

**Yenepoya (Deemed to be University)**

**Recognized under Sec 3(A) of the UGC Act 1956 Accredited by NAAC with ‘A+’ Grade**

**Bachelor of Computer Applications (BCA)**

**Specialization in**

1. Cloud Computing, Cybersecurity and Ethical Hacking
2. Cybersecurity, Ethical Hacking and Digital Forensics
3. AIML, Robotics and IoT
4. Cybersecurity, Digital Forensics and Data Science
5. Game Development and AR/VR
6. Artificial Intelligence, Cloud Computing and DevOps
7. Data Science and Big Data Analytics

Offered by

**The Yenepoya Institute of Arts, Science, Commerce and Management (YIASCM)**

**Under the**

**FACULTY OF SCIENCE**

**Yenepoya (Deemed to be University)**

|  |  |
| --- | --- |
| PO1 | Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering and business practices. |
| PO2 | Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| PO3 | Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large. |
| PO4 | Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |
| PO5 | Problem analysis: Identify, formulate, review research literature, and analyze complex computational and business problems. |
| PO6 | Design/development of solutions: Design solutions for complex problems and design system components or processes that meet the specified needs. |
| PO7 | Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| PO8 | Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern IT tools including prediction and modeling to complex computing activities with an understanding of the limitations |

**Semester 3 / Year 2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Semester 3** | | | | | | | | | | |
| **SL. No.** | **Course Code** | **Course Type** | **Course Title** | **Hours per Week**  **(Theory | Practical | Experiential / Project)** | | | **CIA** | **SEE** | **Total** | **Credits** |
| 1 | BCA301T | CC | Data Structures and Algorithms (Theory) | 3 | 0 | 0 | 50 | 50 | 100 | 3 |
| 2 | BCA301P | CC | Data Structures and Algorithms (Practical) | 0 | 2 | 0 | 50 | 50 | 100 | 1 |
| 3 | BCA302T | CC | Python Programming (Theory) | 2 | 0 | 0 | 50 | 50 | 100 | 2 |
| 4 | BCA302P | CC | Python Programming (Practical) | 0 | 4 | 0 | 50 | 50 | 100 | 2 |
| 5 | BCA303 | CC | Fundamentals of Cybersecurity and Cryptography | 2 | 0 | 3 | 50 | 50 | 100 | 3 |
| 6 | BCA304 | CC | Introduction to Artificial Intelligence | 2 | 0 | 3 | 50 | 50 | 100 | 3 |
| 7 | BCA305 | ME | ME Group 3 | 3 | 0 | 0 | 25 | 25 | 50 | 3 |
| 8 | BCA306 | VAC | VAC Group 3 | 2 | 0 | 0 | 100 | - | 100 | 2 |
| 9 | BCA307 | SEC | SEC Group 1 | 3 | 0 | 0 | 100 | - | 100 | 3 |
|  |  |  |  |  |  |  |  |  |  | **22** |

**Semester 3 / Year 2**

|  |  |  |
| --- | --- | --- |
| Semester 3 | BCA301T | Data Structures and Algorithms |
| 3 Credits | TPE : 3:0:0 | Core Course – CC |

**Course Outcomes**: On successful completion of the course the learner will be able to

|  |  |
| --- | --- |
| **CO#** | **Course Outcomes** |
| BCA301T.1 | Differentiate primitive and non-primitive structures. |
| BCA301T.2 | Design and apply appropriate data structures for solving computing problems. |
| BCA301T.3 | Apply sorting and searching algorithms to the small and large data sets. |
| BCA301T.4 | Implement various algorithms on various data structures |

**Data Structures and Algorithms (Theory)**

|  |
| --- |
| **Course Contents:** |
| **Module 1: Basics of Strutures (9 Hours)** |
| Structure and Problem Solving, Arrays, Pointers, structure (struct), type definitions (typedef), memory allocation, recursion, Data structures, Data structure Operations, Algorithm: complexity, Time-space trade-off. |
|  |
| **Module 2: Linked Lists (9 Hours)** |
| Introduction, Array Representation of Stack, Linked List Representation of stack, Application of stack - expressions - infix, postfix, prefix, evaluation of expressions using stack, Queue, Array Representation of Queue, Linked List Representation of Queue. Types of queues, Linked lists, Representation of linked lists in Memory, Traversing a linked list, Searching a linked list, Memory allocation and Garbage collection, insertion into linked list, Deletion from a linked list, Types of linked list. |
|  |
| **Module 3: Sorting (9 Hours)** |
| Searching Techniques, Linear, Binary, Sorting Techniques: Insertion sort, Selection sort, Bubble sort, Merge sort, Quick Sort, Complexities of searching and sorting algorithms. |
|  |
| **Module 4: Trees (9 Hours)** |
| Definitions and Concepts, Types of trees, Operations on Binary Trees, Representation of binary tree, Tree Traversal - inorder, pre order, postorder. Binary Search Tree, various operations on BST |
| **Module 5: Graphs (9 Hours)** |
|  |
| Graphs: Definitions and Concepts, Types of graphs, Matrix Representation of Graphs, List Representations of Graphs, Types of Directed Graphs; Binary Relation As a Digraph; Euler’s Digraphs; Matrix Representation of Digraphs, Breadth First Search, Depth First Search, Spanning Trees. |
|  |
| **References:** |
| * Samanta, D. (2003). Classic data structures. Prentice-Hall of India. |
| * Horowitz, E., Sahni, S., & Anderson-Freed, S. (1992). Fundamentals of data structures in C. WH Freeman & Co.. |
| * Mark Allen Weiss. (2005). Data structures & algorithm analysis in C++. Addison-Wesley. |
| * Skiena, S. S. (1998). The algorithm design manual (Vol. 2). New York: springer. |
| * Aho Alfred, V., Hopcroft John, E., Ullman Jeffrey, D., Aho Alfred, V., Bracht Glenn, H., Hopkin Kenneth, D., ... & Johnson, C. A. (1983). Data structures and algorithms. USA: Addison-Wesley. |
| * Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2008). Algorithms. New York: McGraw-Hill Higher Education. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Semester 3 | BCA301P | Data Structures and Algorithms (Practical) | | Credits 1 | TPE: 0:2:0 | Core Course – CC |     **Course Outcomes**: On successful completion of the course the learner will be able to   |  |  | | --- | --- | | CO# | Course Outcomes | | BCA301P.1 | Understanding of common sorting and searching algorithms such as bubble sort, insertion sort, quicksort, and binary search. | | BCA301P.2 | Understanding the basics of data structures such as arrays, linked lists, stacks, queues, trees, and graphs. | | BCA301P.3 | Ability to apply data structures and algorithms to solve various types of problems. | | BCA301P.4 | Ability to design and implement efficient algorithms to solve a given problem. | |
| **Data Structures and Algorithms (Practical)**  **Program List:** |
| 1. Write a program to demonstrate linear and binary search. |
| 2. Write a program to demonstrate selection sort. |
| 3. Write a program to demonstrate insertion sort. |
| 4. Write a program to demonstrate bubble sort. |
| 5. Write a program to demonstrate merge sort. |
| 6. Write a program to demonstrate quick sort. |
| 7. Write a program to implement stack using array. |
| 8. Write a program to implement queue using arrays |
| 9. Write a program to implement a linked list and perform various operations. |
| 10. Write a program to evaluate postfix expressions. |
| 11. Write a program to perform inorder, preorder and post order traversal of a binary tree. |
| 12. Write a program to implement BFS and DFS. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Semester 3 | BCA302T | Python Programming (Theory) | | Credits 2 | TPE: 2:0:0 | Core Course – CC |     **Course Outcomes:** On successful completion of the course the learner will be able to   |  |  | | --- | --- | | **CO#** | **Course Outcomes** | | BCA302T.1 | Explain the basic concepts of Python Programming. | | BCA302T.2 | Demonstrate proficiency in the handling of loops and creation of functions. | | BCA302T.3 | Identify the methods to create and manipulate lists, tuples and dictionaries. | | BCA302T.4 | Discover the commonly used operations involving file handling. | | BCA302T.5 | Interpret the concepts of Object-Oriented Programming as used in Python. | | BCA302T.6 | Develop the emerging applications of relevant fields using Python | |
| **Python Programming (Theory)** |
| **Course Contents:** |
| **Module 1: Introduction (6 Hours)** |
| 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 |
|  |
| **Module 2: Exception Handling (6 Hours)** |
| 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; Keyword Arguments; Recursive Functions; Scope and Lifetime of Variables in Functions. Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods. |
|  |
| **Module 3: Lists (6 Hours)** |
| 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 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 |
|  |
| **Module 4: File Handling (6 Hours)** |
| 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. |
|  |
| **Module 5: GU Interface (6 Hours)** |
| GU Interface: The tkinter Module; Window and Widgets; Layout Management- pack, grid and place. 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 and Tuples. Operations on DataFrames. Data Visualisation: Introduction to Data Visualisation; Matplotlib Library; Different Types. |
|  |
| **References:** |
| * Brown, F., & George, P. (2001). Python: the complete reference. |
| * Matthes, E. (2019). Python crash course: A hands-on, project-based introduction to programming. no starch press. |
|  |

|  |  |  |
| --- | --- | --- |
| Semester 3 | BCA302P | Python Programming (Practical) |
| Credits 2 | TPE: 0:4:0 | Core Course – CC |

**Course Outcomes:** On successful completion of the course the learner will be able to

|  |  |
| --- | --- |
| **CO#** | **Course Outcomes** |
| BCA302P.1 | Understanding the basics of Python programming language including syntax, data types, and control structures. |
| BCA302P.2 | Knowledge of popular Python libraries and frameworks such as NumPy, Pandas, Matplotlib, Flask, and Django. |
| BCA302P.3 | Ability to write Python programs to solve various types of problems. |
| BCA302P.4 | Understanding of object-oriented programming concepts and how to apply them in Python. |

**Python Programming (Practical)**

**Programs List:**

**Part-A**

1. Check if a number belongs to the Fibonacci Sequence.

2. Solve Quadratic Equations.

3. Find the sum of n natural numbers.

4. Display Multiplication Tables.

5. Check if a given number is a Prime Number or not

6. Implement a sequential search.

7. Create a calculator program.

8. Explore string functions.

9. Implement Selection Sort.

10. Implement Stack.

11. Read and write into a file.

**Part-B**

1. Demonstrate usage of basic regular expressions.

2. Demonstrate use of advanced regular expressions for data validation.

3. Demonstrate use of List.

4. Demonstrate use of Dictionaries.

5. Create SQLite Database and Perform Operations on Tables.

6. Create a GUI using the Tkinter module.

7. Demonstrate Exceptions in Python.

8. Drawing Line chart and Bar chart using Matplotlib.

9. Drawing Histogram and Pie chart using Matplotlib.

10. Create Array using NumPy and Perform Operations on Array.

11. Create DataFramefrom Excel sheet using Pandas and Perform Operations on DataFrames

|  |  |  |
| --- | --- | --- |
| Semester 3 | BCA303 | Fundamentals of Cybersecurity and Cryptography |
| Credits 4 | TPE : 3: 0: 3 | Core Course – CC |

**Course Outcomes**: On successful completion of the course the learner will be able to

|  |  |
| --- | --- |
| **CO#** | **Course Outcomes** |
| BCA303.1 | Understanding of basic cybersecurity concepts and the importance of cybersecurity in protecting information systems. |
| BCA303.2 | Understanding of cryptography and its role in providing confidentiality, integrity, and authentication in information systems. |
| BCA303.3 | Familiarize with different types of cryptosystems |
| BCA303.4 | Create an awareness for the design of various cryptographic primitives |
| BCA303.5 | Analyse different types of attacks on various cryptosystems |

**Fundamentals of Cybersecurity and Cryptography**

**Course Contents:**

**Module 1: Basics of Algebra and Number Theory (9 Hours)**

Integer Arithmetic Modular Arithmetic, Algebraic structures, Prime Numbers, Fermat’s and Euler’s Theorem, Factorization, Chinese, Remainder Theorem - Linear and Quadratic Congruence, Discrete Logarithms

**Module 2: Introduction to Security (9 Hours)**

Security Goals, Security services, Confidentiality, Integrity, Authentication, Non-repudiation, Access control, Security Mechanisms (Encipherment, Data Integrity, Digital Signature, Authentication Exchange, Traffic Padding, Routing Control, Notarization, Access control, Security Principles.

**Module 3: Ciphers (9 Hours)**

Traditional Secret Key Ciphers: Substitution Ciphers (mono alphabetic ciphers, poly alphabetic ciphers)-Transposition Ciphers-Stream and Block Ciphers. Modern Secret KeyCiphers: Substitution Box-Permutation Box-Product Ciphers

**Module 4: Data Encryption Standard (9 Hours)**

Data Encryption Standard (DES) (Fiestel and Non-Fiestel Ciphers, Structure of DES, DES Attacks, 2-DES, 3-DES) - Advanced Encryption Standard (AES) (Structure, Analysis) Cryptographic Hash Functions, Properties, Secure Hash Algorithm, Message Authentication Code (MAC)

**Module 5: Public Key Cryptosystems (9 Hours)**

Public Key Cryptosystems (PKC): Types of PKC, Trapdoor, one way functions, RSA Cryptosystem (Integer Factorisation Trapdoor, Key Generation, Encryption, Decryption), El Gamal Cryptosystem (Discrete Logarithm Trapdoor, Key Generation, Encryption, Decryption), Diffie-Hellman Key Exchange Protocol, Man in the Middle attack on Diffie-Hellman Protocol

**Text Books:**

1. Behrouz A. Forouzan and Debdeep Mukhopadhyay, Cryptography & Network Security, Second Edition, Tata McGraw Hill, New Delhi, 2010
2. Douglas R. Stinson, “Cryptography: Theory and Practice”, Third Edition, CRC Press.
3. William Stallings, “Cryptography and Network Security – Principles and Practices”, Pearson Education, Fourth Edition, 2006.
4. Atul Kahate, “Cryptography and Network Security”, 2nd Edition, Tata McGraw Hill,2003.

|  |  |  |
| --- | --- | --- |
| Semester 3 | BCA304 | Introduction to Artificial Intelligence |
| Credits 3 | TPE : 2: 0: 3 | Core Course – CC |

**Course Outcomes**: On successful completion of the course the learner will be able to

|  |  |
| --- | --- |
| **CO#** | **Course Outcomes** |
| BCA304.1 | To develop semantic-based and context-aware systems to acquire, organize, process, share and use the knowledge embedded in multimedia content. |
| BCA304.2 | Research will aim to maximize automation of the complete knowledge lifecycle and achieve semantic interoperability between Web resources and services. |
| BCA304.3 | The field of Robotics is multi-disciplinary as robots are an amazingly complex system comprising mechanical, electrical, electronic H/W and S/W and issues germane to all these. |

|  |
| --- |
| **Introduction to Artificial Intelligence**  **Course Contents:** |
| **Module 1: Scope of AI (8 Hours)** |
| Problem solving and Scope of AI Introduction to Artificial Intelligence. Applications Games, theorem proving, natural language processing, vision and speech processing, robotics, expert systems. AI techniques- search knowledge, abstraction. Problem Solving State space search; Production systems, search space control: depth-first, breadth-first search. Heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis. LA\* Algorithm, L(AO\*) Algorithm |
|  |
| **Module 2: Knowledge Representation (7 Hours)** |
| Knowledge Representation Knowledge Representation issues, first order predicate calculus, Horn Clauses, Resolution, Semantic Nets, Frames, Partitioned Nets, Procedural Vs Declarative knowledge, Forward Vs Backward Reasoning |
|  |
| **Module 3: Natural Language Processing (7 Hours)** |
| Understanding Natural Languages Introduction to NLP, Basics of Syntactic Processing, Basics of Semantic Analysis, Basics of Parsing techniques, Basics of grammar free analyzers, Basics of sentence generation, and Basics of translation |
| **Module 4: Expert System (8 Hours)** |
| Expert System: Need and justification for expert systems, knowledge acquisition, Case studies: MYCIN, R1 Learning: Concept of learning, learning automation, genetic algorithm, learning by inductions, neural nets |
|  |
| **References:** |
| * E. Rich and K. Knight, “Artificial intelligence”, TMH, 2nd ed., 1992. * Nilsson, N. J. (1986). Principles of artificial intelligence. Morgan Kaufmann. * Craig, J. J. (2009). Introduction to robotics: mechanics and control, 3/E. Pearson * Education India. * Klafter, R. D., Chmielewski, T. A., &Negin, M. (1989). Robotic engineering : an integrated approach. Prentice-Hall. * Yoshikawa, T. (1990). Foundations of robotics: analysis and control. MIT press. |