MANGALORE UNIVERSITY

State Education Policy – 2024
[SEP-2024]

CURRICULUM STRUCTURE

FOR

BCA

BACHELOR OF COMPUTER APPLICATIONS

MANGALORE UNIVERSITY

Suggested Programme Structure for the Under Graduate Programmes

[BCA, BCA (A.I & M.L), BCA (D.A)]

Semester	Course 1	Course 2	Course 3	Elective / Optional	Course	Language	Compulsory	Total Credit	Total Working hour
I	5 (3T+2P)	5 (3T+2P)	5 T			3+3	2	23	4+4+4+4+5+4+4+2=31
II	5 (3T+2P)	5 (3T+2P)	5T			3+3	2	23	4+4+4+4+5+4+4+2=31
III	5 (3T+2P)	5 (3T+2P)	5T	2		3+3		23	4+4+4+4+5+4+4+2=31
IV	5 (3T+2P)	5 (3T+2P)	5T	2		3+3	2	25	4+4+4+4+5+2+4+4+2=33
V	8[(2x3T)+2P]	8[(2x3T)+2P]	8[(2x3T)+2P]				2	26	3+3+4+3+3+4+3+3+4+2=32
VI	3T	3T	3T		3T	Project work 12		24	3+3+3+3+24=36
Total Credits for the Programme								**144**	

Note:

- **Course1 and Course2: I to IV Semester: Theory 3 credits=4 contact hours & Practical 2 credits=4 contact hours**
- **Course3: I to IV Semester: Theory 5 credit=5 contact hours**
- **Course1, Course2 and Course3: V and VI Semester: Theory 3 credits=3 contact hours & Practical 2 credits=4 contact hours**
- **Elective/Optional: 2 credits=2 contact hours**
- **Languages: 3 credits=4 contact hours**
- **Compulsory: 2 credits=3 contact hours**

Semester IV								
Sl. No	**Course Code**	**Title of the Course**	**Category of Courses**	**Teaching Hours per Week**	**SEE**	**IA**	**Total Marks**	**Credits**
1		**Language-I**	**Lang**	**4**	**80**	**20**	**100**	**3**
2		**Language-II**	**Lang**	**4**	**80**	**20**	**100**	**3**
3	**BCACACS401**	Python Programming	**Core**	**4**	**80**	**20**	**100**	**3**
4	**BCACACS402**	Advanced JAVA and J2EE	**Core**	**4**	**80**	**20**	**100**	**3**
5	**BCACACS403**	Operating System Concepts	**Core**	**5**	**80**	**20**	**100**	**5**
6	**BCACAPS404**	Python Programming-Lab	**Practical**	**4**	**40**	**10**	**50**	**2**
7	**BCACAPS405**	Advanced JAVA and J2EE-Lab	**Practical**	**4**	**40**	**10**	**50**	**2**
8	**BCACAES401**	A) Distributed Computing B) Object Oriented Analysis & Design C) Digital Image Processing	**Elective**	**2**	**40**	**10**	**50**	**2**
9	**BCACASS401**	Basic Web Designing Skills.	**Compulsory**	**2**	**40**	**10**	**50**	**2**
Sub-Total				**33**	**560**	**140**	**700**	**25**

SEMESTER IV

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Python Programming (Theory)**		
Course Code:	**BCACACS401**	No. of Credits	**03**
Contact hours	**52Hours**	Duration of SEE/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the basic concepts of Python Programming.
- Demonstrate proficiency in the handling of loops and creation of functions.
- Identify the methods to create and manipulate lists, tuples and dictionaries.
- Discover the commonly used operations involving file handling.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Develop the emerging applications of relevant fields using Python.

Unit	Description	Hours
1	**Introduction to Features and Applications of Python**; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. **Python Basics**: Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples. **Python Control Flow**: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range () and exit () functions. **Exception Handling**: Types of Errors; Exceptions; Exception Handling using try, except and finally. Python Functions: Types of Functions; **Function Definition-** Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Recursive Functions; Scope and Lifetime of Variables in Functions	**13**
2	**Strings**: Creating and Storing Strings; Accessing String Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifies; Escape Sequences; Raw and Unicode Strings; Python String Methods. **Lists**: Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists. **Dictionaries:** Creating Dictionaries; Operations on Dictionaries; Built-in Functions	**13**

	on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries. **Tuples and Sets**: Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built in Functions on Sets; Set Methods.	
3	**File Handling**: File Types; Operations on Files– Create, Open, Read, Write, Close Files; File Names and Paths; Format Operator. **Object Oriented Programming**: Classes and Objects; Creating Classes and Objects; Constructor Method; Classes with Multiple Objects; Objects as Arguments; Objects as Return Values; Inheritance- Single and Multiple Inheritance, Multilevel and Multipath Inheritance; Encapsulation- Definition, Private Instance Variables; Polymorphism- Definition, Operator Overloading. **GU Interface**: The Tkinter Module; Window and Widgets; Layout Management- pack, grid and place	**13**
4	**Python SQLite**: The SQLite3 module; SQLite Methods- connect, cursor, execute, close; Connect to Database; Create Table; Operations on Tables Insert, Select, Update. Delete and Drop Records. **Data Analysis**: NumPy- Introduction to NumPy, Array Creation using NumPy, Operations on Arrays; Pandas- Introduction to Pandas, Series and DataFrames, Creating DataFrames from Excel Sheet and .csv file, Dictionary 13 and Tuples. Operations on DataFrames. **Data Visualization**: Introduction to Data Visualization; Matplotlib Library; Different Types of Charts using Pyplot- Line chart, Bar chart and Histogram and Pie chart	**13**

Text Books:

1. Introduction to python programming by Gowrishankar S. and Veena A., CRC Press.
2. Core python programming by Dr. R. Nageswara Rao, Dreamtech.

Reference Books:

1. Introduction to Python Programming, Gowrishankar S et al., CRC Press, 2019.
2. Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language, Fabio Nelli, Apress®, 2015
3. Advance Core Python Programming, Meenu Kohli, BPB Publications, 2021.
4. Core PYTHON Applications Programming, Wesley J. Chun, 3rd Edition, Prentice Hall, 2012.
5. Automate the Boring Stuff, Al Sweigart, No Starch Press, Inc, 2015.
6. Data Structures and Program Design Using Python, D Malhotra et al., Mercury Learning and Information LLC, 2021.

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Advanced JAVA and J2EE(Theory)**		
Course Code:	**BCACACS404**	No. of Credits	**03**
Contact hours	**52Hours**	Duration of SEE/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes:

After the successful completion of the course, the student will be able to:

- Identify the need for advanced Java concepts like Enumerations and Collections
- Construct client-server applications using Java socket API
- Make use of JDBC to access database through Java Programs
- Adapt servlets to build server side programs
- Demonstrate the use of JavaBeans to develop component-based Java software

Unit	Description	Hours
1	**Enumerations, Autoboxing and Annotations**(metadata): Enumerations, Enumeration fundamentals, the values() and valueOf() Methods, java enumerations are class types, enumerations Inherits Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at runtime by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations. **Java Beans:** Definition, Advantages of java beans, introspection, bound and constraint properties, persistence, customizers, java beans API,example	**13**
2	**The collections and Framework**: Collections Overview, Recent Changes to Collections, The Collection Interfaces, The Collection Classes, Accessing a collection Via an Iterator, Storing User Defined Classes in Collections, The Random Access Interface, Working with Maps, Comparators. **MVC Architecture in Java**: What is MVC architecture in Java, Advantages of MVC Architecture, Implementation of MVC using Java, MVC Architecture Layers,	**13**
3	**String Handling** :The String Constructors, String Length, Special String Operations, StringLiterals, String Concatenation, String Concatenation with Other Data Types, StringConversion and toString() Character Extraction, charAt(), getChars(), getBytes() toCharArray(), String Comparison, equals() and equalsIgnoreCase(), regionMatches( )startsWith() and endsWith(), equals() Versus == , compareTo() Searching Strings,Modifying a String, substring(), concat(), replace(), trim(), Data Conversion Using valueOf(), Changing the Case of Characters Within a String, **Additional String Methods:-** StringBuffer, StringBuffer Constructors, length() and capacity(), ensureCapacity(),setLength(), charAt() and setCharAt(), getChars(), append(), insert(),	**13**

	reverse(), delete( )and deleteCharAt(), replace(), substring(), Additional StringBuffer Methods, StringBuilder. **RMI Distributed Applications**. How client and server communicate through remote objects. Object Persistence and Serialization, Introduction to Distributed Computing, RMI Architecture, Importance of RMI Registry, Developing Simple RMI application, Callback Implementation in RMI.	
4	**Background; The Life Cycle of a Servlet**; Using Tomcat for Servlet Development; A simpleServlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking. **Java Server Pages** (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects **The Concept of JDBC**; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.	**13**

Text Books:

1. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.
2. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

Reference Books:

1. Y. Daniel Liang: Introduction to JAVA Programming, 7thEdition, Pearson Education, 2007.
2. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education,2004.
3. Uttam K Roy, Advanced JAVA programming, Oxford University press, 2015.

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

4	**Process Synchronization:** Introduction; Race Condition; Critical Section Problem, Semaphores; Classic Problems of Synchronization- Readers and Writers Problem, Dining Philosophers Problem. **Deadlocks:** Deadlock Characterization, Methods of Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.	**15**

Text Book:

1. Abraham Silberschartz and Peter Galvin, Operating System Concepts, 6th edition, TMH

Reference Books:

1. Operating System Concepts - Engineering Handbook, Ghosh PK, 2019.
2. Understanding Operating Systems, McHoes A et al., 7th Edition, Cengage Learning, 2014.
3. Operating Systems - Internals and Design Principles, William Stallings, 9th Edition, Pearson.
4. Operating Systems – A Concept Based Approach, Dhamdhere, 3rd Edition, McGraw Hill Education India.
5. Modern Operating Systems, Andrew S Tanenbaum, 4th Edition, Pearson

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Python Programming - Lab**		
Course Code:	**BCACAPS404**	No. of Credits	**02**
Contact hours	**52 Hours**	Duration of SEE/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

1. Write a program create list with N elements. find all unique elements in the list. If an element is found only once in the list, then add that element to the unique list.
2. Program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
3. Consider a tuple t1= (1,2,5,7,9,2,4,6,8,10). Write a program to perform following operations:
    a) Print half the values of tuple in one line and the other half in the next line.
    b) Print another tuple whose values are even numbers in the given tuple.
    c) Concatenate a tuple t2= (11,13,15) with t1.
    d) Return maximum and minimum value from this tuple.

4. Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count.
5. Write a function nearly equal to test whether two strings are nearly equal. two strings a and b are nearly equal if one character change in b results in string a.
6. Write a program to create a text file and compute the number of characters, words and lines in a file
7. Write a Pandas program to join the two given data frames along rows. Sample Data frame may contain details of student like rollno , name , Total Marks.

PART-B

1. Program to create a class Employee with empno, name, depname, designation, age and salary and perform the following function.
    i) Accept details of N employees
    ii) Search given employee using empno
    iii) Display employee details in neat format.

2. Write a program menu driven to create a BankAccount class. class should support the following methods for
    i) Deposit
    ii) Withdraw

iii) GetBalanace .

Create a subclass SavingsAccount class that behaves just like a BankAccount, but also has an interest rate and a method that increases the balance by the appropriate amount of interest.

3.Create a GUI to input Principal amount, rate of interest and number of years, Calculate Compound interest. When button submit is pressed Compound interest should be displayed in a textbox. When clear button is pressed all contents should be cleared.

4.Write a GUI program to implement Simple Calculator

5. Create a table student table (regno, name and marks in 3 subjects) using MySQL/SQLite and perform the followings

   a. To accept the details of students and store it in database.
   b. To display the details of all the students
   c. Delete particular student record using regno.

6. Create a table employee (empno, name and salary) using MySQL/SQLite and perform the followings

   a. To accept the details of employees and store it in database.
   b. To display the details of a specific employee
   c. To display employee details whose salary lies within a certain range

7.Consider following data and draw the bar graph using matplot library.(Use CSV or Excel).

Batsman	**2017**	**2018**	**2019**	**2020**
Virat Kohli	2501	1855	2203	1223
Steve Smith	2340	2250	2003	1153
Babar Azam	1750	2147	1896	1008
Rohit Sharma	1463	1985	1854	1638
Kane Williamson	1256	1785	1874	1974
Jos Butler	1125	1853	1769	1436

Display appropriate title for axis and chart. Also show legends.

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:5 Marks Execution: 8Marks**	**15 Marks**
Program-2	**PART-B** **Writing:10 Marks Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**
Total		**40 Marks**

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Advanced JAVA and J2EE -Lab**		
Course Code:	**BCACAPS405**	No.of Credits	**02**
Contact hours	**52 Hours**	Duration of SEE/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

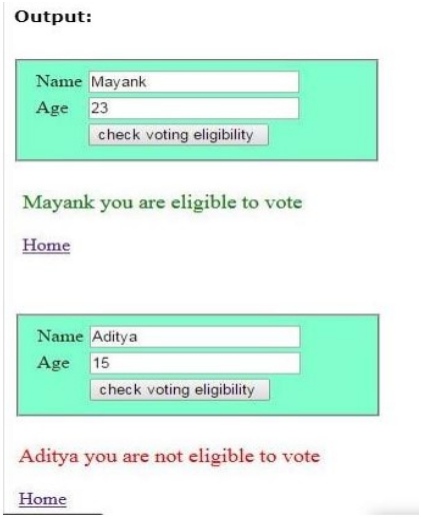
1. Write a program to convert numbers into words using Enumerations with constructors, methods and instance variables.(INPUT RANGE-0 TO 99999)

   EX: 36 THIRTY SIX

2. Find the second maximum and second minimum in a set of numbers using auto boxing and unboxing.
3. Write a menu driven program to create an Arraylist and perform the following operations
   i) Adding elements
   ii) Sorting elements
   iii) Replace an element with another
   iv) Removing an element
   v) Displaying all the elements
   vi) Adding an element between two elements
4. Write a java program to find words with even number of characters in a string, then swap the pair of characters in those words and also toggle the characters in a given string
   EX: Good Morning everyone
   Output: oGdo vereoyen
   gOOD mORNING EVERYONE

5. Write a Servlet program that accepts the age and name and displays if the user is eligible for voting or not

Output:



6. Write a JSP program to print first 10 Fibonacci and 10 prime numbers.
7. Write a java Servlet program to Download a file and display it on the screen(A link has to be provided in HTML, when the link is clicked corresponding file has to be displayed on screen).

PART-B

1. Write a menu driven JDBC program to perform basic operations with Student Table.

MENU

1. **Add new Student**
2. **Delete a specified students Record**
3. **Update Students Address specified students Record**
4. **Search for a particular Student**
5. **Exit**

Student

StRegNo	StName	Stdob	StAddress	StClass	StCourse

2. Write a menu driven JDBC program to perform basic operations with Bank Table.

MENU

1. **Add new Account Holder information.**
2. **Amount Deposit**
3. **Amount Withdrawal (Maintain minimum balance 500 Rs)**
4. **Display all information**
5. **Exit**

Bank

ACC_NO	ACC_NAME	ACC_ADDRESS	BALANCE

3. Write a Java class called Tax with methods for calculating Income Tax. Have this class as a servant and create a server program and register in the rmiregistry. Write a client program to invoke these remote methods of the servant and do the calculations. Accept inputs interactively.

<₹ 3,00,000	No Tax
₹ 3,00,001 to ₹ 6,00,000	5%
₹ 6,00,001 to ₹ 9,00,000	10%
₹ 9,00,001 to ₹ 12,00,000	15%
₹ 12,00,001 to ₹ 15,00,000	20%
>₹ 15,00,000	30%

4. Write a Java class called SimpleInterest with methods for calculating simple interest. Have this class as a servant and create a server program and register in the rmiregistry. Write a client program to invoke these remote methods of the servant and do the calculations. Accept inputs at command prompt.

5. Write a java JSP program to get student information through a HTML and create a JAVA Bean Class, populate Bean and Display the same information through another JSP

6. Write a menu driven program to create a linked list and perform the following operations.
   a. to Insert some Elements at the Specified Position
   b. swap two elements in a linked list
   c. to Iterate a LinkedList in Reverse Order
   d. to Compare Two LinkedList
   e. to Convert a LinkedList to ArrayList

7. Implement a java application based on the MVC design pattern. Input student Rolno, name, marks in three subject calculate result and grade and display the result in neat format.

Percentage of Marks	Grade
Above 90%	A
80% to 90%	B
70% to 80%	C
60% to 70%	D
Below 60%	E

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:5 Marks Execution: 8Marks**	**15 Marks**
Program-2	**PART-B** **Writing:1o Marks Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**
Total		**40 Marks**

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Distributed Computing (Elective)**		
Course Code:	**BCACAES401**	No. of Credits	**02**
Contact hours	**26 Hours**	Duration of SEE/Exam	**2 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

Course Outcomes (COs):

At the end of the course the students will be able to:

- Understand concepts behind Distributed Systems.
- Design and build application programs on distributed systems
- Develop, test and debug RPC based client-server programs.

Unit	Description	Hours
1	**Introduction:** Definition, History, Different Forms Of Computing, Strengths and Weakness Interposes Communications An archetypal IPC Program interface, event synchronization, timeouts and threading, deadlocks and timeouts, data representation (Page 78 only), text based protocols, request response protocols, event diagram and sequence diagram.	**8**
2	**Distributed computing paradigms:** Paradigms and abstraction, An example application, paradigms for distributed applications, tradeoffs. The socket API Background, the socket metaphor in IPC, The datagram socket API The stream mode socket API, The socket with non-blocking I/O operations, secure socket API	**8**
3	**Group communication**- unicasting and multicasting, multicast API, connection oriented versus connectionless multicast, reliable multicast versus unreliable multicasting, the java based multicast API, **Distributed objects**-Message passing versus distributed objects, an archetypal distributed object architecture, Distributed object system, remote procedure calls, Remote method invocation.	**10**

Text Book:

1. M.L.Liu, Distributed Computing-Principles and Applications, Pearson Education, 2004.

Reference Books:

1. Mukesh Singhal, Niranjan G.Shivaratri, Advanced Concepts in Operating System, Tata McGraw Hill

2. Willaim Grosso, Java RMI, Shroff/O'reilly, 2002

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Object Oriented Analysis and Design (Elective)**		
Course Code:	**BCACAES402**	No. of Credits	**02**
Contact hours	**26 Hours**	Duration of SEE/Exam	**2 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

Course Outcomes (COs):

At the end of the course the students will be able to:

- Explain the principles and requirements of OOA and Design
- Describe the object-oriented approach to system development, modeling objects, relationships and interactions.
- Analyze Objects and Classes of the software system.
- Construct object model using object types, attributes, structures and associations.
- Analyze Functional and Dynamic Modeling

Unit	Description	Hours
1	**Introduction** Object orientation concept, OO development concept - Modeling concept, OO methodology, three methods, OO Themes - Abstraction, Encapsulation, combining data & behavior, sharing, Emphasis on the essence of an Object, Synergy **Modeling as a design Technique** Modeling, Abstraction, The three models **Class modeling** Object and class concepts - Objects, Classes, Class diagram, Values & attributes, Operation and methods, Link and Association Concepts - Link and association, Multiplicity, Association and names, Ordering, Bags & Sequences, Association Class, Qualified Association, Generalization and Inheritance- Definition, Use of generalization, Overriding features	**8**
2	**State Modeling** Events - Signal event, change event, Time event, States, Transistors and conditions State Diagrams - Sample State Diagram, one shot state Diagrams, Summary of Basic state diagram notations, State Diagram Behavior - Activity Effects, Do Activities, Entry and Exit Activities, Completion Transition, Sending Signals **Sequence Model**: Scenarios, Sequence Diagram, Communication Diagram, Activity Model - Activities, Branches, Introduction & termination, Concurrent Activities, Executable Activity diagram, Guidelines for Activity models, Deployment Diagram	**8**

3	**Class Design** Overview of Class Design, Bridging the Gap, Realizing Use Cases, Designing Algorithms - Choosing Algorithms, Choosing Data structures, Defining Internal classes and Operations, Assigning Operations to Classes, Recursing Downward - Functionality Layers, Mechanism Layers, Refactoring, Design Optimization - Adding Redundant associations for Efficient Access, Saving derived values to avoid Re-computation, Rectification of Behavior, Adjustment of Inheritance - Rearranging Classes and Operations, Abstracting out Common Behavior, Using Delegation to share Behavior	**10**

Text Book:

1. Object Oriented Modeling and Design with UML Michael R. Blaha James R. Rumbaugh, Second Edition, Pearson

Reference Books:

1. UML™ 2 ToolKit – Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, WILEY Publishing
2. Object Oriented Analysis and Design with Applications Grady Booch Second Edition (Pearson Education)
3. Object Oriented Software Engineering Bernd Brugge and Allen H. Dutoit Pearson Education

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Digital Image Processing (Elective)**		
Course Code:	**BCACAES403**	No. of Credits	**02**
Contact hours	**26 Hours**	Duration of SEE/Exam	**2 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

Course Outcomes (COs):
After the successful completion of the course, the student will be able to:
.

- Remember the fundamental concepts of Image Processing
- Explain different Image enhancement techniques
- Understand and review image transforms
- Analyze and evaluate digital images.
- Apply digital image techniques in real world scenarios

Unit	Description	Hours
1	**Introduction:** What is Digital image processing, The origin of DIP, Examples of fields that use DIP, Fundamentals steps in DIP, Components of an image processing system. **Digital Image Fundamentals:** Elements of Visual perception, Light and the electromagnetic spectrum, Image sensing and acquisition, Image sampling and Quantization, Some Basic relationship between Pixels	**8**
2	**Image Enhancement in the Spatial Domain:** Background, some basic Gray Level Transformations, Histogram Processing, Enhancement using Arithmetic / Logic operations, Basics of spatial filtering, Smoothing Spatial Filters, Sharpening spatial filters.	**8**
3.	**Color Image Processing:** Color Fundamentals, Color Models, Pseudocolor Image Processing, Color transformations, Smoothing and Sharpening, Color Segmentation, Noise in Color Images	**10**
Text Books: 1.Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Second Edition, PHI/Pearson **Reference Books:** 1. B. Chanda, D. Dutta Majumder, "Digital Image Processing and Analysis", PHI, 2003.		

2. Nick Efford, “Digital Image Processing a practical introducing using Java”, Pearson Education, 2004.
3. Education. Alexander M., Abid K., “OpenCV-Python Tutorials”, 2017.
4. Kothari, Ashish M, Digital Image Processing using SCILAB, Springer publication, 2019.

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/ Trouble Shooting.

Program Name	**BCA-GENERAL**	Semester	**IV**
Course Title	**Basic Web Designing Skills (Compulsory)**		
Course Code:	**BCACASS401**	No. of Credits	**02**
Contact hours	**26 Hours**	Duration of SEE/Exam	**2 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand the fundamentals of HTML5 and its evolution from previous versions.
- Identify the structure and components of an HTML5 document.
- Utilize HTML5 semantic elements to create well-structured web pages.
- Implement multimedia elements such as audio and video using HTML5.
- Demonstrate the use of HTML5 forms and input types for user data collection.
- Apply best practices for web accessibility and SEO in HTML5 documents.

Unit	Description	Hours
1	**Introduction to Computers and the Internet-**Introduction, The Internet in Industry and Research, HTML5, CSS3, Demos, Evolution of the Internet and World Wide Web, Web Basics. **Introduction to HTML5:** Introduction, Editing HTML5, First HTML5 Example, W3C HTML5 Validation Service, Headings, Linking, Images, Special Characters and Horizontal Rules, Lists, Tables, Forms, Internal Linking, meta Elements. **New HTML5 Form input Types**, input and data list Elements and autocomplete Attribute, Page-Structure Elements.	**8**
2	**Cascading Style Sheets-**Introducing CSS, Where You Can Add CSS Rules, CSS Properties-Controlling Fonts, Text Formatting, Text Pseudo-Classes, Selectors, Lengths, Percentages.	**8**
3	**More Cascading Style Sheets: -**Links, Backgrounds, Lists, Tables, Outlines, The: focus and: active Pseudo-Classes.	**10**

Text Books:

1. Deitel, Paul_Deitel, Harvey_Deitel, Abbey - Internet and World Wide Web How to Program-Pearson Education (US) (2011)
2. Jon Duckett -Beginning Web Programming with HTML, XHTML, and CSS (Wrox Beginning

Guides)-Wrox (2004)

Reference Books:

1. The Complete Reference HTML and CSS, 5th Edition, Thomas A Powell, 2017.
2. Animation in HTML, CSS, and JavaScript, Kirupa Chinnathambi, 1st Edition, Create space Independent Pub, 2013
3. Web Programming with HTML5, CSS, and JavaScript-John Dean

Pedagogy: Lecture/ PPT/ Videos/ Animations/Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/Mini Projects/Problem Solving/Trouble Shooting.

Questions Paper for Pattern Core Subjects

Duration:3 Hours Max.Marks:80

Note: Answer any ten Questions from Part-A. And one full Questions from each unit in Part-B

Part-A

1. 10*2=20

a.
b.
c.
d.
e.
f.
g.
h.
i.
j.
k.
l.

Part-B

UNIT-I, II, III, IV

Each unit contain main questions and it carry 15 Marks.
Each main questions contain 2 or more sub question.

4*15=60

UNIT-I

2.

a.
b.
c.

3.

a.
b.
c.

UNIT-II

4.

a.

b.

c.

5.

a.

b.

c.

Questions Paper Pattern for Elective and Compulsory Subjects

Duration:2 Hours Max.Marks:40

Note: Answer any Five from Eight questions from Part-A. And any six Questions out of Nine Questions from Part-B

Part-A

1. 5*2=10

a.
b.
c.
d.
e.
f.
g.
h.

Part-B

Answer any Six questions out of Nine questions.

6*5=30

2.
3.
4.
5.
6.
7.
8.
9.
10.