Semester: I/II			
Course Title	Introduction to Python Programming	**CIE Marks**	50
Course Code	BCSPLC105/205A	**SEE Marks**	50
Course Type	Integrated	**Total Marks**	100
Course Category	PLC	**SEE Duration**	03 hrs
Hours Per Week	2L: 0T: 2P	**SEE Type**	Written
Total Hours	40 hrs	**Credits**	03

Course Learning Objectives
This course will enable the students to
• Learn the syntax and semantics of the Python programming language.
• Illustrate the process of structuring the data using Python data structures.
• Identify the need to work with various documents like Excel, PDF, Word and others.
• Demonstrate the use of built-in functions to navigate the file system.
• Implement the object-oriented programming concepts in Python.

Module-1	**(8 hrs)**
Python Basics: The Python Programming Language, Program, Debugging, Variables, Expressions, and Statements, Entering Expressions into the Interactive Shell, 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 Module-s, Ending a Program Early with sys.exit(). **Text Book 1: Chapters 1 – 2 Text Book 2: Chapters 1.1-1.6, Chapter 2.1-2.10**	
Module-2	**(8 hrs)**
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. **Text Book 1: Chapters 3, 4**	
Module-3	**(8 hrs)**
Dictionaries and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things. **Manipulating Strings:** Working with Strings, Useful String Methods, Project: Password Locker. **Text Book 1: Chapters 5, 6**	
Module-4	**(8 hrs)**
Reading and Writing Files: Files and File Paths, The os.path Module-, The File Reading/Writing Process, Saving Variables with the shelve Module-, Saving Variables with the print.format() Function, Project: Generating Random Quiz Files. **Working with Excel Spreadsheets:** Installing the openpyxl Module-, Reading Excel Documents, Writing Excel Documents. **Text Book 1: Chapters 8, 9 and 12**	

Module-5	(8 hrs)
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. **Text Book 2: Chapters 15 – 17**	

Practical Components

Sl. No.	Programs for Implementation:
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 percentages with suitable messages. 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 a Fibonacci sequence of length (N). Read N from the console. b) Write a function to calculate the factorial of a number. Develop a program to compute the 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 the 10 most frequently appearing words in a text file. [Hint: Use a dictionary with distinct words and their frequency of occurrences. Sort the dictionary in reverse order of frequency and display the dictionary slice of the 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.	Implement a Python program to perform the following operations on an Excel spreadsheet: a. Reading the first 5 rows of all columns b. Appending a new row / new column c. Delete row/column d. To perform aggregate functions
8.	Write a function named DivExp which takes TWO parameters a, b and returns the value of c, where c=a/b. Write a suitable assertion for a>0 in the function DivExp and raise an exception for when b=0. Develop a suitable program that reads two values from the console and calls a function DivExp.
9.	Define a function that 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 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 scorecard 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 get Marks() method to read marks into the list, and display() method

	to display the scorecard details.

Text Books:	
1.	"Automate the Boring Stuff with Python", Al Sweigart, No Starch Press, 1st Edition, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/).(Chapters 1 to 12, except 7th Chapter) for lambda functions, use this link: https://www.w3schools.com/python/python_lambda.asp
2.	"Think Python: How to Think Like a Computer Scientist", Allen B. Downey, Green Tea Press, 2nd Edition, 2015. (Available under CC-BY-NC license at https://greenteapress.com/thinkpython2/thinkpython2.pdf (Chapters 15-17)
Reference Books:	
1.	"Python for Everybody: Exploring Data Using Python 3", Charles R. Severance, CreateSpace Independent Publishing Platform, 3rd Edition, 2016. http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf

Course Outcome Statements: After the successful completion of the course, the student will be able to:	
CO1	Apply the fundamental programming constructs to solve basic computational problems.
CO2	Analyze the flow of execution and the need for debugging to rectify program bugs proficiently.
CO3	Utilize the methods to create and manipulate the importance of data structures in problem-solving.
CO4	Develop structured and modular code using appropriate Object-Oriented Programming paradigms.
CO5	Demonstrate the acquired skills to provide solutions to real-world problems.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	-	2	-	-	-	2	-	-	1
CO2	2	2	2	-	2	-	-	-	2	-	-	1
CO3	2	2	2	-	2	-	-	-	2	-	-	2
CO4	2	2	2	-	2	1	-	-	2	-	-	2
CO5	2	2	2	1	2	1	-	1	2	2	1	2
Avg.	**2**	**2**	**2**	**1**	**2**	**1**	-	**1**	**2**	**2**	**1**	**1.6**

3: High, 2: Medium, 1: Low