**B.E. in COMPUTER SCIENCE AND ENGINEERING**

Choice-Based Credit System (CBCS) applicable for 2024 Scheme

**SEMESTER – I/II**

**Introduction to Python Programming** (2:0:2) 3

(Effective from the academic year 2024-25)

| | | | |
|---|---|---|---|
| Course Code | **BPLC15B/25B** | CIE Marks | 50 |
| Teaching Hours/Week (L:T:P) | 2:0:2 | SEE Marks | 50 |
| Total Number of Contact Hours | 26 hours of theory<br>+<br>14 hours of Practical | Exam Hours | 3 |

**Course Objectives:**

This course will enable students to:

1. Learn fundamental features of Python
2. Set up Python IDE to create, debug and run simple Python programs.
3. Learn object oriented concepts using programming examples.
4. Study the concepts of modular programming & recursion.

**Preamble:** Python is an open-sourced programming language that combines the features of C and Java. It has exceptional procedural as well as object-oriented capabilities. Its simplicity and readability make it an ideal choice for beginners, while its extensive libraries and frameworks offer advanced capabilities for experienced developers. This course introduces undergraduate students to the fundamentals of Python, focusing on core concepts such as variables, control structures, data types, and functions. By the end of the course, students will be equipped with the skills needed to write efficient Python programs and apply them to solve real-world problems across various domains.

**Module – 1**

**Python Basics**: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, **Flow control:** Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit()

**Textbook 1: Chapters 1 – 2** **(5 Hours)**

**Module – 2**

**Functions:** def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number

**Lists:** The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples, References

**Textbook 1: Chapters 3-4** **(5 Hours)**

**Module – 3**

**Dictionaries and Structuring Data:** The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things

**Manipulating Strings:** Working with Strings, Putting Strings Inside Other Strings, Useful String Methods, Numeric Values of Characters with the ord() and chr() Functions, Coping and Pasting Strings with the pyperclip Module, Project: Multi-Clipboard Automatic Messages

**Textbook 1: Chapter 5-6** **(5 Hours)**

**Module – 4**

**Reading and Writing Files:** Files and File Paths, The File Reading/Writing Process, Saving Variables with the shelve Module,Saving Variables with the print.format() Function

**Working with Excel Spreadsheets :** Excel Documents, Installing the openpyxl Module, Reading Excel Documents

**Working with CSV files and JSON data :** The csv Module, Project: Removing the Header from CSV Files, JSON and APIs, The json Module

**Textbook 1: Chapters 9,13,16** **(6 Hours)**

**Module – 5**

**Classes and objects:** Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying

**Classes and functions:** Time, Pure functions, Modifiers

**Classes and methods:** Object-oriented features, Printing objects, Another example, A more complicated example, The init method, The str method, Operator overloading, Type-based dispatch, Polymorphism, Interface and implementation

**Textbook 2: Chapters 15 – 17** **(5Hours)**

| Sl. No. | Experiments |
|---|---|
| 1. | a. Develop a program to read the student details like Name, USN, and Marks in three subjects. Display the student details, total marks and percentage with suitable messages.<br>b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not. |
| 2. | a. Develop a program to generate Fibonacci sequences of length (N). Read N from the console.<br>b. Write a function to calculate the factorial of a number. Develop a program to compute binomial coefficient (Given N and R). |
| 3. | Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages. |
| 4. | Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with a suitable message. |
| 5. | Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]] |
| 6. | Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()]. |
| 7. | Create a program multiplicationTable.py that takes a number N from the command line and creates an N×N multiplication table in an Excel spreadsheet. |
| 8. | Consider a studData.csv file. File has the USN, Name and CGPA of the students in the class. Develop a program to find the first topper of the class. |
| 9. | Define a function which takes TWO objects representing complex numbers and returns a new complex number with an addition of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition<br>of N complex numbers. |
| 10. | Develop a program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details. [Hint: Use |

| | |
|---|---|
| | list to store the marks in three subjects and total marks. Use __init__() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the score card details.] |

**Course Outcomes:** The students will be able to

CO1: Demonstrate problem-solving techniques and Python constructs.
CO2: Apply Python programming concepts to solve practical problems.
CO3: Analyse concepts of Object-Oriented Programming as used in Python.
CO4: Develop solutions to problems by breaking them down into modular components.

**Textbooks:**

1. Al Sweigart,"Automate the Boring Stuff with Python Practical Programming for total beginners", 2nd Edition, No Starch Press, 2019. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) (Chapters 1 to 18, except 12) for lambda functions use this link:https://www.learnbyexample.org/python-lambda-function/
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)

**Reference Books:**

1. Introduction to Computer Science Using Python: A Computational Problem-Solving Focus, by Charles Dierbach, Wiley India Edition, 2018, ISBN : 978-81-265-5601-4
2. PYTHON PROGRAMMING AN INTRODUCTION TO COMPUTER SCIENCE, John Zelle, Franklin, Beedle & Associates Inc,2016,ISBN 9781590282755

Practical Programming, Third Edition An Introduction to Computer Science Using Python 3.6 by by Paul Gries, Jennifer Campbell, Jason Montojo,O'Reilly; 3rd edition (9 January 2018)

**Alternate Assessment Tools (AATs) suggested:**

- Conduct on spot problem solving based on Python Programming
- Use of Online Platforms: Utilize platforms like LeetCode, HackerRank, or CodeSignal to set up assessments that test specific Python skills through problem-solving tasks.

**Web links / e – resources:**

- **https://www.learnbyexample.org/python/**
- **https://www.learnpython.org/**
- **https://pythontutor.com/visualize.html#mode=edit**