SEMESTER 5

Course code	**CC-ID13**			
Category	Core Course			
Course title	**Computer Graphics**			
Course ID	240/BCA/CC501			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have five parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no. 2 to 9 will be 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

Course Outcomes:

CO1: Develop a comprehensive understanding of computer graphics systems, including display devices, input methods, and their application across various domains.

CO2: Master 2D transformations (translation, scaling, rotation, reflection, shear) and viewing techniques, enabling precise manipulation and visualization of graphical objects.

CO3: Gain proficiency in representing 3D objects using polygonal and spline surfaces, and implementing basic illumination models for realistic rendering.

CO4: Acquire advanced skills in 3D transformations (translation, rotation, scaling, reflection, shear) and viewing techniques, facilitating complex 3D scene creation and visualization.

UNIT-I

Graphics Primitives: Introduction to computer graphics, Basics of Graphics systems, Application areas of Computer Graphics, overview of graphics systems, video-display devices, and raster-scan systems, random scan systems, graphics monitors and workstations and input devices.

Output Primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary fill and flood- fill algorithms.

UNIT-II

2-D Geometrical Transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D Viewing: The viewing pipeline, viewing coordinate reference frame, window to view- port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT-III

3-D Object Representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon-rendering methods.

UNIT-IV

3-D Geometric Transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D Viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

TEXT AND REFERENCE BOOKS:

1. Donald Hearn and M. Pauline Baker : Computer Graphics, PHI Publications.
2. Plastock : Theory & Problem of Computer Gaphics, Schaum Series.
3. Foley & Van Dam : Fundamentals of Interactive Computer Graphics, Addison-Wesley.
4. Newman : Principles of Interactive Computer Graphics, McGraw Hill.
5. Tosijasu, L.K. : Computer Graphics, Springer-Verleg.

Computer Graphics Lab

List of Experiment

1. Implement algorithms to draw points and lines using DDA (Digital Differential Analyzer) and Bresenham's line drawing algorithm.
2. Practice drawing circles and ellipses using midpoint circle and ellipse algorithms.
3. Implement scan-line polygon fill algorithm to fill simple polygons.

4. Implement boundary fill and flood-fill algorithms to fill regions with different colors.
5. Implement translation, scaling, rotation, reflection, and shear transformations using matrix representations and homogeneous coordinates.
6. Perform composite transformations (e.g., translate then rotate).
7. Implement the Cohen-Sutherland line clipping algorithm to clip lines against a rectangular window.
8. Implement the Sutherland-Hodgman polygon clipping algorithm to clip polygons against a convex clipping window.
9. Implement algorithms for drawing Bezier curves and B-Spline curves.
10. Implement 3-D transformations including translation, rotation, scaling, reflection, and shear.
11. Combine transformations to perform composite transformations on 3-D objects.
12. Implement perspective and orthographic projections for 3-D objects.

Course code	**CC-ID14**			
Category	Core Course			
Course title	**Python Programming**			
Course ID	240/BCA/CC502			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have five parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no. 2 to 9 will be 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

Course Outcomes:

CO1: Develop proficiency in fundamental programming concepts including variables, data types, control statements (if-else, loops), and file manipulation (reading/writing text and CSV files).

CO2: Master data structures such as lists, tuples, and dictionaries, and understand function design principles including arguments, return values, and managing program complexity.

CO3: Acquire skills in OOP with classes, inheritance, polymorphism, and exception handling, enabling effective data modeling, abstraction, and code reusability.

CO4: Gain practical experience in creating graphical user interfaces (GUIs) using tkinter, including designing simple interfaces with widgets like buttons, labels, and entry fields, and managing layouts and event-driven programming paradigms.

UNIT-I

The concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; comments in the program; understanding error messages; Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation; Strings and text files; manipulating files and directories, text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab separated); String manipulations: subscript operator, indexing, slicing a string.

UNIT-II

Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments.

UNIT-III

Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects; Encapsulation, Information hiding Method, Signature, Classes and Instances, Review of Abstraction, inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; exception handling, try block

UNIT-IV

Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

TEXT AND REFERENCE BOOKS:

1. Phillips, Dusty. Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8. Packt Publishing Ltd, 2018.

2. Steven F. Lott, Mastering Object-Oriented Python - Second Edition, published by Packt.

3. Python Object Oriented Programming Cookbook, published by Packt.

4. Mark Lutz, Programming Python: Powerful Object-Oriented Programming.

5. Irv Kalb, Object-Oriented Python: Master OOP by Building Games and GUIs Kindle Edition

Python Programming Lab

List of Practical:

1. Write some Basic building blocks of a Python program using:
   a) Variables, conditional statements
   b) Loops
   c) Libraries
   d) Functions

2. Implement various operations in python using :

a) Lists

b) Tuples

c) Dictionaries

d) Strings

3. Implement Object Oriented programming concept in python :
   a) classes, objects
   b) Inheritance
   c) polymorphism, abstract classes
   d) exception handling

4. Create graphical user interfaces for interactive programs in Python

Course code	**CC-ID15**			
Category	Core Course			
Course title	**Analysis and design of algorithm**			
Course ID	240/BCA/CC503			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have five parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no. 2 to 9 will be 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OUTCOMES:

CO1: Apply the best data structure for designing an algorithm to solve a given problem.

CO2: Evaluate different algorithms with respect to time and space complexity.

CO3: Create algorithms to solve various computational problems.

CO4: Understand different complexity classes.

CO5: Explain the hardness of real world problems with respect to algorithmic efficiency and learning to cope with it.

UNIT I

BASIC CONCEPTS OF ALGORITHMS: Basic Concepts of Algorithms: Notion of Algorithm, Fundamentals of Algorithmic Solving, Important problem types, Fundamentals of the Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of nonrecursive algorithms. Mathematical analysis of recursive algorithm: recurrence relations, solution of recurrence relations using substitution method

UNIT II

SORTING ALGORITHMS: Brute Force, Divide and Conquer Strategy: Selection sort, Bubble sort, Sequential searching (Linear Search), Brute force string matching, General method, Merge sort, Quick Sort, Binary Search, Strassen's matrix multiplication

UNIT III

GREEDY AND DYNAMIC PROGRAMMING: Greedy Approach and Dynamic Programming: Fractional Knapsack problem, Minimum cost spanning tree: Prim's and Kruskal's algorithm, Single source shortest path problem, Principle of optimality, Multi-stage graph problem, all pair shortest path problem, 0/1 Knapsack problem, Traveling salesperson problem

UNIT IV

BACKTRACKING & BRANCH AND BOUND: Backtracking and Branch and Bound: General method backtracking, N-Queen problem, 0/1 Knapsack problem, General method of branch & bound, 0/1 Knapsack problem, Traveling sales person problem, P, NP and NP Complete problems

TEXT AND REFERENCE BOOKS:

1. Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education 2014.
2. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Book Publishing 2018.
3. Fundamentals of algorithms, Horowitz E, Sahini S, Rajasekaran S., University Press 2008.
4. Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, PHI. 2012
5. An introduction to analysis of algorithms, R. Sedgewick, 1st edition, Pearson Education 1996.
6. Data Structures and Program Design in C By Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education. 2007.

Analysis and Design of Algorithms Lab

List of Practical

1. Write a program to implement different sorting techniques.
    - Bubble Sort
    - Insertion Sort
    - Selection Sort
    - Quick Sort
    - Merge Sort
2. Write a program to find minimum cost spanning tress.

3. Write a program to implement travelling sales person problem.
4. Write a program to find Longest Path in a Directed Acyclic Graph.
5. Write a program for Shortest path with exactly k edges in a directed and weighted graph.
6. Write a program find maximum number of edge disjoint paths between two vertices
7. Implement 0/1 Knapsack problem using Dynamic Programming.
8. Perform various tree traversal algorithms for a given tree.
9. Implement N-Queens Problem

Course code	VOC-1			
Category	Vocational Courses (VOC)			
Course title	**Digital Marketing Using CANVA**			
Course ID	240/BCA/VO501			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs.			

Note: The examiner will set nine questions in total. Question one will have five parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no. 2 to 9 will be 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OUTCOMES:

CO1: Understanding of the evolution of digital marketing from traditional to modern eras, including the role of the internet and current trends impacting business and society.

CO2: Proficiency in employing digital marketing strategies within the P.O.E.M. framework and various digital marketing models.

CO3: Competence in utilizing internet marketing techniques and the digital marketing mix to enhance integrated marketing communications (IMC) and optimize digital channels.

CO4: Proficiency in implementing social media marketing strategies across platforms like Facebook, Twitter, Instagram, Snapchat, and mobile marketing, including campaign development and analytics.

Unit 1

Introduction to Digital Marketing, Evolution of Digital Marketing from traditional to modern era, Role of Internet; Current trends, Info-graphics, implications for business & society; Emergence of digital marketing as a tool; Drivers of the new marketing environment; Digital marketing strategy; P.O.E.M. framework, Digital landscape, Digital marketing plan, Digital marketing models.

Unit 2

Graphic Design for Marketing using Canva: Introduction to Canva interface, Creating designs using templates, Customizing fonts, colors and images, Designing social media posts, Posters and banners, Working with Canva elements and grids, Downloading and sharing content

Unit 3

Social Media Marketing – Role of Influencer Marketing, Tools & Plan– Introduction to social media platforms, penetration & characteristics; Building a successful social media marketing strategy. **Facebook Marketing**: **-** Business through Facebook Marketing, Creating Advertising Campaigns, Adverts, Facebook Marketing Tools, **Twitter Marketing**: **-** Introduction to Twitter Marketing, how twitter Marketing is different than other forms of digital marketing, framing content strategy, **Twitter Advertising Campaigns Instagram and Snapchat: -** Digital Marketing Strategies through Instagram and Snapchat

Unit 4

Introduction to SEO, SEM, Web Analytics, Mobile Marketing, Trends in Digital Advertising– - Introduction and need for SEO

Introduction to SEM Web Analytics: - Google Analytics & Google AdWords; data collection for web analytics, multichannel attribution, Universal analytics, Tracking code Trends in digital advertising.

TEXT AND REFERENCE BOOKS:

1. "Digital Marketing: Strategy, Implementation and Practice" by Dave Chaffey and Fiona Ellis-Chadwick
2. "Social Media Marketing: A Strategic Approach" by Melissa Barker, Donald I. Barker, Nicholas F. Bormann, and Krista E. Neher
3. "Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity" by Avinash Kaushik
4. "SEO 2024: Learn Search Engine Optimization with Smart Internet Marketing Strategies" by Adam Clarke
5. Canva Design School – https://designschool.canva.com

List of Practical:

Students are advised to do laboratory/practical practice not limited to but including the following types of problems:

1. Creating a Canva account and exploring templates
2. Designing a basic social media post
3. Creating a brand logo and color palette

4. Designing a marketing poster for a product/service
5. Creating Instagram and Facebook ad creatives
6. Designing a YouTube thumbnail
7. Preparing an email header for a promotional campaign
8. Creating a business card or brochure
9. Designing carousel posts and story graphics
10. Creating a content calendar using Canva templates

Course code	VOC-1			
Category	Vocational Course (VOC)			
Course title	Software Testing			
Course ID	240/BCA/VO502			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	03 Hours			

Note: The examiner will set nine questions in total. Question one will have Five parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no. 2 to 9 will be 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OUTCOMES:

On the completion of the course students will:

1. To understand the basic terminologies and types of testing
2. Understand different testing methods
3. Understand the testing process
4. Manage the tests, plan testing process and create reports
5. Testing the software/projects using various techniques

UNIT I

Introduction: Definition of Software Testing and its Role, **Terms: -** Failure, Error, Fault, Defect, Bug, Goals of Testing, Principles of Testing, Software Testing Life Cycle, Verification and Validation: - V-testing Life cycle

UNIT II

Types of Testing: Black Box Testing: Overview: What is & When? Techniques: Boundary Value Analysis, Equivalence class testing, Decision Table White Box Testing: What is white box Testing, Need of white box Testing, Classification, Structural: Coverage, Path testing

.

UNIT III

Levels of Testing Unit Testing : Overview, Integration Testing : Overview, Techniques: Graph based & Path based, Functional Testing, System Testing : Overview, Categories: Reliability Security Performance Recovery, Acceptance Testing : Overview, Types of Acceptance Testing.

UNIT IV

Test Planning: Preparing a Test plan, Scope management, Decide Test Approach, Setting Up Criteria, for testing, Identifying responsibilities, Staffing, training needs, Resource requirements, Test deliverables, Testing Tasks.

TEXT AND REFERENCE BOOKS:

1. Software Testing: Principles and Practice by Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Publication
2. Software Testing: Principles and Practice by Naresh Chauhan, Oxford
3. Software Testing: Easy Learning Approach by Shubha Agarwal Kundlas.

List of Practical:

Students are advised to do laboratory/practical practice not limited to but including the following types of problems:

1. Prepare a small project and submit SRS, design, coding and test plan.
2. Study of any one of the testing tools. (e.g win runner, test direct, etc)
3. MANUAL TESTING for the project
   a. Whitebox Testing
   b. Blackbox Testing
4. Functional Testing
   a. Boundary value Testing
   b. Equivalence class testing
5. Structural Testing
   a. Path testing
   b. Data-flow testing