on the line.
2. Develop a program to generate a complete circle based on
   a. Bresenham's circle algorithm b. Midpoint Circle Algorithm
3. Implement the Bresenham's/DDA algorithm for drawing line (programmer is expected to shift the origin to the center of the screen and divide the screen into required quadrants).
4. Write a program to implement a stretch band effect. (A user will click on the screen and drag the mouse/arrow keys over the screen coordinates. The line should be updated like rubber-band and on the right-click gets fixed).
5. Write program to perform the following 2D and 3D transformations on the given input figure
   a. Rotate through θ.
   b. Reflection
   c. Scaling
   d. Translation.
6. Write a program to demonstrate shear transformation in different directions on a unit square situated at the origin.
7. Develop a program to clip a line using Cohen-Sutherland line clipping algorithm between $(x_1,y_1)(x_2, y_2)$ against a window $(x_{min} , y_{min})(x_{max} , y_{max})$.
8. Write a program to implement polygon filling.
9. Write a program to generate a 2D/3D fractal figures (Sierpinski triangle, Cantor set, tree etc).
10. Write a program to draw Bezier and B-Spline Curves with interactive user inputs for control polygon defining the shape of the curve.
11. Write a program to demonstrate 2D animation such as clock simulation or rising sun
12. Write a program to implement the bouncing ball inside a defined rectangular window.

CLASS: B. Sc (Information technology)		Semester – III	
COURSE: Advanced SQL			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Structured Query Language:** Writing Basic SQL Select Statements**,** Restricting and Sorting Data, Single-Row Functions**,** Joins (Displaying Data from Multiple Tables), Aggregating Data using Group Functions, Subqueries, Manipulating Data, Creating and Managing Tables, Including Constraints, Creating Views, Creating Other Database Objects(Sequences, Indexes and Synonyms)
Unit-II	**Advanced SQL:** Controlling User Access, Using SET operators, DateTime Functions, Enhancements to Group by clause(cube, Rollup and Grouping), Advanced Subqueries (Multiple column subqueries, Subqueries in FROM clause, Scalar and correlated subqueries), WITH Clause, Hierarchical retrieval,
Unit-III	**PLSQL:** Introduction, Overview and benefits of PL/SQL, Subprograms, types of PL/SQL blocks, Simple Anonymous Block, Identifiers, types of Identifiers, Declarative Section, variables, Scalar Data Types, The %TYPE Attribute, Bind Variables, Sequences in PL/SQL Expressions, Executable Statements, PL/SQL Block Syntax, Comment the Code, Deployment of SQL Functions in PL/SQL, Convert Data Types, Nested Blocks, Operators. Interaction with the Oracle Server, Invoke SELECT Statements in PL/SQL, SQL Cursor concept, Data Manipulation in the Server using PL/SQL, SQL Cursor Attributes to Obtain Feedback on DML, Save and Discard Transactions.
Unit-IV	**Control Structures:** Conditional processing using IF Statements and CASE Statements, Loop Statement, While Loop Statement, For Loop Statement, the Continue Statement, **Composite Data Types:** PL/SQL Records, The %ROWTYPE Attribute, Insert and Update with PL/SQL Records, INDEX BY Tables, INDEX BY Table Methods, Use INDEX BY Table of Records, **Explicit Cursors**, Declare the Cursor, Open the Cursor, Fetch data from the Cursor, Close the Cursor, Cursor FOR loop, The %NOTFOUND and %ROWCOUNT Attributes, the FOR UPDATE Clause and WHERE CURRENT Clause, **Exception Handling,** Handle Exceptions with PL/SQL, Trap Predefined and non-predefined Oracle Server Errors, User-Defined Exceptions, Propagate Exceptions, RAISE_APPLICATION_ERROR Procedure,
Unit-V	**Stored Procedures:** Create a Modularized and Layered Subprogram Design, the PL/SQL Execution Environment, differences between Anonymous Blocks and Subprograms, Create, Call, and Remove Stored Procedures, Implement Procedures Parameters and Parameters Modes, View Procedure Information, Stored Functions and Debugging Subprograms, Create, Call, and Remove a Stored Function, advantages of using Stored Functions, the steps to create a stored function, Invoke User-Defined Functions in SQL Statements, Restrictions when calling Functions, Control side effects when calling Functions, View Functions Information, debug Functions and Procedures, **Packages,** advantages of Packages, components of a Package, Develop a Package, enable visibility of a Package's Components, Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer, Invoke the Package Constructs, View the PL/SQL Source Code using the Data Dictionary, Deploying Packages, Overloading Subprograms in PL/SQL, Use the STANDARD Package, Use Forward Declarations, Implement Package Functions in SQL and Restrictions, Persistent State of Packages, Persistent State of a Package Cursor, Control side effects of PL/SQL Subprograms, Invoke PL/SQL Tables of Records in Packages

Unit-VI	**Dynamic SQL:** The Execution Flow of SQL, Declare Cursor Variables, Dynamically Executing a PL/SQL Block, Configure Native Dynamic SQL to Compile PL/SQL Code, invoke DBMS_SQL Package, Implement DBMS_SQL with a Parameterized DML Statement, Dynamic SQL Functional Completeness, **Triggers,** the Trigger Event Types and Body, Business Application Scenarios for Implementing Triggers, Create DML Triggers using the CREATE TRIGGER Statement and SQL Developer, Identify the Trigger Event Types, Body, and Firing (Timing), Statement Level Triggers and Row Level Triggers, Create Instead of and Disabled Triggers, Manage, Test and Remove Triggers. Creating Compound, DDL and Event Database Triggers, Compound Trigger Structure for Tables and Views, Compound Trigger to Resolve the Mutating Table Error, Comparison of Database Triggers and Stored Procedures, Create Triggers on DDL Statements, Create Database-Event and System-Events Triggers, System Privileges Required to Manage Triggers

Books:

Murach's Oracle SQL and PLSQL by Joel Murach, Murach and Associates.

Oracle Database 11g PL/SQL Programming Workbook, ISBN: 9780070702264, By: Michael McLaughlin,John Harper, TATAMCGRAW-HILL

Reference:

Oracle PL/SQL Programming, Fifth Edition By Steven Feuerstein, Bill Pribyl

Oracle 11g: SQL Reference Oracle press

Oracle 11g: PL/SQL Reference Oracle Press.

Expert Oracle PL/SQL, By: Ron Hardman,Michael McLaughlin, TATAMCGRAW-HILL

Oracle database 11g: hands on SQL/PL SQL by Satish Asnani (PHI) EEE edition

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Suggested Practical:

Should contain at least 10 programs. Some sample practicals are listed below**.**

1. **Practical 1: Select queries and joins**
   a. Select queries on single table using alias, where and order by clause.
   b. Select queries on single table using aggregate functions and group by clause.
   c. Querying data from multiple tables using all types of joins.

2. **Practical 2: Subqueries, DML and DDL**
   a. Querying single and multiple tables using subqueries.
   b. Manipulating data (Insert, update and delete)
   c. Creating simple tables and tables with constraints.

3. **Practical 3: Creating database objects, Controlling user access and using set operators**
   a. Creating Views, Sequences, Indexes and synonyms.
   b. Granting and revoking privileges on user objects.
   c. Using set operators, date-time functions, roll up, cube and grouping sets.

4. **Practical 4: Working with advanced subqueries and WITH clause**
   a. Multiple column subqueries, subqueries in from clause,
   b. Scalar subqueries and correlated subqueries,
   c. WITH Clause and hierarchical retrieval.

5. **Practical 5: Basic PL/SQL, INDEX BY tables, PL/SQL Record and FOR loop.**
   a. Creating anonymous PL/SQL blocks.
   b. Define, create, and use INDEX BY tables and a PL/SQL record.
   c. Process a number of rows from a table and populate another table with the results using a cursor FOR loop.

6. **Practical 6: Cursors, Exceptions and procedures issuing DML and query commands.**
   a. Cursors with parameters to process a number of rows from multiple tables.
   b. Create exception handlers for specific situations.
   c. Create procedures that issue DML and query commands.

7. **Practical 7: Functions and Stored Procedures**
    a. Creating and invoking functions from SQL statements.
    b. Creating and invoking stored procedures.
    c. Re-create the source code for a procedure and a function.

8. **Practical 8: Working with packages**
    a. Create package specifications and package bodies. Invoke the constructs in the packages.
    b. Create a package containing an overloaded function.
    c. Create a one-time-only procedure within a package to populate a PL/SQL table.

9. **Practical 9: Working with Large Objects and triggers**
    a. Create a table with both BLOB and CLOB columns. Use the DBMS_LOB package to populate the table and manipulate the data.
    b. Create statement and row triggers.
    c. Create procedures that will be invoked from the triggers.

10. **Practical 10: Working with INSTEAD OF triggers, business rules and recompiling procedures, functions, packages and views.**
    a. Create instead of triggers for views.
    b. Implement a number of business rules. Create triggers for those rules that should be implemented as triggers. The triggers must execute procedures that that are placed in a package.
    c. Use the DEPTREE_FILL procedure and the IDEPTREE view to investigate dependencies in your schema. Recompile invalid procedures, functions, packages, and views.

CLASS: B. Sc (Information technology)		Semester – III	
COURSE: Object Oriented Programming with C++			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Introduction to OOPs:** Need object oriented programming, comparison of procedural and object oriented approach, characteristics of OOPs – object , classes , polymorphism, inheritance, reusability, data hiding and abstraction, applications of OOPs
Unit-II	**Classes and Objects:** Class declaration, constructors, constructor initialization lists, access functions, private member functions, the copy constructor, the class destructor ,constant objects, structures, pointers to objects, static data members, static function members
Unit-III	**Operator Overloading:** overloading the assignment operator, the this pointer, overloading arithmetic operators, overloading the arithmetic assignment, operators, overloading the relational operators ,overloading the stream operators, conversion operators ,overloading the increment and decrement operators, overloading the subscript operator
Unit-IV	**Composition and Inheritance:** inheritance, protected class members , overriding and dominating inherited members, private access verses protected access, virtual functions and polymorphism, virtual destructors, abstract base classes **File Handling:** Classes for file stream operations, opening and closing a file, detecting end of file, file modes, file pointers and their manipulations, sequential input and output operations, random access, file operations error handling, command line argument
Unit-V	**Strings and Streams:** the string class interface, the constructors and destructor , the copy constructor, the assignment operator, the addition operator , an append operator, access functions , the comparison operators, stream operators, stream classes, the ios class, ios format flags, ios state , variables ,the istream and ostream classes, unformatted input functions , unformatted output functions, stream manipulators.
Unit-VI	**Templates and Iterators:** function templates, class templates, container classes, subclass templates, passing template classes to template parameters, iterator classes **Libraries:** the standard C++ library, proprietary libraries, contents of the standard c headers, string streams, file processing, the standard template library

Books:

SCHAUM'S OUTLINE OF THEORY AND PROBLEMS of PROGRAMMING WITH C++ JOHN R. HUBBARD, TATAMCGRAW-HILL

Object Oriented Programming with C++, E.Balagurusamy,Fourth Edition, TATAMCGRAW-HILL.

Object Oriented Programming with C++, by P. Sarang 2nd Edition,(PHI) EEE edition

Reference:

C++ programming , 3rd Edition, Bjarne Stroustrup

Mastering C++, 2nd Edition, Venugopalan, TataMcgrawHill

C++ Programming, , Robert Lafore,

C++ for Beginners, P. M. Harwani, X-Team Series,

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Suggested Practical:
Should contain at least 10 programs. Some sample practicals are listed below**.**

1 **Implement the following**

a. Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will be private method.
b. Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively.Where getData() will be private method.
c. Design the class Demo which will contain the following methods: readNo() ,factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not.Where readNo() will be private method.

2 Implement the following

a. Write a friend function for adding the two complex numbers, using a single class.
b. Write a friend function for adding the two different distances and display its sum, using two classes.
c. Write a friend function for adding the two matrix from two different classes and display its sum.

3 Implement the following

a. Design a class Complex for adding the two complex numbers and also show the use of constructor.
b. Design a class Geometry containing the methods area() and volume() and also overload the area() function .
c. Design a class StaticDemo to show the implementation of static variable and static function.

4 Implement the following

a. Overload the operator unary(-) for demonstrating operator overloading.
b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.
c. Overload the + for concatenating the two strings. For e.g “c” + “++” = c++

5 Implement the following

a. Design a class for single level inheritance using public and private type derivation.
b. Design a class for multiple inheritance.
c. Implement the hierarchical inheritance.

6. Implement the following

a. Implement the concept of method overriding.
b. Show the use of virtual function
c. Show the implementation of abstract class.

7. Implement the following

a. String operations for string length , string concatenation
b. String operations for string reverse, string comparison,
c. Console formatting functions.

8. Implement the following:

a. Show the implementation of exception handling
b. Show the implementation for exception handling for strings
c. Show the implementation of exception handling for using the pointers.

9. Show the implementation

a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
b. Design a class to handle multiple files and file operations
c. Design a editor for appending and editing the files

10. Show the implementation for the following

a. Show the implementation of template class library for swap function.
b. Design the template class library for sorting ascending to descending and vice-versa
c. Design the template class library for concatenating two strings

CLASS: B. Sc (Information technology)		Semester – III	
COURSE: Modern Operating Systems			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Introduction to Operating Systems:** OS and Computer System, System performance, Classes of OS, Batch processing, time-sharing, multiprocessing, real time, distributed and modern operating systems, Desktop Systems, Handheld Systems, Clustered Systems, Assemblers, Compilers and Interpreters, Linkers.
Unit-II	**Operating-System Structures:** Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Virtual Machines, Operating-System Generation, System Boot.
Unit-III	**Processes and Process Synchronization:** Process Concept, Process Scheduling, Scheduling Criteria, Scheduling Algorithms, Operations on Processes, Interprocess Communication, Multithreading Models, Threading Issues, Thread Scheduling, Communication in Client–Server Systems, The Critical-Section Problem, Peterson's Solution, Semaphores.
Unit-IV	**Memory Management:** Memory management without swapping or paging; Swapping, Virtual Memory, Page replacement algorithms, Modeling paging algorithms, Design issues for paging systems, segmentation
Unit-V	**File-System Interface and Implementation:** File Concept, File-System Mounting, Free-SpaceManagement, File Sharing, NFS. **Mass-Storage Structure:** Disk Structure, Disk Management, Swap-Space Management, RAID Structure, Stable-Storage Implementation. Deadlocks , Deadlock detection and recovery, avoidance and prevention
Unit-VI	**I/O Systems:** Application I/O Interface, Transforming I/O Requests to Hardware Operations, STREAMS, Performance. **Protection and Security:** Principles of Protection, Domain of Protection, Access Matrix, Access Control, Capability-Based Systems, Language-Based Protection, The Security Problem, System and Network Threats, Implementing Security Defenses.

Books:
Modern Operating Systems, Andrew Tanenbaum,
Operating Systems, 2nd Edition, K. A.Sumitra Devi and N.P Banashree, SPD
Operating System Concepts, 8th Edition, Abraham Silberschatz, Peter B.Galvin, Greg Gagne, Wiley publication

Reference:
Operating Systems- A concept based approach , 2nd Edition, D.M. Dhamdhere, McGrawHill publications
Operating Systems, 3rd Edition , Godbole and Kahate, McGrawHill publications.

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Tutorial: At least **t**hree tutorials based on above syllabus must be conducted.

Case Studies (Suggested):

a) MS-DOS
b) Windows NT
c) Windows 2008 Server
d) Windows 7
e) Unix
f) Linux
g) OS/2
h) MAC OS
i) Symbian
j) Chrome
k) Android

CLASS: B. Sc (Information technology)		Semester – IV	
COURSE: Software Engineering			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**An Introduction** : To Software, Software Engineering, Software Process, Software Engineering Methods; CASE Tools, Attributes of good software. **Socio-technical system :** Essential characteristics of socio technical systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computers, Dealing Legacy Systems. **Critical system** : Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems
Unit-II	**Software processes :** Fundamental activities of software process, Different software process models, Process Iteration and Activities, The Rational Unified Process, CASE in detail. **Project Management** : Software Project Management, Management activities, Project Planning, Project Scheduling, Risk Management. **Software Requirements:** Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements
Unit-III	**Requirements Engineering Processes** : Feasibility study, Requirements elicitation and anlaysis, Requirements Validations, Requirements Management. **System Models** : Models and its types, Context Models, Behavioural Models, Data Models, Object Models, Structured Methods. **Architectural Design :** Architectural Design Decisions, System Organisation, Modular Decomposition Styles, Control Styles, Reference Architectures
Unit-IV	**Application Architectures** : Data Processing Systems, Transaction Processing Systems, Event Processing Systems, Language Processing Systems **Object Oriented Design** : Objects and Object Classes, An object Oriented Design Process, Design Evolution **User Interface Design** : Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation **Rapid Software Development :** Agile Methods, Extreme Programming, Rapid Application Development, Software Prototyping
Unit-V	**Component based Software Engineering :** Components and Component models, The CBSE Process, Component Composition. **Verification and Validation :** Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods. **Software Testing :** System Testing, Component Testing, Test Case Design, Test Automation. **Software Cost Estimation :** Software Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing
Unit-VI	**Quality Management :** Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics **Process Improvement :** Process and product quality, Process Classification, Process Measurement, Process Analysis and Modeling, Process Change, The CMMI Process Improvement Framework. **Security Engineering :** Security Concepts, Security Risk Management, Design for Security, System Survivability. **Service Oriented Software Engineering** : Services as reusable components, Service Engineering, Software Development with Services

Books:
Software Engineering, "Ian Somerville", 8th edition, Pearson Education.
Software Engineering, Pankaj Jalote, Narosa Publication
Reference:
Software Design, "D.Budgen", 2nd edition, Pearson education.
Software engineering, A practitioner's approach, Roger Pressman, TATAMCGRAW-HILL.
Software Engineering by KL James, PHI(2009) EEE edition
Software Engineering principles and practice by WS Jawadekar TATAMCGRAW-HILL

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Tutorial: At least **t**hree tutorials based on above syllabus must be conducted.

Case Studies (Suggested)
1. Project Initiation and scheduling.
2. Analyzing a system and specifying the requirements
    a. Structured Approach
    b. Object oriented Approach
3. Project Cost Estimation
4. Designing the database design
5. Designing the User interface design
6. Use of testing methodologies
7. Cost Estimation Techniques
8. Cost benefit Analysis

CLASS: B. Sc (Information technology)		Semester – IV	
COURSE: Multimedia			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Introduction:** What is multimedia? Defining the scope of multimedia. Applications of multimedia, hardware and software requirements, multimedia database.
Unit-II	**Digital representation:** Introduction, Analog representation, waves, digital representation, need for digital representation, A to D conversion, D to A conversion, relation between sampling rate and bit depth, Quantization error, Fourier representation, pulse modulation. Importance and drawback of digital representation.
Unit-III	**Text and Image:** Introduction, Types of text, Font, insertion, compression,File formats. Types of images, colour models, Basic steps for image processing, principle and working of scanner and digital camera, Gamma and gamma correction.
Unit-IV	**Audio and Video technology:** Fundamental characteristics of sound, psycho-acoustics, Raster scanning principles, sensors for TV cameras, color fundamentals, additive and subtractive color mixing, Liquid crystal display (LCD), Plasma Display Panel (PDP), file formats
Unit-V	**Compression and coding**: What is compression? Need for compression, Types of compression- basic compression techniques-run length, Huffman's coding, JPEG, zip coding. Overview of Image and Video compression techniques.
Unit-VI	**Multimedia presentation and authoring:** Overview, multimedia authoring metaphor, multimedia production, presentation and automatic authoring, Design paradigms and user interface, overview of tools like adobe premier, director, flash and dreamweaver. Barriers to wide spread use.

Books:

Principles of Multimedia by Ranjan Parekh. TATAMCGRAW-HILL

Reference:

Multimedia Systems Design by Prabhat K. Andleigh and Kiran Thakrar-PHI publication
Multimedia systems by John F. Koegal Buford-Pearson Education.
Fundamentals of multimedia by Ze-Nian Li and MS Drew. PHI EEE edition.

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Tutorial: At least **t**hree tutorials based on above syllabus must be conducted.

Mini Project: Develop a multimedia application

CLASS: B. Sc (Information technology)		Semester – IV	
COURSE: Java and Data Structures			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Core Java:** Features of Java, JVM, Data Types, Variables, and Arrays, Operators, Control Statements, type-casting, Classes, Objects and Methods, Constructor, method overriding, finalize methods.
Unit-II	**Derived concepts:** Inheritance, Packages and Interfaces, Exception Handling, String handling, Multithreaded Programming
Unit-III	**The Java I/O Classes and Interfaces:** File, Directories, Using FilenameFilter, The listFiles( ) Alternative, Creating Directories, The Stream Classes, The Byte Streams, InputStream, OutputStream, FileInputStream, FileOutputStream, ByteArrayInputStream, ByteArrayOutputStream, Filtered Byte Streams, Buffered Byte Streams, SequenceInputStream, PrintStream, RandomAccessFile, The Character Streams, Reader, Writer, FileReader , FileWriter, CharArrayReader, CharArrayWriter, BufferedReader, BufferedWriter, PushbackReader, PrintWriter
Unit-IV	**Data Structures:** Complexity and analysis of algorithms – algorithm, time and space complexity, asymptotic notations, Types of data structures, Arrays - Properties of Arrays, Duplicating an Array, sequential search algorithm, binary search algorithm, Stacks- Stack Operations, indexed Implementation, Linked Implementation, , Applications - recursion, Queue - Queue Operations, indexed Implementation, Applications, Circular Queue – insertion , deletion
Unit-V	**Linked Lists** – representation of linked list, traversing, searching, insertion, deletion and doubly linked list. **Hash table methods** – hashing functions , collision-resolution techniques **Trees-** Binary Trees, traversing binary tree, traversing algorithm using stacks, header nodes, threads, binary search trees (insertion and deletion), AVL trees , B trees
Unit-VI	**Heaps –** insertion and deletion **Sorting –** selection, bubble, merge, tree , radix , insertion **Graphs –** graph theory, sequential representation, adjacency matrix, path matrix, Warshall's algorithm, linked representations, operations, traversing.

Books:

Core Java for Beginners, Sharanam Shah and V Shah, The X Team SPD
Java 2 Complete Reference, 5th Edition , Osborne , Tata-McGrawhill
Data Structures,S Lipschutz, Tata-McGrawhill

Reference:

An introduction to data structures with applications, second edition, Jean-Paul Tremblay, P Sorenson, Tata-McGrawhill
Data Structures with Java, 2nd edition, J Hubbard, Tata-McGrawhill

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Practical List:

1 **Implement the following**

a. Design a java program for type casting different types of variables.
b. Design a Calculator class in java, and implement all the methods required by calculator operations.
c. Design a java class for method overloading and method overriding.

2 Implement the following

a. Design a java program for different types of inheritance.
b. Design a java class for the use of interface.
c. Design a java class performing string operations.

3 Implement the following

a. Design a class in java to add two complex numbers using constructors.
b. Design a java class for performing all the matrix operations i.e addition, multiplication, transpose etc.
c. Design a java class for implementing the packages.

4 Implement the following

a. Design a java class for implementing the concept of threading and multithreading.
b. Design a java class for performing all the file-operations.
c. Design a java class for operating the random access files using

5 Implement the following

a. Design a class for sorting the names or numbers in ascending and descending order.
b. Design a java class for implementing the operations of stack.

6. Implement the following

a. Design a class in java for implementing the operations of queue.(insert, delete, display, exit)
b. Design a class in java for implementing the operations of circular queue.

7. Implement the following

a. Design a class to implement the operations of singly link-list. (insertion , deletion, sorting, display)
b. Design a class to implement the operations of doubly-linked list.

8. Implement the following

a. Implement the concept of hashing technique and also show its collision avoidance.
b. Design a class to create a tree and also implement the binary search tree.

9. Show the implementation

a. Design a class in java for creating the heap and also show heap sort for it.
b. Design a class in java for implementing selection and insertion sort.

10. Show the implementation for the following

a. Design a class in java for bubble and merge sort.
b. Design a class in java for implementing the graph

CLASS: B. Sc (Information technology)		Semester – IV	
COURSE: Quantitative Techniques			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Errors**, **Solutions of Algebraic and Transcendental Equations** using - Bisection Method, the Method of False Position, Newton-Raphson Method. **Interpolation:** Interpolation: - Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation.
Unit-II	**Solution of simultaneous algebraic equations (linear) using iterative methods:** Gauss-Jordan Method, Gauss-Seidel Method. **Numerical Integration:** Trapezoidal Rule, Simpson's 1/3 rd and 3/8 th rules. **Numerical solution of 1st and 2nd order differential equations:** - Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2nd Order Differential Equations.
Unit-III	**Random variables:** Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance. **Moments and moment generating functions:** Relation between Raw moments and Central moments. **Distributions:** Binomial, Poisson, Normal, exponential, uniform distributions for detailed study, Central Limit theorem (statement only) and problems based on this theorem.
Unit-IV	**Fitting of curves:** Least square method, Fitting the straight line and parabolic curve, Correlation, Covariance, Karl Pearson's coefficient and Spearman' s Rank, correlation coefficient, Regression coefficients and lines of regression.
Unit-V	**Sampling distribution:** Test of Hypothesis, Level of Significance, Critical Region, One Tailed and Two Tailed Test, Interval Estimation of Population Parameters, Test of Significance for large Samples and small Samples, Student's 't' Distribution and its properties.
Unit-VI	**Chi-Square Distribution and its properties,** Test of the Goodness of Fit and Independence of Attributes, Contingency Table, Yates Correction **Mathematical Programming:** Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution, Primal Simplex Method.

Books:

Introductory Methods of Numerical Methods, Vol-2, S.S.Shastri, PHI

Fundamentals of Mathematical Statistics, S.C.Gupta, V.K.Kapoor

Reference:

Elements of Applied Mathematics, Volume 1 and 2, P.N.Wartikar and J.N.Wartikar, A. V. Griha, Pune

Engineering Mathematics, Vol-2, S.S.Shastri, PHI

Applied Numerical Methods for Engineers using SCILAB and C, Robert J.Schilling and Sandra L.Harris, " , Thomson Brooks/Cole

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Practical List to be performed in Scilab:

1. **Practical 1: Solution of algebraic and transcendental equations:**
    a. Program to solve algebraic and transcendental equation by bisection method.
    b. Program to solve algebraic and transcendental equation by false position method.
    c. Program to solve algebraic and transcendental equation by Newton Raphson method.
2. **Practical 2: Interpolation**
    a. Program for Newton's forward interpolation.
    b. Program for Newton's backward interpolation.
    c. Program for Lagrange's interpolation.
3. **Practical 3: Solving linear system of equations by iterative methods:**
    a. Program for solving linear system of equations using Gauss Jordan methods.
    b. Program for solving linear system of equations using Gauss Seidel methods.
4. **Practical 4: Numerical Integration**
    a. Program for numerical integration using Trapezoidal rule.
    b. Program for numerical integration using Simpson's $1/3^{rd}$ rule.
    c. Program for numerical integration using Simpson's $3/8^{th}$ rule.
5. **Practical 5: Solution of differential equations:**
    a. Program to solve differential equation using Euler's method
    b. Program to solve differential equation using modified Euler's method.
    c. Program to solve differential equation using Runge-kutta 2^{nd} order and 4^{th} order methods.
6. **Practical 6: Random number generation and distributions**
    a. Program for random number generation using various techniques.
    b. Program for fitting of Binomial Distribution.
    c. Program for fitting of Poisson Distribution.
    d. Program for fitting of Negative Binomial Distribution.
7. **Practical 7: Moments, Correlation and Regression**
    a. Computation of raw and central moments, and measures of skewness and kurtosis.
    b. Computation of correlation coefficient and Fitting of lines of Regression (Raw and Frequency data)
    c. Spearman's rank correlation coefficient.
8. **Practical 8: Fitting of straight lines and second degree curves**
    a. Curve fitting by Principle of least squares. (Fitting of a straight line, Second degree curve)
9. **Practical 9: Sampling:**
    a. Model sampling from Binomial and Poisson Populations.
    b. Model sampling from Uniform, Normal and Exponential Populations.
    c. Large sample tests-(Single mean, difference between means, single proportion, difference between proportions, difference between standard deviations.)
    d. Tests based on students 't-test'(Single mean, difference between means and paired 't')
10. **Practical 10: Chi-square test and LPP**
    a. Test based on Chi-square- Distribution (Test for variance, goodness of Fit,)
    b. Chi-square test of independence of attributes.
    c. Solution of LPP by Simplex method.

CLASS: B. Sc (Information technology)		Semester – IV	
COURSE: Embedded Systems			
Periods per week **1 Period is 50 minutes**	**Lecture**	**5**	
	TW/Tutorial/Practical	**3**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	TW/Tutorial/Practical	**--**	**50**

Unit-I	**Introduction:** Embedded Systems and general purpose computer systems, history , classifications, applications and purpose of embedded systems **Core of embedded systems:** microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little endian processors, Application specific ICs, Programmable logic devices, COTS, sensors and actuators, communication interface, embedded firmware, other system components, PCB and passive components.
Unit-II	**Characteristics and quality attributes of embedded systems:** characteristics, operational and non-operational quality attributes, application specific embedded system – washing machine, domain specific - automotive.
Unit-III	**Programming embedded systems:** structure of embedded program, infinite loop, compiling , linking and locating, downloading and debugging
Unit-IV	**Embedded Hardware:** Memory map, i/o map, interrupt map, processor family, external peripherals, memory – RAM , ROM, types of RAM and ROM, memory testing, CRC ,Flash memory
Unit-V	**Peripherals:** Control and Status Registers, Device Driver, Timer Driver- Watchdog Timers, Embedded Operating System, Real-Time Characteristics, Selection Process
Unit-VI	**Design and Development:** embedded system development environment – IDE, types of file generated on cross compilation, disassembler/ decompiler, simulator , emulator and debugging , embedded product development life-cycle, trends in embedded industry.

Books:

Programming Embedded Systems in C and C++, First Edition January**,** Michael Barr ,O'Reilly

Introduction to embedded systems, Shibu K V, TATAMCGRAW-HILL.

References:

Embedded Systems, Rajkamal, TATAMCGRAW-HILL

Term Work: Should contain at least 6 assignments (one per unit) covering the syllabus.

Tutorial: At least **t**hree tutorials based on above syllabus must be conducted.

Practical List:

1) Configure timer control registers of 8051 and develop a program to generate given time delay.
2) Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED's. Simulate binary counter (8 bit) on LED's
3) Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen. Signify end of message by carriage return.

4) Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope.
5) Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope.
6) Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051.
7) Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction.
8) Generate traffic signal.
9) Temperature controller.
10) Elevator control.