**SESSION PLAN**

**Academic Year: 2024 – 2025**

**Day 1-2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO** | **Topic** | **Day 1-2: Unit I - Introduction, Data, Expressions, Statements** | **CO** | **BTL** | **PO** |
| **1** | **•** Python interpreter, interactive mode  • Data types: int, float, boolean, string, list  • Variables, expressions, and operator precedence  • Functions: definition, parameters, arguments  • Illustrative programs:  1. Exchange values of two variables  2. Circulate values of n variables | **Assignment:**  1. Write a program to swap two variables without using a third variable.  2. Calculate the distance between two points using functions.  **Progress Analytics:**  • Ensure students understand Python basics and syntax through assignment completion rates. | CO1 |  | POs Addressed: PO1,  PO2,  PO1,2 |

**Day 3-4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO** | **Topic** | **Day 3-4: Unit II - Control Flow, Functions** | **CO** | **BTL** | **PO** |
|  | * Conditional statements: if, if-else, if-elif-else * Loops: for, while, break, continue * Functions: return values, recursion * String handling: slicing, methods, immutability * Illustrative programs:   1. Compute GCD using recursion   2. Perform linear and binary search | **Assignment:**   1. Write a program to check if a string is a palindrome. 2. Develop a recursive function to compute factorial.   **Progress Analytics:**   * Evaluate control flow and recursion implementation via coding accuracy. | **COs Covered:** CO2 |  | **POs Addressed:** PO1, PO2, PO3 |

**Day 5-6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO** | **Topic** | **Day 5-6: Unit III - Lists, Tuples, Dictionaries** | **CO** | **BTL** | **PO** |
|  | **•** Lists: operations, slicing, mutability, comprehension  • Tuples: assignment, return values  • Dictionaries: methods and applications  • Sorting algorithms: selection, insertion, merge  • Illustrative programs:  1. Create a histogram of word frequencies  2. Implement merge sort | **Assignment:**  1. Write a program to find the most frequent word in a list of strings.  2. Implement selection sort on a list of integers.  **Progress Analytics:**  • Track list and dictionary usage proficiency through submission reviews**.** | COs Covered: CO3 |  | POs Addressed: PO1, PO3, PO9 |

**Day 7-8**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO** | **Topic** | **Day 7-8: Unit IV - Files, Modules, Packages** | **CO** | **BTL** | **PO** |
|  | • File handling: read, write, append  • Exception handling: try-except, raising exceptions  • Modules and packages: importing libraries, creating packages  • Illustrative programs:  1. Count the number of words in a text file  2. Copy contents of one file to another | **Assignment:**   1. Write a program to log user data into a text file. 2. Create a package for basic mathematical operations.   **Progress Analytics:**   * Check the ability to handle file I/O operations and modularize code. | **COs Covered:** CO4 |  | **POs Addressed:** PO1, PO2, PO5 |

**Day 9-10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. NO** | **Topic** | **Day 9-10: Unit V - Python for Data Science** | **CO** | **BTL** | **PO** |
| **1** | • Data manipulation: loading, cleaning, and normalizing data  • Reading/writing CSV files, SQL database access  • Simple plotting using Matplotlib  • Debugging and code profiling  • Illustrative programs:  1. Load a CSV file and calculate column-wise averages  2. Plot a bar graph of word frequencies | **Assignment:**   1. Write a program to read a CSV file and filter rows based on specific conditions. 2. Develop a Python script to normalize a dataset.   **Progress Analytics:**   * Assess data analysis skills via data manipulation and visualization tasks. | **COs Covered:** CO5 | **4** | **POs Addressed:** PO2, PO3, PO5, PO9 |

**Day 11: Comprehensive Review and Quiz**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. NO** | **Activities:** | **CO** | **BTL** | **PO** |
| **1** | * Recap of all units with Q&A sessions * Quiz on Python programming fundamentals, control flow, and data structures * Discuss key concepts in file handling and data science applications | **COs Covered:** All | **4** | **POs Addressed:** All |

**Day 12: Project-Based Learning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. NO** | **Task:** | **CO** | **BTL** | **PO** |
| **1** | Students will work in teams to develop a real-world Python project, such as:   * A basic text analyzer (word count, sentiment analysis). * A mini data analysis pipeline using Python libraries. | **COs Covered:** All | **4** | **POs Addressed:** All |
|  | **Evaluation:**   * Peer review of projects and presentations. |  |  |  |

**Assessment Plan**

| **Assessment Type** | **Weightage** | **COs Assessed** |
| --- | --- | --- |
| Assignments (5 Units) | 30% | CO1-CO5 |
| Quizzes (Mid and Final) | 20% | CO1-CO4 |
| Project Evaluation | 30% | CO3-CO5 |
| Participation and Q&A | 10% | CO1-CO5 |
| Attendance | 10% | - |

**Text Book:**

1. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist ‘‘, 2nd edition, Updated for Python 3, Shroff/O ‘Reilly Publishers, 2016 (http://greenteapress.com/wp/think- python/)
2. Guido van Rossum and Fred L. Drake Jr, ―An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.

**Reference Book:**

1. John V Guttag, ―Introduction to Computation and Programming Using Python‘‘, Revised and expanded Edition, MIT Press , 2013.
2. Robert Sedgewick, Kevin Wayne, Robert Dondero, ―Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
3. Timothy A. Budd, ―Exploring Python‖, Mc-Graw Hill Education (India) Private Ltd.,,2015.
4. Kenneth A. Lambert, ―Fundamentals of Python: First Programs‖, CENGAGE Learning, 2012.
5. Charles Dierbach, ―Introduction to Computer Science using Python: A Computational Problem- Solving Focus, Wiley India Edition,

2013.

1. Paul Gries, Jennifer Campbell and Jason Montojo, ―Practical Programming: An Introduction to Computer Science using Python 3‖,

Second edition, Pragmatic Programmers, LLC, 2013.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Course Outcomes** | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** | **P9** | **P10** | **P11** | **P12** | **PSO1** | **PSO2** |
| **CO1** | CO1 Understand and Implement Key Principles: Gain a thorough understanding of encapsulation, inheritance, and polymorphism principles, and effectively implement them in Java programming. | 3 | 3 |  | 1 | 2 |  |  |  |  |  |  |  | 2 |  |
| **CO2** | CO2 Design Real-Time Applications: Design and develop a real-time application capable of managing data efficiently, including operations for storage, retrieval, deletion, and updating. | 2 | 2 | 1 | 3 |  |  |  |  |  |  |  |  |  |  |
| **CO3** | CO3 Develop Robust Exception Handling: Implement robust exception handling mechanisms to ensure the application maintains stability and reliability during runtime. | 3 | 3 | 2 |  | 2 |  |  |  |  |  |  |  | 1 | 2 |
| **CO4** | CO4 Enhance Performance with Multithreading: Apply multithreading techniques to enhance the performance and responsiveness of the application, allowing for the concurrent execution of tasks. | 3 | 2 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |
| **CO5** | CO5 Seamlessly Integrate Data Sources: Connect to various data sources using simple Java APIs, ensuring seamless integration and access to external data. | 3 | 2 | 2 | 1 |  |  |  |  |  |  |  |  | 2 |  |
| **CO6** | CO6 Create Interactive User Interfaces: Utilize AWT (Abstract Window Toolkit) and Applets to create user interfaces for the real-time application, providing an intuitive and interactive experience for users. (Applicable for Python as well) | 3 | 2 | 2 |  |  |  |  |  |  |  |  |  |  | 2 |

**Notes:**

1. **Mapping Explanation**:
   * **3**: Strongly aligned.
   * **2**: Moderately aligned.
   * **1**: Slightly aligned.
   * Blank: No alignment.
2. **CO6 - Applicability for Python**: You have explicitly mentioned "same for Python." This implies that the mapping remains identical for applications using Python, ensuring alignment consistency across platforms.
3. **Ensure Alignment with Curriculum Goals**: If needed, review the alignment with the curriculum's vision and mission to maintain relevance and coverage of intended skills and knowledge areas.