# DEPARTMENT OF BSc. COMPUTER SCIENCE

| | | |
|---|---|---|
| 1 | B.SC | OBJECT ORIENTED PROGRAMMING WITH C++ |
| 2 | B.SC | DATA STRUCTURES AND FILE PROCESSING |
| 3 | B.SC | DATABASE MANAGEMENT SYSTEM |
| 4 | B.SC | DESIGN AND ANALYSIS OF ALGORITHMS |
| 5 | B.SC | WEB PROGRAMMING |
| 6 | B.SC | ELECTIVE<br>A) OPERATING SYSTEM<br>B) DATA MINING<br>C) CRYPTOGRAPHY |
| 7 | B.SC | VISUAL PROGRAMMING |
| 8 | B.SC | ELECTIVE 2<br>A) MULTIMEDIA SYSTEM APPLICATION<br>B) COMPUTER GRAPHICS<br>C) COMPUTER NETWORKS |

Dr A. Sheshachalam
PRINCIPAL
Principal
Vaagdevi Degree & P.G. College
Kishanpura, Hanamkonda

# KAKATIYA UNIVERSITY
# WARANGAL-506009


**B.Sc. with Computer Science Syllabus**

Under the

**CHOICE BASED CREDIT SYSTEM**
**(With effect from 2016-17)**

## DEPARTMENT OF COMPUTER SCIENCE

## University College, KU, Warangal-506009

**DEPARTMENT OF COMPUTER SCIENCE**
**KAKATIYA UNIVERSITY**
**WARANGAL–506 009**

Department of Computer Science, Kakatiya University offers B.Sc (Computer Science) as core subjects at UG level (3 Year course) with six semesters with internal assessment for theory papers under Choice Based Credit System (CBCS) in University constituent and affiliated colleges from 2016-17 academic year onwards.

1. Each of first four Semesters (i.e. I, II III and IV) contains one theory core paper (20 marks for Internal Assessment and 80 marks for Semester End Exam equivalent to 4 credits) as Discipline Specific Course (DSC) and one practical paper (50 marks equivalent to 02 credits), whereas each of last two semesters (i.e. V and VI) contains one theory core paper as DSC (15 marks for Internal Assessment and 60 marks for Semester End Exam equivalent to 3 credits), one theory elective paper as Discipline Specific Elective (DSE) (15 marks for Internal Assessment and 60 marks for Semester End Exam equivalent to 3 credits) and two practical papers (25 marks in each paper equivalent to 01 credit). For total six semesters in Computer Science courses, the total marks are 1000 and credits are 40 for each.
2. Internal Assessment examination will be conducted twice in every Semester. Marks will be awarded from the average of the two Internal Assessment Exams in each Semester.
3. Scheme for CBCS, work-load for each paper, distribution of marks and credits; and scheme of question paper are attached herewith.
4. The syllabi of B.Sc. (Computer Science) theory and practical papers of I, II, III, IV, V and VI semesters are enclosed.
5. The practical examination will be conducted at the end of each semester. A minimum of 40% marks should be obtained by the student to pass the practical examinations in all semesters.
6. All the theory papers and practical papers of B.Sc. (Computer Science) in I, II, III and IV semesters are common to all students. But, elective theory (DSE) papers of B.Sc. (Computer Science) in V and VI Semesters are to be chosen by the student from the available options.
7. Elective (DSE) papers of B.Sc. (Computer Science) enclosed here with. Every student has to choose one elective from the Electives being offered.

# B.Sc. (Computer Science)
## CBCS pattern in Semester System (w. e. from 2016-2017)

| Code | Semester | Course category | Title of the Paper | No. of Credits | HPW | Max. Marks | | | Total Marks |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | I.A | End Exam | Total | |
| ***FIRST YEAR*** | | | | | | | | | |
| **BS106** | **I** | **DSC-1A (Theory)** | **Object Oriented Programming with C++** | **4** | **4** | **20** | **80** | **100** | **125** |
| | | **DSC-1A (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| **BS206** | **II** | **DSC-1B (Theory)** | **Data Structures and File Processing** | **4** | **4** | **20** | **80** | **100** | **125** |
| | | **DSC-1B (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| ***SECOND YEAR*** | | | | | | | | | |
| **BS306** | **III** | **DSC-1C (Theory)** | **Database Management System** | **4** | **4** | **20** | **80** | **100** | **125** |
| | | **DSC-1C (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| | | | | | | | | | |
| **BS406** | **IV** | **DSC-1D (Theory)** | **Design and Analysis of Algorithms** | **4** | **4** | **20** | **80** | **100** | **125** |
| | | **DSC-1D (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| | | | | | | | | | |
| ***THRID YEAR*** | | | | | | | | | |
| **BS505** | **V** | **DSC-1E (Theory)** | **Web programming** | **3** | **3** | **15** | **60** | **75** | **100** |
| | | **DSC-1E (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| **BS508** | | **DSC-1E (Theory)** | **Elective 1 (A/B/C)**<br>**A)** Operating Systems<br>**B)** Data Mining<br>**C)** Cryptography | **3** | **3** | **15** | **60** | **75** | **100** |
| | | **DSC-1E (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| | | | | | | | | | |
| **BS605** | **VI** | **DSC-1F (Theory)** | **Visual Programming** | **3** | **3** | **15** | **60** | **75** | **100** |
| | | **DSC-1F (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| **BS608** | | **DSC-1F (Theory)** | **Elective 2 (A/B/C)**<br>**A)** Multimedia Systems & Applications<br>**B)** Computer Graphics<br>**C)** Computer Networks | **3** | **3** | **15** | **60** | **75** | **100** |
| | | **DSC-1F (Practical)** | | **1** | **2** | **-** | **25** | **25** | |
| | | | | | | | | | |
| | | | **Summary of Credits** | **36** | **-** | **-** | **-** | **-** | **900** |

Note: - Skill Enhancement Courses (SEC1, SEC2, SEC3 and SEC4) will be introduce later

## Core 1: Object Oriented Programming with C++

### Unit I

Algorithm and its characteristics, pseudo code / flow chart, program.

Object Oriented Programming: Introduction, Generation of programming Languages, Programming Paradigms, Features of Object Oriented Programming, Merits and Demerits of Object Oriented Programming Language.

Basics of C++ Programming: Introduction, History, Structure, Writing the First C++ Program, Files used in a C++ Program, Compiling and Executing, Using Comments, Tokens, Characters Set, Keywords, Identifier, Data Types, Variables, Constants, Input and Output, Statements , Operators, Type Conversion and Type Casting,

Decision Control and Looping Statements: Introduction to Decision Control Statements, Conditional Branching Statements, Iterative Statements, Nested Loops, Break Statement, Continue Statement, Goto Statement, Avoiding Usage of Break, Continue, and Goto Statements

### Unit II

Functions: Introduction, Need for Functions, Using Functions, Function Declaration or function, Prototype, Function Definition, Function Call, Return Statement, Passing Parameters to the Function, Default Arguments, Return by Reference, Passing Constants as Arguments, Variables Scope, Storage Classes, Inline Functions, Function Overloading, Recursive Functions, Recursion Versus Iteration, Function with Variable Number of Arguments

Arrays: Introduction, Declaration of Arrays, Accessing Elements of the Array, Storing Values in Arrays, Calculating the Length of Array, Operations that can be Performed on Arrays, One Dimensional Arrays for Inter Function Communication, Two Dimensional Arrays, Multi Dimensional Arrays,

Pointers: Defining Pointers, Declaring Pointer Variables, Pointer Expressions and Pointer Arithmetic, Null Pointers, Generic Pointers, Passing Arguments to Function Using Pointer, Pointer and Arrays, Passing Array to Function, Differences Between Array Name and Pointer, Pointer to function, Arrays of Function Pointers, Memory Allocation in C++, Dynamic Memory Allocation,

### Unit III

Structure, Union, and Enumerated Data Types: Structure Declaration, Typedef Declaration, Initialization the Structures, Accessing the Members of a Structures, Union, Union Inside Structures, Enumerated Data Types.

Classes and Objects : Specifying a Class, Creating Objects, Accessing Object Members, Nested Member Functions, Making a Memory Function Inline, Memory Allocation for Class and Objects, Returning Objects, this Pointer, Constant Parameters and Members, Pointers within a Class, Local Classes, Nested Classes in C++, Empty Classes, Friend Function, Friend Class, Bit-Field in Classes, Pointers and Class Members.

Constructors and Destructors: Constructor, Types of Constructors, Constructor with Default Arguments, Constructor Overloading, Destructors.

**Unit IV**

Operator overloading and Type Conversions : Scope of Operator Overloading, Syntax for Operator Overloading, Operators that can and cannot be Overloaded, Implementing Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Overloading Special Operators, Type Conversions.

Inheritance and Run-Time Polymorphism : Defining Derived Classes, Access Specifiers, Type of Inheritance, Single Inheritance, Constructors and Destructors in Derived Classes, Multi level Inheritance, Constructor in Multi Level Inheritance, Multiple Inheritance, Constructor and Destructor in Multiple Inheritance, Ambiguity in Multiple Inheritance, Hierarchical Inheritance, Constructors and Destructors in Hierarchical Inheritance, Hybrid Inheritance, Multi-path Inheritance, Virtual Base Classes, Object Slicing, Pointer to Derived Class, Run time Polymorphism, Virtual Functions, Pure Virtual Functions, Abstract Base Classes, Concept of Vtables, Virtual Constructor and Destructor.

Templates: Introduction, Use of templates, Function templates, Class templates.

Text Book:


1. Reema Thareja “Object Oriented Programming with C++” Oxford university Press, 2015

Recommended Books

1. E. Balagurusamy “Object Oriented Programming with C++” TMH, 6th edition, 2013.
2. Richard Johnson, *An Introduction to Object-Oriented Application Development*, Thomson Learning, 2006
3. B. Stroupstrup, *The C++ Programming Language*, Addison Wesley, 2004.
4. Spoken Tutorial on “C++” as E-resource for Learning:- http://spoken-tutorial.org

**Practical: Object Oriented Programming with C++**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1) Write a program to test Arithmetic operators.
2) Write a program to Swap two numbers.
3) Write a program to demonstrate Switch statement.
4) Write a program to find roots of a quadratic equation.
5) Write a program to check whether the given number is palindrome or not.
6) Write a program to convert binary number to decimal number.
7) Write a program to print the following format.

   1

   2 3

   4 5 6

   7 8 9 10

8) Write a program to search an element in a given list.
9) Write a program to perform addition of two Matrices.
10) Write a program to perform multiplication of two Matrices.
11) Write a program to find factorial of a given number using recursion.
12) Write a program to demonstrate Pointer arithmetic
13) Write a program to demonstrate Call-By-Value, Call-By-Address,Call-By-Reference.

14) Write a program to demonstrate Structure data type.

15) Write a program to demonstrate Enumerated data type.

16) Write a program to demonstrate inline functions.

17) Write a program to demonstrate Function Overloading.

18) Write a c++ program to demonstrate Class concept.

19) Write a c++ program on Constructor overloading.

20) Write a c++ program on Destructor.

21) Write a c++ program for copy constructor.

22) Write a c++ program to demonstrate Friend function.

23) Write a c++ program for Unary operator overloading (Friend function/Member function).

24) Write a c++ program for Binary operator overloading (Friend function/Member function).

25) Write a c++ program for Member Function overloading within a class

26) Write a c++ program for Single and Multilevel Inheritance.

27) Write a c++ program for Overriding of member functions.

28) Write a c++ program to demonstrate constructor calling mechanism in inheritance.

29) Write a c++ program for Multiple and Hybrid inheritance.

30) Write a c++ program to demonstrate pure virtual function implementation.

## Core 2: Data Structures and File Processing

### Unit I

Basic data Structure: Introduction to Data Structures, Types of Data Structures, and Introduction to Algorithms, Pseudocode, and Relationship among data, data structures, and algorithms, Implementation of data structures, Analysis of Algorithms.

Stacks: Concept of Stacks and Queues, Stacks, Stack Abstract Data Type, Representation of Stacks Using Sequential Organization (Arrays), Multiple Stacks, Applications of Stack, Expression Evaluation and Conversion, Polish notation and expression conversion, Processing of Function Calls, Reversing a String with a Stack, Recursion.

Memory Management: Garbage collection algorithms for equal sized blocks, storage allocation for objects with mixed size, buddy systems

### Unit II

Recursion: Introduction, Recurrence, Use of Stack in Recursion, Variants of Recursion, Recursive Functions, Iteration versus Recursion.

Queues: Concept of Queues, Queue as Abstract Data Type, Realization of Queues Using Arrays, Circular Queue, Multi-queues, Deque, Priority Queue, Applications of Queues,

Linked Lists: Introduction, Linked List, Linked List Abstract Data Type, Linked List Variants, Doubly Linked List, Circular Linked List, Representation of Sparse Matrix Using Linked List, Linked Stack, Linked Queue, Generalized Linked List, More on Linked Lists.

### Unit III

Trees: Introduction, Types of Trees, Binary Tree, Binary Tree Abstract Data Type, Realization of a Binary Tree, Insertion of a Node in Binary Tree, Binary Tree Traversal, Other Tree Operations, Binary Search Tree, Threaded Binary Tree, Applications of Binary Trees.

Searching and Sorting: Searching, Search Techniques, Sorting, Multiway Merge and Polyphase Merge, Comparison of All Sorting Methods, Search Trees: Symbol Table, Optimal Binary Search Tree, AVL Tree (Height-balanced Tree).

**Unit IV**

Hashing: Introduction, Key Terms and Issues, Hash Functions, Collision Resolution Strategies, Hash Table Overflow, Extendible Hashing, Dictionary, Skip List, Comparison of Hashing and Skip Lists.

Heaps: Basic Concepts, Implementation of Heap, Heap as Abstract Data Type, Heap Applications,

Indexing and Multiway Trees: Introduction, Indexing, Types of Search Trees Files: Introduction, External Storage Devices, File Organization, Sequential File Organization, Direct Access File Organization, Indexed Sequential File Organization, Linked Organization.

Text books:


1. Varsha H. Patil " Data structures using C++" Oxford university press, 2012

2. M.T. Goodrich, R. Tamassia and D. Mount, *Data Structures and Algorithms in C++*, John Wiley and Sons, Inc., 2011.


Recommended Books


1. Adam Drozdek "Data structures and algorithm in C++" Second edition, 2001

2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms,* 2nd Ed., Prentice-Hall of India, 2006.

3. Robert L. Kruse and A.J. Ryba, *Data Structures and Program Design in C++*, Prentice Hall, Inc., NJ, 1998.

4. B. Stroupstrup, *The C++ Programming Language*, Addison Wesley, 2004

5. D.E. Knuth, *Fundamental Algorithms* (Vol. I), Addison Wesley, 1997

## Practical: Data Structures and File Processing

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Write C++ programs to implement the following using an array

   a) Stack ADT

   b) Queue ADT

2. Write a C++ program to implement Circular queue using array.

3. Write C++ programs to implement the following using a single linked list.

   a) Stack ADT

   b) Queue ADT

4. Write a C++ program to implement Circular queue using Single linked list.

5. Write a C++ program to implement the double ended queue ADT using double linked list.

6. Write a C++ program to solve tower of hanoi problem recursively

7. Write C++ program to perform the following operations:

   a) Insert an element into a binary search tree.

   b) Delete an element from binary search tree.

   c) Search for a key in a binary search tree.

8. Write C++ programs for the implementation of BFS and DFS.

9. Write a C++ program that uses non-recursive functions to traverse a binary tree.

a)Pre-order

b)In-order

c)Post-order

10. Write a C++ program to find height of a tree.

11 Write a C++ program to find MIN and MAX element of a BST.

12 Write a C++ program to find Inorder Successor of a given node.

13. Write C++ programs to perform the following operations on B-Trees and AVL Trees.

a)Insertion

b)Deletion

14 Write C++ programs for sorting a given list of elements in ascending order using the following sorting methods.

a)Quick sort

b)Merge sort

15. Write a C++ program to find optimal ordering of matrix multiplication.

16. Write a C++ program that uses dynamic programming algorithm to solve the optimal binary search tree problem

17. Write a C++ program to implement Hash Table

18. Write C++ programs to perform the following on Heap

a)Build Heap

b)Insertion

c)Deletion

19. Write C++ programs to perform following operations on Skip List

a)Insertion

b)Deletion

20. Write a C++ program to Heap sort using tree structure.

## Core 3: Database Management System

### Unit I

Introduction: Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Database Users and Administrators.

Introduction to the Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations.

### Unit II

Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues, Extended E-R Features, Alternative Notations for Modeling Data, Other Aspects of Database Design.

Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Decomposition Using Multivalued Dependencies, More Normal Forms, Database-Design Process.

### Unit III

Database-System Architectures: Centralized and Client –Server Architectures, Server System Architectures, Parallel Systems, Distributed Systems, Network Types.

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database.

### Unit IV

Intermediate SQL: Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.

Advanced SQL: Accessing SQL From a Programming Language, Functions and Procedures, Triggers, Recursive Queries.

Text book:

1. A. Silberschatz, H. Korth and S. Sudarshan, *Database System Concepts,* 6th Ed., Tata McGraw Hill, 2011

References:

1. J. Morrison, M. Morrison and R. Conrad, *Guide to Oracle 10g*, Thomson Learning, 2005.
2. Loney and Koch, *Oracle 10g: The Complete Reference*, Tata McGraw Hill, 2006.
3. David Flanagan, Java Script, *The Definitive Guide*, O'Reilly Media, 2006.
4. Marty Hall, Larry Brown, and Yaakov Chaikin, *Core Servlets and Java Server Pages: Core Technologies* (Vol. II), 2nd Ed., Sun Microsystems Press, 2006.
5. S.K. Singh, *Database Systems Concepts, Design and Applications*, Pearson Education 2006.
6. Spoken Tutorial on "MySQL" as E-resource for Learning:- http://spoken-tutorial.org

**Practical: Database Management System**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Create a database having two tables with the specified fields, to computerize a library system of a Delhi University College.

   **LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price)**

   **IssuedBooks (Accession number, Borrower)**

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
   b) Delete the record of book titled "Database System Concepts".
   c) Change the Department of the book titled "Discrete Maths" to "CS".
   d) List all books that belong to "CS" department.
   e) List all books that belong to "CS" department and are written by author "Navathe".
   f) List all computer (Department="CS") that have been issued.
   g) List all books which have a price less than 500 or purchased between "01/01/1999" and "01/01/2004".

2. Create a database having three tables to store the details of students of Computer Department in your college.

   **Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone number)**

   **Paper Details (Paper code, Name of the Paper)**

   **Student's Academic and Attendance details (College roll number, Paper code, Attendance, Marks in home examination).**

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
   b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.
   c) List all students who live in "Delhi" and have marks greater than 60 in paper 1.
   d) Find the total attendance and total marks obtained by each student.
   e) List the name of student who has got the highest marks in paper 2.

3. Create the following tables and answer the queries given below:

**Customer (CustID, email, Name, Phone, ReferrerID)**
**Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)**
**BicycleModel (ModelNo, Manufacturer, Style)**
**Service (StartDate, BicycleID, EndDate)**

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) List all the customers who have the bicycles manufactured by manufacturer "Honda".
c) List the bicycles purchased by the customers who have been referred by customer "C1".
d) List the manufacturer of red colored bicycles.
e) List the models of the bicycles given for service.

4. Create the following tables, enter at least 5 records in each table and answer the queries given below.

**EMPLOYEE ( Person_Name, Street, City )**
**WORKS ( Person_Name, Company_Name, Salary )**
**COMPANY ( Company_Name, City )**
**MANAGES ( Person_Name, Manager_Name )**

a) Identify primary and foreign keys.
b) Alter table employee, add a column "email" of type varchar(20).
c) Find the name of all managers who work for both Samba Bank and NCB Bank.
d) Find the names, street address and cities of residence and salary of all employees who work for "Samba Bank" and earn more than $10,000.
e) Find the names of all employees who live in the same city as the company for which they work.
f) Find the highest salary, lowest salary and average salary paid by each company.
g) Find the sum of salary and number of employees in each company.
h) Find the name of the company that pays highest salary.

5. Create the following tables, enter at least 5 records in each table and answer the queries given below.

**Suppliers (SNo, Sname, Status, SCity)**
**Parts (PNo, Pname, Colour, Weight, City)**
**Project (JNo, Jname, Jcity)**
**Shipment (Sno, Pno, Jno, Qunatity)**

a) Identify primary and foreign keys.
b) Get supplier numbers for suppliers in Paris with status>20.
c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.
d) Get suppliers names for suppliers who do not supply part P2.
e) For each shipment get full shipment details, including total shipment weights.
f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.

g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.
h) Get the names of cities that store more than five red parts.
i) Get full details of parts supplied by a supplier in Delhi.
j) Get part numbers for part supplied by a supplier in Allahabad to a project in Chennai.
k) Get the total number of project supplied by a supplier (say, S1).
l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

## Core 4: Design and Analysis of Algorithms

### Unit I

Fundamentals of the Analysis of Algorithm Efficiency: The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes.

Divide-and-Conquer: maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences.

Dynamic Programming: Rod cutting, Matrix-chain multiplication, Elements of dynamic programming, longest common subsequence, Optimal binary search trees.

Greedy Algorithms: An activity-selection problem, Elements of the greedy strategy, Huffman codes, Matroids and greedy methods, task-scheduling problem as a matroid.

### Unit II

Searching and Sorting Techniques: Review of elementary sorting techniques-selection sort, Bubble sort, insertion sort, more sorting techniques-quick sort, heap sort, merge sort, shell sort, external sorting.

Limitations of Algorithm: Lower-Bound Arguments, Decision Trees, *P*, *NP*, and *NP*-Complete Problems.

Polynomials and the FFT: Representing polynomials, The DFT and FFT, Efficient FFT implementations.

Number-Theoretic Algorithms: Elementary number-theoretic notions, Greatest common divisor(GCD), Modular arithmetic, Addition and Multiplication of two large numbers.

### Unit III

String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm.

NP-Completeness: Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs, NP-complete problems.

Approximation Algorithms: The vertex-cover problem, The traveling-salesman problem, The set-covering problem, Randomization and linear programming, The subset-sum problem.

**Unit IV**

Elementary Graph Algorithms: Representations of graphs, Breadth-first search, Depth-first search, Topological sort, strongly connected components.

Minimum Spanning Trees: Growing a minimum spanning tree, the algorithms of Kruskal and Prim.

Single-Source Shortest Paths: The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties.

Text book:


1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, MIT press, 3rd edition, 2009.
2. Anany Levitin, Introduction to the design and analysis of algorithms, 3rd edition, 2012.


References:


1. J. Kleinberg and E. Tardos, *Algorithms Design*, Pearson Education, 2006.
2. S. Baase, *Computer Algorithms: Introduction to Design and Analysis*, Addison Wesley, 1999.
3. A.V. Levitin, *Introduction to the Design and Analysis of Algorithms*, Pearson Education, 2006.

**Practicals: Design and Analysis of Algorithms**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

Sorting Algorithm:

1. To analyze time complexity of insertion sort
2. To analyze time complexity of Quick sort
3. To analyze time complexity of merge sort

Dynamic Algorithm:

4. To implement largest common subsequence
5. To implement optimal binary search tree
6. To implement martrix chain multiplication

Divide And Conquer:

7. Implement Binary Search Algorithm.
8. Implement Merge Sort Algorithm.
9. Implement Quick Sort Algorithm.
10. To implement strassen's martrix multiplication algorithm

The Greedy Method:

11. Implement activity selection problem
12. Implement fractional Knapsack Problem Algorithm.
13. Implement Job Sequencing with Deadlines Algorithm.

14. Implement Minimum-Cost Spanning Trees: Prim's Algorithm.

15. Implement Single Source Shortest Paths: Dijkstra's Algorithm.

Dynamic Programming:

16. Implement Single-Source Shortest Paths: Bellman-Ford's Algorithm.
17. Implement All-Pairs Shortest Paths: Floyd & Warshall's Algorithm.

Graphs:

18. Implement Dijkstra's algorithm

19. Implement Warshall algorithm

20. Implement Bellman Fords algorthim

21. Implement depth first search algorithm

22. Implement depth first search algorithm

String Matching Algorithm:

23. Implement Naïve string matching algorithm

24. Implement Rabin Karp string matching algorithm

Spanning Trees:

25. Implement prim's algorithm

26. Implement Kruskal's algorithm

## Core 5: Web Programming

**Unit – I**

HTML- Basic HML, The document body, Text, Hyperlinks, Adding More Formatting, Lists, Using Color and Images, Images, Tables, Frames, Forms- Toward Interactivity . Cascading Stylesheets - Introduction, Inline Styles, Embedded Style Sheets, Linking external sheets, Backgrounds, text flow and box model.

**Unit - II**

JavaScript- Introduction, simple programming, Obtaining User Input with prompt Dialogs, Operators (arithmetic, Decision making, assignment, logical, increment and decrement). Control Structures - if… else selection statement, while, do… while repetitions statement, for statement, switch statement, break and continue statements. Functions - program modules in JavaScript, programmer defined functions, function definition, Random-number generator, scope rules, global functions, recursion.

**Unit - III**

JavaScript: Arrays, JavaScript: Objects - Math Object, String Object, Date Object, Boolean & Number Object, document and window Objects. Event Model - on click, on load, on error, onmouseover, onmouseout, on focus, on blur, on submit, on reset, more DHTML events.

Text Book:

1. Internet& World Wide Web- H. M. Deitel, P.J. Deitel, A. B. Goldberg-Third Edition

**Practical: Web Programming**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

Practical exercises based on concepts listed in theory using HTML.

1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create HTML document with Table:

<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="3"></td><td colspan="2" rowspan="2">Some image here</td></tr>
<tr><td colspan="3"></td></tr>
</table>

4. Create Form with Input Type, Select and Text Area in HTML.
5. Create an HTML containing Roll No., student's name and Grades in a tabular form.
6. Create an HTML document (having two frames) which will appear as follows:

<table>
<tr><td colspan="2">About department</td></tr>
<tr><td>Department1</td><td rowspan="3">This frame would show the contents according to the link clicked by the user on the left Frame.</td></tr>
<tr><td>Department1</td></tr>
<tr><td>Department1</td></tr>
</table>

7. Create an HTML document containing horizontal frames as follows:

| Department Names (could be along with Logos) |
| --- |
| Contents according to the Link clicked |

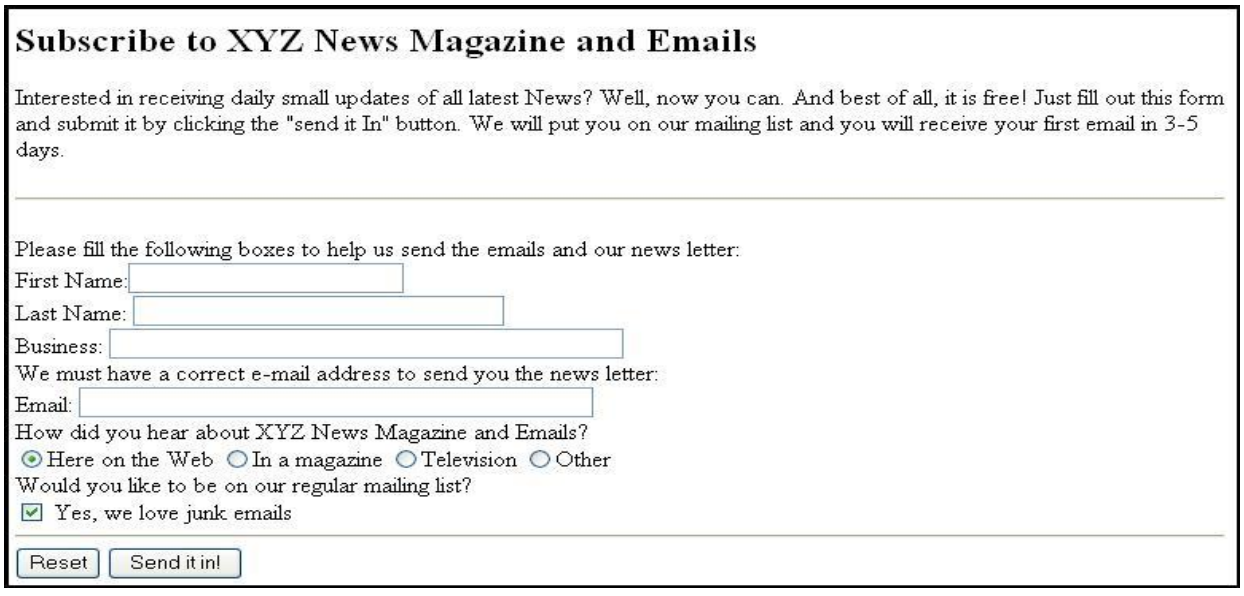
8. Create a website of 6 – 7 pages with different effects as mentioned in above problems.
9. Create HTML documents (having multiple frames) in the following three formats:

| rame1 | |
| --- | --- |
| ame2 | |

| Frame1 | |
| --- | --- |
| Frame 2 | Frame 3 |

10. Create a form using HTML which has the following types of controls:

I. Text Box
II. Option/radio buttons
III. Check boxes
IV. Reset and Submit buttons



11. Create a student Bio-Data, using forms.

12. Create a web page using following style sheets

    i. Inline style sheets. ii. Embedded style sheets. iii. External style sheets

13. Create a web page using “class” style sheets with different “border-width” property values like thick, medium, thin, grove, inset, and outset, red & blue.

**JavaScript:**

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.
7. Write a JavaScript program to accept two values from form and apply any 5 mathematical functions.
8. Display the current date and time in both GMT and local form.
9. Write a JavaScript program on MouseOver, MouseOut, blur events.

## Core 6: Visual Basic Programming

**Unit – I**

Introduction: Visual Basic, Visual Studio Integrated Development Environment, Test-Driving the Visual Basic. Overview of the Visual Studio 2012 IDE, Menu Bar and Toolbar, Navigating the Visual Studio IDE, Using Help, Using Visual App Development to Create a Simple App that Displays Text and an Image.

Introduction to Visual Basic Programming: Introduction, Programmatically Displaying Text in a Label, Addition Program, Building the Addition Program, Memory Concepts, Arithmetic,Decision Making-Equality and Relational Operators.

Introduction to Problem Solving and Control Statements: Introduction, Algorithms, Pseudocode Algorithm, Control Structures, If ... Then Selection Statement, If ... Then ... Else Selection Statement, Nested If ... Then ... Else Selection Statements, Nested Control Statements, Using the Debugger: Locating a Logic Error.

**Unit – II**

Problem Solving and Control Statements: Introduction, For ... Next Repetition Statement, Examples Using the For ... Next Statement, Nested Repetition Statements, Select ... Case Multiple-Selection Statement, Do ... Loop While and Do ... Loop Until Repetition Statements, Using Exit to Terminate Repetition Statements, Using Continue in Repetition Statements, Logical Operators,

Methods: Introduction, Classes and Methods, Subroutines - Methods That Do Not Return a Value, Functions - Methods That Return a Value, Implicit Argument Conversions, Option Strict and Data-Type Conversions, Passing Arguments - Pass-by-Value vs. Pass-by-Reference, Scope of Declarations, Method Overloading, Optional Parameters, Using the Debugger – Debugging Commands

Arrays: Introduction, Arrays, Declaring and Allocating Arrays, Initializing the Values in an Array, Summing the Elements of an Array, Passing an Array to a Method, For Each ... Next Repetition Statement, Rectangular Arrays, Resizing an Array with the ReDim Statement.

**Unit – III**

Windows Forms GUI: A Deeper Look: Introduction, Controls and Components, Creating Event Handlers, Control Properties and Layout, GroupBoxes and Panels, ToolTips, Mouse-Event Handling, Keyboard-Event Handling, Menus, MonthCalendar Control, DateTimePicker Control, LinkLabel Control, ListBox and CheckedListBox Controls, Multiple Document Interface (MDI) Windows, Visual Inheritance, Animation with the Timer Component. Exception Handling: A Deeper Look (Appendix)

Object-Oriented Programming - Classes and Objects: Introduction, Classes, Objects, Methods and Instance Variables, Account Class, Value Types and Reference Types, Class Scope, Object Initializers, Auto-Implemented Properties, Using Me to Access the Current Object, Garbage Collection, Shared Class Members, Const and ReadOnly Fields, Shared Methods and Class Math, Object Browser.

Object-Oriented Programming - Inheritance and Polymorphism : Introduction, Base Classes and Derived Classes, Class Hierarchy, Constructors in Derived Classes, Protected Members, Introduction to Polymorphism - A Polymorphic Video Game, Abstract Classes and Methods,

Databases and LINQ : Introduction, Relational Databases, A Books Database, LINQ to Entities and the ADO.NET Entity Framework, Querying a Database with LINQ, Dynamically Binding Query Results, Retrieving Data from Multiple Tables with LINQ, Creating a Master/Detail View App.

Text Books:

1. Paul Detel, Harvey Detel, Abbey Detel - Visual Basic 2012: How to Program Sixth Edition, 2014.

**Practical: Visual Basic Programming**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Print a table of numbers from 5 to 15 and their squares and Cubes.
2. Print the largest of three numbers.
3. Find the factional of a number n.
4. Enter a list of positive numbers terminated by zero. Find the sum and average of these numbers.
5. A person deposits Rs. 1000 in a fixed account yielding 5% interest. Complete the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.
7. Read n numbers. Count the number of negative numbers, positive numbers and zeroes in the list.use arrays.
8. Read a single dimension array. Find the sum and average of these numbers.
9. Read a two dimension array. Find the sum of two 2D Array.
10. Create a database Employee and Make a form to allow data entry to **Employee Form** with the following command buttons:

<u>Employee Form</u>

Employee Name: ☐

Employee Id: ☐

Date of Joining: ☐

☐

Designation:

Department:

Address:

Basic Pay:

PREV | NEXT | FIRST | LAST | ADD | SAVE | DELETE | CANCEL

## Elective1.A: Operating Systems

### Unit I

OPERATING SYSTEM: Introduction, The Operating System As A Resource Manager, History Of Operating Systems, The Operating System Zoo(Classifications), Operating System Concepts, System Calls, Operating System Structure(Architecture).

PROCESS- Creation , Hierarchies, States, THREADS- Usage, The Classical Thread Model, POSIX Threads, Pop-Up Threads.

SCHEDULING: Introduction, Scheduling in Batch Systems, Scheduling in Interactive Systems, Scheduling in Real-Time Systems, Policy versus Mechanism, Thread Scheduling.

### Unit II

MEMORY MANAGEMENT: No Memory Abstraction, Memory Abstraction-Address Spaces, Virtual Memory, Page Replacement Algorithms, Design Issues for Paging Systems, Segmentation.

FILE SYSTEMS: Files, Directories, File System Implementation, File-System Management and Optimization, Example File Systems.

### Unit III

INPUT/OUTPUT: Principles Of I/O Hardware, Principles Of I/O Software, I/O Software Layers, Disks, Clocks, User Interfaces: Keyboard, Mouse, Monitor, Thin Clients, Power Management.
DEADLOCKS: Resources, Introduction to Deadlocks, Deadlock Detection and Recovery, Deadlock Avoidance, Deadlock Prevention.

SECURITY: The Security Environment, Operating Systems Security, Formal Models of Secure Systems, Authentication, Insider Attacks, Malware, Defenses.

### Text book:

1. A.S. Tanenbaum, and Herbert Bos, "Modern *Operating Systems",* 4th Ed., Prentice-Hall of India, 2015.

### Recommended Books

1. William Stallings, *Operating Systems: Internals and Design Principles*, 5th Ed., Prentice-Hall of India, 2006.
2. Gary Nutt, *Operating Systems: A Modern Approach,* 3rd Ed., Addison Wesley, 2004.
3. D.M. Dhamdhere, *Operating Systems: A Concept Based Approach,* 2nd Ed., Tata McGraw-Hill, 2007.

**Practical: Operating System**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Simulate the following CPU Scheduling algorithms a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all file allocation strategies. a) Sequential b) Indexed c) Linked
3. Simulate MVT and MFT
4. Simulate all File organization techniques. a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance 6. Simulate Bankers Algorithm Dead Lock Prevention.
6. Simulate all Page replacement algorithms. a) FIFO b) LRU c) LFU d) Etc.…
7. Simulate Paging Techniques of memory management.

## Elective1.B: Data mining

**Unit I**

Data Mining and Knowledge Discovery Process: data mining, Data Mining Differ from Other Approaches - The Knowledge Discovery Process-Introduction, Knowledge Discovery Process, Knowledge Discovery Process Models.

Data Understanding: data, Concepts of Learning, Classification, and Regression

**Unit II**

Data Mining: Methods for Constructing Data Models: Unsupervised Learning: Clustering-From Data to information Granules or Clusters, Categories of Clustering Algorithms, Hierarchical Clustering, Objective Function-Based Clustering, Cluster Validity, random Sampling and Clustering as a Mechanism of Dealing with large datasets.

Association Rules – Introduction, Association Rules and transactional Data , Mining Single Dimensional , Single-Level Boolean Association Rules, Mining Other Types of Association Rules.

**Unit III**

Supervised Learning: Bayesian Methods, Regression- Decision Trees, Rule and Hybrids Algorithms.

Text Mining: Introduction, Information Retrieval Systems, Improving Information Retrieval Systems.

Data Security, Privacy and Data Mining: Privacy in Data Mining, Privacy Versus Levels of Information Granularity, Distributed Data Mining, Collaborative Clustering.

Text Books:

1. Data mining A knowledge discovery approach , Pedrycz, Kurgan, Springer , 2007

References:

1. Data mining Concepts and Techniques , Micheline Kamber, third edition, MK Elsevier publications
2. Principles of data mining , David hand Heikki Mannila , PHI publications-2004

**Practical: Data mining**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Demonstration of preprocessing on dataset student.arff

2. Demonstration of preprocessing on dataset labor.arff

3. Demonstration of Association rule process on dataset contactlenses.arff using apriori algorithm

4. Demonstration of Association rule process on dataset test.arff using apriori algorithm

5. Demonstration of classification rule process on dataset student.arff using j48 algorithm

6. Demonstration of classification rule process on dataset employee.arff using j48 algorithm

7. Demonstration of classification rule process on dataset employee.arff using id3 algorithm

8. Demonstration of classification rule process on dataset employee.arff using naïve bayes algorithm

9. Demonstration of clustering rule process on dataset iris.arff using simple k-means

10. Demonstration of clustering rule process on dataset student.arff using simple kmeans

## Elective1.C: Cryptography

### Unit I

Introduction: Security Trends, Security Attacks, Security Services, Security Mechanisms, Model for Network Security,

Symmetric Ciphers: Classical Encryption Techniques, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography.

Data Encryption Standard: Block Cipher Principles, The Data Encryption Standard, The Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles. Advanced Encryption Standard: Evaluation Criteria For AES, The AES Cipher.

### Unit II

Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm,

Public-Key Cryptosystems: Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

### Unit III

Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs.

Digital Signatures and Authentication Protocols: Digital Signatures, Kerberos, X.509 Authentication Service, Public-Key Infrastructure.

Text book:

1. W. Stallings, *Cryptography and Network Security Principles and Practices,* 4th Ed., Prentice-Hall of India, 2006.

Book Recommended

1. C. Pfleeger and S.L. Pfleeger, *Security in Computing,* 3rd Ed., Prentice-Hall of India, 2007.

2. M.Y. Rhee, Network Security, John Wiley and Sons, NY, 2002.

**Practical: Cryptography**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Study various cryptography techniques.
2. Implement the Pure Transposition Cipher
3. Implement Additive cipher
4. Implement DES Encryption and Decryption
5. implement double transposition cipher
6. Implement RSA Encryption Algorithm
7. Implement RSA algorithm to achieve confidentiality
8. Implement RSA algorithm to create Digital Signatures
9. Implementation of Hash Functions
10. Implement Diffie Hellman Key Exchange.

## Elective2.A: Multimedia Systems and Applications

### Unit I

Multimedia: Introduction, Definitions, Where to Use Multimedia- Multimedia in Business, Schools, Home, Public Places, Virtual Reality; Delivering Multimedia.

Text: Meaning, Fonts and Faces, Using Text in Multimedia, Computers and Text, Font Editing and Design Tools, Hypermedia and Hypertext.

Images: Before You Start to Create, Making Still Images, Color.

Sound: The Power of Sound, Digital Audio, MIDI Audio, MIDI vs. Digital Audio, Multimedia System Sounds, Audio File Formats. Adding Sound to Your Multimedia Project.

Animation: The Power of Motion, Principles of Animation, Animation by Computer, Making Animations.

### Unit II

Video: Using Video, How Video Works and Is Displayed, Digital Video Containers, Obtaining Video Clips, Shooting and Editing Video.

Making Multimedia: The Stages of a Multimedia Project, the Intangibles, Hardware, Software, Authoring Systems

Planning and Costing: The Process of Making Multimedia, Scheduling, Estimating; Designing and Producing.

### Unit III

The Internet and Multimedia: Internet History, Internetworking, Multimedia on the Web.

Designing for the World Wide Web: Developing for the Web, Text for the Web, Images for the Web, Sound for the Web, Animation for the Web, Video for the Web.

Delivering: Testing, Preparing for Delivery, Delivering on CD-ROM, DVD and World Wide Web, Wrapping.

Text book:

1. Tay Vaughan, “Multimedia: Making it work”, TMH, Eighth edition.

Reference books:


1. Ralf Steinmetz and Klara Naharstedt, “Multimedia: Computing, Communications Applications”, Pearson.
2. Keyes, “Multimedia Handbook”, TMH.
3. K. Andleigh and K. Thakkar, “Multimedia System Design”, PHI.
4. Spoken Tutorial on “GIMP”as E-resource for Learning:- http://spoken-tutorial.org
5. Spoken Tutorial on “Blender” as E-resource for Learning:- http://spoken-tutorial.org

**Practical: Multimedia Systems and Applications**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

Practical exercises based on concepts listed in theory using Presentation tools in office automation tool/ GIMP/Blender / Audacity/ Animation Tools/ Image Editors/ Video Editors.

Implement the followings using Blender -

1. Create an animation using the tools panel and the properties panel to draw the following – Line, pe , oval, circle, rectangle , square, pencil , brush , lasso tool

2. Create an animation using text tool to set the font , size , color etc.

3. Create an animation using **Free transform tool** that should use followings-

   Move Objects

   Skew Objects

   Stretch Objects

   Rotate Objects

   Stretch Objects while maintaining proportion

   Rotate Objects after relocating the center dot

4. Create an animation using layers having following features-

Insert layer, Delete layer, guide layer, Mask layer.

5. Modify the document (changing background color etc.)Using the following tools

Eraser tool

Hand tool

Ink bottle tool

Zoom tool

Paint Bucket tool

Eyedropper tool

6. Create an animation for bus car race in which both starts from the same point and car wins the race.
7. Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).
8. Create an animation having five images having fade-in fade-out effect.
9. Create an scene to show the sunrise (using multiple layers and motion tweening)
10. Create an animation to show the ripple effect.
11. Create an animation (using Shape tweening and shape hints) for transforming one shape into another.
12. Create an animation for bouncing ball (you may use motion guide layer).

## Elective2.B: Computer Graphics

### Unit I

Computer Graphics: Graphs and Charts, Computer-Aided Design, Virtual-Reality Environments, Data Visualizations, Education and Training, Computer Art, Entertainment, Image Processing, Graphical User Interfaces.

Computer Graphics Hardware: Video Display Devices, Raster-Scan System, GraphicsWorkstations and Viewing Systems, Input Devices, Hard-Copy Devices, Graphics Networks, Graphics on the Internet.

Computer Graphics Software: Coordinate Representations, Graphics Functions, Software Standards, Other Graphics Packages, Introduction to OpenGL.

### Unit II

Graphics Output Primitives: Coordinate Reference Frames, Specifying A Two-Dimensional World-Coordinate Reference Frame in OpenGL, OpenGL Point Functions, OpenGL Line Functions, OpenGL Curve Functions, Fill-Area Primitives, Polygon Fill Areas, OpenGL Polygon Fill-Area Functions, OpenGL Vertex Arrays, Pixel-Array Primitives, OpenGL Pixel-Array Functions, Character Primitives, OpenGL Character Functions, Picture Partitioning, OpenGL Display Lists, OpenGL Display-Window Reshape Function.

Attributes of Graphics Primitives: OpenGL State Variables, Color and Grayscale, OpenGL Color Functions, Point Attributes, OpenGL Point-Attribute Functions, Line Attributes, OpenGL Line-Attribute Functions, Curve Attributes, Fill-Area Attributes, OpenGL Fill-Area Attribute Functions, Character Attributes, OpenGL Character-Attribute Functions, openGL Antialiasing Functions, OpenGL Query Functions, OpenGL Attribute Groups.

Algorithms for Graphics Primitives and Attributes: Line-Drawing Algorithms, Circle-Generating Algorithms, Ellipse-Generating Algorithms.

### Unit III

Two-Dimensional Geometric Transformations: Basic Two-Dimensional Geometric Transformations, Matrix Representations, Inverse Transformations, Two-Dimensional Composite Transformations, Raster Methods for Geometric Transformations, OpenGL Raster Transformations, Transformations between

Two-Dimensional Coordinate Systems, OpenGL Functions for Two-Dimensional Geometric Transformations.

Two-Dimensional Viewing: The Two-Dimensional Viewing Pipeline, The ClippingWindow, Normalization and Viewport Transformations, OpenGL Two-Dimensional Viewing Functions, Clipping Algorithms, Two-Dimensional Point Clipping, Two-Dimensional Line Clipping, Polygon Fill-Area Clipping, Curve Clipping, Text Clipping.

Text Book:

1. Donald D. Hearn, M. Pauline Baker, Warren Carithers “Computer Graphics with Open GL” 4th Edition, 2011.

References:

1. J.D. Foley, A van Dam, S.K. Feiner and J.F. Hughes, *Computer Graphics: Principals and Practices,* 2nd Ed., Addison-Wesley, MA, 1990.
2. D.F. Rogers, *Procedural Elements in Computer Graphics,* 2nd Ed., McGraw Hill Book Company, 2001.
3. D.F. Rogers and A.J. Admas, *Mathematical Elements in Computer Graphics*, 2nd Ed., McGraw Hill Book Company, 1990.

**Practical: Computer graphics**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Program to recursively subdivide a tetrahedron to from 3D Sierpinski gasket. The number of recursive steps is to be specified by the user.
2. Program to implement Liang-Barsky line clipping algorithm.
3. Program to draw a color cube and spin it using OpenGL transformation matrices.
4. Program to create a house like figure and rotate it about a given fixed point using OpenGL functions.
5. Program to implement the Cohen-Sutherland line-clipping algorithm. Make provision to specify the input line, window for clipping and view port for displaying the clipped image.
6. Program to create a cylinder and a parallel piped by extruding a circle and quadrilateral respectively. Allow the user to specify the circle and quadrilateral.
7. Program using OpenGL functions, to draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the surfaces of the solid object used in the scene.
8. Program to draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing. Use OpenGL functions.
9. Program to fill any given polygon using scan-line area filling algorithm. (Use appropriate data structures.)
10. Program to display a set of values {fij} as a rectangular mesh.

## Elective2.C: Computer Networks

**Unit I**

Introduction: data communication- components, Networks, Protocols and Standards, Line configuration, topology, transmission mode, categories of networks.

OSI and TCP/IP Models: Layers and their functions, comparison of models.

Multiplexing: Many To One/One To Many, Frequency-Division Multiplexing (FDM), Wave-Division Multiplexing (WDM), Time-Division Multiplexing (TDM),

**Unit II**

Error Detection and Correction: Types of Errors, Detection, Vertical Redundancy Check (VRC), Longitudinal Redundancy Check (LRC), Cyclic Redundancy Check (CRC), Checksum, Error Correction.

Data Link Control: Line Discipline, Flow Control, Error Control, Asynchronous Protocols, Synchronous Protocols, Character-Oriented Protocols, Bit-Oriented Protocols, Link Access Procedures

Switching: Circuit Switching, Packet Switching, Message Switching.

**Unit III**

Transport Layer: Duties of the Transport Layer, Connection, the OSI Transport Protocol.

Upper OSI Layers: Session Layer, Presentation Layer, Application Layer.

TCP/IP Protocol Suite: Overview of TCP/IP, Network Layer, Addressing, Subnetting, Other Protocols In The Network Layer, Transport Layer

Text Book:


1. Behrouz A. Forouzan, *Data Communication and Networking*, 2nd Ed., Tata McGraw Hill.


Books Recommended


1. A.S. Tenanbaum, *Computer Networks*, 4th Ed., Pearson Education Asia, 2003.
2. D. E. Comer, *Internetworking with TCP/IP*, Pearson Education Asia, 2001.
3. William Stallings, *Data and Computer Communications*, 7th Ed., Pearson education Asia, 2002.

**Practical: Computer Networks**

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

Simulation Exercises: The following experiments shall be conducted using either NS3/OPNET or any other simulators.

1. Simulate a three nodes point-to-point network with duplex links between them. Set the queue size vary the bandwidth and find the number of packets dropped.
2. Simulate a four node point-to-point network, and connect the links as follows: n0- n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets by TCP/UDP.
3. Simulate the different types of Internet traffic such as FTP a TELNET over a network and analyze the throughput.
4. Simulate the transmission of ping messaged over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
5. Simulate an Ethernet LAN using N-nodes (6-10), change error rate and data rate and compare the throughput.
6. Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and determine collision across different nodes.
7. Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and plot congestion window for different source/destination.
8. Simulate simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.