Program	Bachelor of Technology (BTech)	**Semester - 4**
Type of Course	Professional Core	
Prerequisite		
Course Objective	Python is a modern language for writing compact codes specifically for programming in the area of Server-side Web Development, Data Analytics, AI and scientific computing, production tools, and game programming. This course covers the basics of python programming to harness its potential for modern computing requirements.	

Teaching Scheme (Contact Hours)				**Examination Scheme**				
Lecture	**Tutorial**	**Practical**	**Credit**	**Theory Marks**		**Practical Marks**		**Total Marks**
				SEE	**CIA**	**SEE**	**CIA**	
3	0	2	4	70	30	25	25	150

***SEE** - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	**Topics**	**T**	**W**
1	**Introduction to Python** Introduction to python, Advantages of python, Installing python, Program Structures, User interface and IDE, Indentation, Comments, Reading Input, Print Output **Python Object and Data Structure** Python Data types, Tokens, Variables, Numbers, Boolean, String, Type Conversions, Concept of Mutability, Indexing and Slicing of String, formatting string, String functions ,List, Set, Tuple, Dictionary	**11**	**20**
2	**Python Operators** Arithmetic, Assignment, Comparison, Logical, Identity, Membership and Bitwise Operator. **Conditional and Looping Statements** If statement, if else statement, if elif, else statement, nested if statement, For loop statement, While loop statement, break, continue, pass keywords. **Functions in Python** Creating function, DOCSTRING, Types of arguments, Calling function, return statement, Lambda expression, Recursion	**11**	**20**
3	**File IO in Python** Open a file, Read a file, Write a file Exception handling Errors and Exceptions, Handling Exceptions, Raising Exceptions, try with else, try with finally	**8**	**20**
4	**Modules** Importing a module, Math module, random module, datetime module, creating custom module **Matplotlib** Graph, Plot, Drawing Multiple Lines and Plots, Export graphs/plots to Image/PDF/SVG, Axis, Ticks, Grids, Line Appearance, Labels, Annotations, Legends, Types of Charts	**8**	**20**
5	**Object Oriented Programming with Python**	**7**	**20**

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
	Object Oriented Approach, Custom Classes: Attributes and Methods, Inheritance, Polymorphism, Custom Collection Classes		
	Total	**45**	**100**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	10	75	15	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**explain** basic features, data types and data structures in python.
CO2	**demonstrate** operators, conditional statement, looping statement and functions.
CO3	**implement** th file management programs and Exception handling.
CO4	**experiment** with modules and matplotlib.
CO5	**use** object-oriented programming approach with python.

Reference Books

1. **Programming in Python 3 : A Complete Introduction to the Python Language**
   By Mark Summerfield
2. **Introduction to Computation and Programming Using Python**
   By John V Guttag | Prentice Hall of India
3. **Core Python Programming**
   By R. Nageswara Rao | Dreamtech Press

List of Practical

1. Perform python programs on basic maths formulas.

1. WAP to print “Hello World”
2. WAP to print your address i) using single print ii) using multiple print
3. WAP to print addition of 2 numbers (without input function)
4. WAP to calculate and print average of 2 numbers (without input function)
5. WAp to add two number entered by user.
6. WAP to calculate area of circle.
7. Purposefully raise Indentation Error and Correct it.
8. WAP to calculate simple interest.
9. WAP Calculate Area and Circumfrence of Circle
10. WAP to print Multiplication table of given number.
11. WAP to calculate Area of Triangle (hint: a = h*b*0.5)
12. WAP to convert degree to Fahrenheit and vice versa.
13. WAP to calculate total marks and Percentage.

2. Perform python programs on if else statements.

1. WAP to check whether the given number is positive or negative.
2. WAP to check whether the given number is odd or even
3. WAP to find out largest number from given two numbers using simple if and ternary operator.

4. WAP to find out largest number from given three numbers.
5. WAP to check whether the given year is leap year or not. [If a year can be divisible by 4 but not divisible by 100 then it is leap year but if it is divisible by 400 then it is leap year]
6. WAP in python to display the name of the day according to the number given by the user
7. WAP to implement simple calculator which performs (add,sub,mul,div) of two numbers based on user input.
8. WAP to calculate electricity bill based on following criteria. Which takes the unit from the user.
    i. First 1 to 50 units – Rs. 2.60/unit
    ii. Next 50 to 100 units – Rs. 3.25/unit
    iii. Next 100 to 200 units – Rs. 5.26/unit
    iv. above 200 units – Rs. 8.45/unit

3. **Perform python programs on For and while loop.**

1. WAP to print 1 to 10
2. WAP to print 1 to n
3. WAP to print odd numbers between 1 to n
4. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3
5. WAP to print sum of 1 to n numbers
6. WAP to print sum of series 1 + 4 + 9 + 16 + 25 + 36 + ...n
7. WAP to print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 ... n
8. WAP to print multiplication table of given number.
9. WAP to find factorial of the given number
10. WAP to find factors of the given number
11. WAP to find whether the given number is prime or not.
12. WAP to print sum of digits of given number
13. WAP to check whether the given number is palindrome or not

4. **Perform python programs on String manipulation.**

1. WAP to check given string is palindrome or not.
2. WAP to reverse the words in given string.
3. WAP to remove ith character from given string.
4. WAP to find length of String without using len function.
5. WAP to print even length word in string.
6. WAP to count numbers of vowels in given string.
7. WAP to convert given array to string.

5. **Perform python programs on List operations.**

1. WAP to find sum of all the elements in List.
2. WAP to find largest element in a List.
3. WAP to split the List into two and append the first part to the end.
4. WAP to interchange first and last elements in list entered by a ser.
5. WAP to interchange the elements on two positions entered by a user.
6. WAP to reverses the list entered by user.
7. WAP to print all even number in list entered by user.

6. **Perform python programs on Tuples, dictionary and set operation.**

1. WAP to sort python dictionary by key or value.
2. WAP to merge two dictionaries given by user.
3. WAP to find tuples that have all elements divisible by K from a list of tuples.
4. WAP to find Tuples with positive elements in List of tuples.
5. WAP which perform union of two sets.

7. **Perform python programs on functions.**

1. WAP to count simple interest using function
2. WAP that defines a function to add first n numbers
3. WAP to find maximum number from given two numbers using function

	4. WAP that defines a function which returns 1 if the number is prime otherwise return 0 5. WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...) 6. WAP to find the factorial of a given number using recursion. 7. WAP to implement simple calculator using lamda function.
8.	**Perform python programs on File handling** 1. WAP to read entire file named abc.txt 2. WAP to print program it self on console. 3. WAP to read first 5 lines from the file named abc.txt 4. WAP to find the longest word in a file named abc.txt 5. WAP to find the size of the file named abc.txt 6. WAP to implement search function to search specific occurance of word in a given text file.
9.	**Perform python programs on Exception handling** 1. WAP to handle divide by zero exception. 2. WAP to raise your custom Exception. 3. WAP to handle file not found Exception. 4. WAP to handle type Exception. 5. WAP to demonstrate valueError and indexError with example.
10.	**Perform python programs on Modules** 1. WAP to create Calculator module which defines functions like add, sub,mul and div. create another file that uses the Calculator module. 2. WAP to Pick a random character from a given String 3. WAP to Pick a random element from a given list. 4. WAP to demonstrate the use of the math module. 5. WAP to demonstrate the use of date time module.
11.	**Perform python programs on Graphs** 1. WAP to demonstrate the use of Pie chart. 2. WAP to to Plot List random of X, Y Coordinates in Matplotlib. 3. WAP to demonstrate the use of Bar chart. 4. WAP to demonstrate the use of Histogram. 5. WAP to display the value of each bar in a bar chart using Matplotlib. 6. WAP create a Scatter Plot with several colors in Matplotlib? 7. WAP to Display an Image in Grayscale in Matplotlib.
12.	**Perform python programs on Object Oriented Programming** 1. Write a Program to create a class by name Students, and initialize attributes like name, age, and grade while creating an object. 2. Create a class named Bank_Account with Account_No, User_Name, Email,Account_Type and Account_Balance data members. Also create a method GetAccountDetails() and DisplayAccountDetails(). Create main method to demonstrate the Bank_Account class. 3. WAP to create Circle class with area and perimeter function to find area and perimeter of circle. 4. Define Time class with hour and minute as data member. Also define addition method to add two time objects. 5. WAP to demonstrate inheritance in python.