Programming in Python

Course Lead: Ms. Vaishali Gupta

Course Co-coordinator: Dr. Balbindar Kaur

Scheme:

<table>
<tr><td>Course Title</td><td colspan="3">Programming in Python</td><td colspan="3">Course Type</td><td colspan="3">Comprehensive</td></tr>
<tr><td>Course Code</td><td colspan="3"></td><td colspan="3">Class</td><td colspan="3">BTech</td></tr>
<tr><td rowspan="6">Instruction Delivery</td><td>Activity</td><td>Credits</td><td>Credit Hours</td><td colspan="4" rowspan="2">Total Number of Classes per Semester</td><td colspan="2" rowspan="2">Assessment in Weightage</td></tr>
<tr><td>Lecture</td><td>3</td><td>3</td></tr>
<tr><td>Tutorial</td><td>0</td><td>0</td><td rowspan="3">Theory</td><td rowspan="3">Tutorial</td><td rowspan="3">Practical</td><td rowspan="3">Self-study</td><td rowspan="3">CIE</td><td rowspan="3">SEE</td></tr>
<tr><td>Practical</td><td>1</td><td>2</td></tr>
<tr><td>Self-study</td><td>1</td><td>3</td></tr>
<tr><td>Total</td><td>5</td><td>8</td><td>45</td><td></td><td>15</td><td></td><td>50%</td><td>50%</td></tr>
<tr><td rowspan="4">Names Course Instructors</td><td>Course Lead</td><td colspan="8">Ms. Vaishali Gupta</td></tr>
<tr><td colspan="3">Theory</td><td></td><td colspan="4">Practice</td></tr>
<tr><td colspan="3"></td><td></td><td colspan="4"></td></tr>
</table>

Course Plan

Name of The Course	Programming in Python	L	T	P	S	C
Course Code		3	1	1	1	5

Course Overview

This course deals with the python programming concept. In this course, the students will be able to learn to write programs in Python using the loops, conditional statements etc. Upon completion of this course, the student will be able to write non trivial python programs dealing with a wide variety of subject matter domains.

Course Objectives

- To impart the basic concepts and constructs of Python programming.
- To understand Lists, tuples, Dictionaries, and Regular Expressions in Python.
- To Apply the concepts of file I/O in Python.
- To Implement the object-oriented programming concepts in Python.
- To do Explanatory data analysis-using Python packages.

Course Outcomes

COs No	Course Outcomes (CO)
On completion of the course, the students shall be able to	
CO1	Understand the basic concepts and constructs of Python programming.
CO2	Apply python programming concept on collection of frame work such as: Lists, tuples, Dictionaries, and Regular Expressions.
CO3	Implement the file handling concepts of file I/O in Python.
CO4	Develop the object-oriented programming concepts in Python.
CO5	Create real life applications using python

CO No.	Bloom's Taxonomy Level (BTL)					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
CO1		K2				
CO2			K3			
CO3				K4		
CO4			K3			
CO5						K6

Program Outcomes:

PO1	**Computing Science Knowledge:** Apply the knowledge of mathematics, statistics, computing science and information science fundamentals to the solution of complex computer application problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex computing science problems reaching substantiated conclusionsusing first principles of mathematics, natural sciences, and computer sciences.
PO3	**Design/development of solutions:** Design solutions for complex computing problems and design system components or processes that meet the specifiedneeds with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysisand interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques,resources, and modern computing science and IT tools including prediction and modeling to complex computing activities with an understanding of the limitations.
PO6	**IT specialist and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional computing science and information science practice.
PO7	**Environment and sustainability:** Understand the impact of the professional computing science solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the computing science practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the IT analyst community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the computing science and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**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.

CO-PO Mapping:

CO/PO Mapping (1 / 2 / 3 indicates strength of correlation) 3 - Strong, 2 - Medium, 1 – Low														
COs	**Programme Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2		3						1			1		
CO2	2	2	3	2	2				1		1		1	1
CO3	2	2	3	2	2				1			1	1	
CO4	1	2	1						3		3	2		2
CO5							1		2			3		

Course Assessment:

AssessmentTools	**CIE**							**TOTAL CIE Marks**	**SEE**
	QUIZ1 / AAT 1	**CAT 1**	**QUIZ2 /AAT2**	**CAT 2**	**LAB**	**LAB Exam**	**Course Based Project**		
Comprehensive		A1		A2	A3	A4			
	0	30	0	30	20	0	20	100	100

Course Based Project Rubric:

AssessmentTools	**CIE**					
	PPP	**TS1**	**TS2**	**VIVA**	**Total**	
Course Based Project		A1		A2	A3	
	5	5	5	5	20	

Course Contents

Theory Contents
Introduction to Python, Unique features of Python, Python versions, Install Python and Environment Setup, First Python Program. Python Identifiers, Keywords and Indentation, Python Data Types, Variables, Operators in Python – Assignment, Logical, Arithmetic etc. Taking User Input (Console), Conditional Statements – If else and Nested If else and elif. List, Tuple, Sets and Dictionary, Understanding Iterators, Generators, Comprehensions Loops in Python – For Loop, While Loop & Nested Loops, String Manipulation – Basic Operations, Slicing & Functions and Methods, User Defined Functions, Lambda Function, Importing Modules – Math Module. Reading and writing text files writing Text Files Appending to Files and Challenge Writing Binary Files Manually Using Pickle to Write Binary Files. Regular expressions, the match Function, The search Function, Matching vs searching, Search and Replace, Extended Regular Expressions, Wildcard. Basics of Object Oriented Programming, Creating Class and Object, Constructors in Python – Parameterized and Non-parameterized, Inheritance in Python, In built class methods and attributes, Multi-Level and Multiple Inheritance, Method Overriding and Data Abstraction, Encapsulation and Polymorphism. Numpy-Introduction, Creating arrays, Using arrays and Scalars, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input and Output. Pandas: Introduction, uses of Panda, Series in pandas, Index objects, Reindex, Drop Entry, Selecting Entries, Data Alignment, Rank and Sort Summary Statics, Missing Data, Index Hierarchy. Matplotlib: Introduction, uses of Matplotlib, Data Visualization. Scikit-Learn: Introduction, Predictive Analysis, and Introduction to Machine Learning in Python.

Practical Contents
Python Programming Languages Lab interprets the use of procedural statements like assignments, conditional statements, loops and function calls. Infer the supported data structures like lists, dictionaries and tuples in Python. Also describe the need for Object-oriented programming concepts in Python, and the use of standard libraries in python.

LESSON PLAN for THEORY COURSES (15 weeks * 3 hours = 45 Classes)

Session No	Topics	Skills to be achieved.
1.	Introduction to python- an integrated high level language, Unique Features of Python Programming Language	Students are aware about Basics of python language, number system. Variables, expression, predefine functions, use of operators**.**
2.	Python versions, Install Python and Environment Setup, First Python Program	
3.	Python Identifiers, Keywords and Indentation	
4.	Python Data Types, Variables, Expressions and Statements	
5.	Operators in Python – Assignment, Logical, Arithmetic etc. Taking User Input (Console).	
6.	Conditional Statements – If else statement its working and execution.	
7.	Nested-if statement and Elif statement in Python, Expression	
8.	Python Collections: List - Accessing values, Updating, Delete elements.	Students are aware about
9.	Indexing and Slicing, List - Built function and methods	
10.	List operations – joining, slicing, + , * , in , not in, List functions and methods - len( ), insert( ), append( ), extend( ), sort( ), remove( ), reverse( ), pop( ), list( ), count( ), extend( ), index( ), cmp( ), max( ), min( ).	
11.	Tuple - Accessing values, Updating, Delete elements, Tuples: Immutable concept, creating, initialising and accessing elements in a tuple.	
12.	Basic Tuple operations, Indexing and slicing. Tuple - Built function and methods.	
13.	Tuple assignment, Tuple slices, Tuple indexing Tuple Functions- cmp(), len(), max(), min(), tuple(), index(), count(), sum(), any(), all(), sorted(), reversed()	

14.	Sets: Create a set, set operation, set built in function, Opening and reading files: Open text file-opening a text file-syntax	Python Collections (Arrays)- List, Tuple, Sets and Dictionary,
15.	Built in Dictionary Functions and Methods, Function: Definition, Calling, Pass by reference vs value.	
16.	Dictionaries: Concept of key-value pair, creating, initializing and accessing the elements in a dictionary.	
17.	Dictionaries: Traversing, appending updating and deleting elements.	
18.	Dictionary Functions and methods: cmp( ), len( ), clear( ), get( ), has_key( ), items( ), key( ), update( ), values( ), pop( ), fromkeys( ), dict( ).	loops in python, string manipulation user defined function, modules
19.	Understanding Iterators, Generators	
20.	Comprehensions Loops in Python: Purpose and working of loops, While loop including its working, For Loop, Nested Loops.	
21.	Random number function: Choice(), Random range(), Random(), Seed(), Shuffle(), Uniform()	
22.	Strings Manipulation: Basic Operations, Slicing & Functions and Methods.	
23.	Strings: Special operators, formatting operator/Operations of Strings	
24.	String operators: +, *, in, not in, range slice [n:m], Comparing strings using relational operators	
25.	String functions & methods: len, capitalize, find, isalnum, isalpha, isdigit, lower, islower, isupper, upper, lstrip, rstrip, isspace, istitile, partition, replace, join.	
26.	String functions & methods: split, count, decode, encode, swapcase,	
27.	String constants, Regular Expressions and Pattern Matching , User Defined Functions	

28.	Variable-length, Lambda function, Filter, Map and Reduce, Recursion	Students are aware about Python File Handling, Regular Expression.
29.	Python File Handling, Regular Expression: Reading and writing text files.	
30.	Appending to Files and Challenge Writing Binary Files Manually Using Pickle to Write Binary Files	
31.	Regular expressions, the match Function, The search Function	
32.	Matching vs searching, Search and Replace, Extended Regular Expressions, Wildcard.	
33.	Different modes of opening a file-The file object attributes-Reading from text file	
34.	Modules : Introduction , Importing Modules - Math Module.	
35.	Object Oriented Programming in Python: Basics of Object Oriented Programming, Creating Class and Object.	Students are aware about Object Oriented Programming in Python, and Inheritance in Python
36.	Class definition, methods, variables and other operations in the classes. Classes Special Methods: _init_, _str_ , comparison methods and Arithmetic methods etc.	
37.	Constructors in Python – Parameterized and Non-parameterized.	
38.	Inheritance in Python, In built class methods and attributes.	
39.	Multi-Level and Multiple Inheritance	
40.	Method Overriding and Data Abstraction, Encapsulation and Polymorphism.	
41.	Libraries in Python, Working of Python Library, Use of Libraries in Python Program. Python standard library: TensorFlow, Matplotlib, Pandas, Numpy, Keras, SciPy, Scrapy, Scikit-learn, PyGame, PyTorch, PyBrain , Pandas Vs NumPy	Students are aware about

42.	Numpy-Introduction, Creating arrays, Using arrays and Scalars, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input and Output.	Libraries in python: Numpy, Reindex, Matplotlib, Scikit.
43.	Pandas: Introduction, uses of Panda, Series in pandas, Index objects, Reindex, Drop Entry, Selecting Entries, Data Alignment, Rank and Sort Summary Statics, Missing Data, Index Hierarchy.	
44.	Matplotlib: Introduction, uses of Matplotlib, Data Visualization. Scikit-Learn: Introduction, Predictive Analysis, and Introduction to Machine Learning in Python.	
45.	Case study of Mini Applications on Python	

LESSON PLAN for PRACTICAL (30 Lab Classes)

Sl No.	Topics	Skills
1.	Write a Python program to print the documents (syntax, description etc.) of Python built-in Functions.	Students are aware of the python built-in function to write a program
2.	Write a Python Program for factorial of a number	
3.	Write a Python program which accepts the radius of a circle from the user and compute the area	Students are aware of the numbering concepts to write a python program
4.	Write a Python program to print the calendar of a given month and year.	
5.	Write a Python Program for simple interest, and compound interest	
6.	Write a Python program to calculate the length of a string.	Students are aware of the concept of string to write a python program
7.	Write a Python program to split and join a string	
8.	Write a Python program to demonstrate various ways of accessing the string- By using Indexing (Both Positive and Negative)	
9.	Write a Python program to multiplies all the items in a list.	Students are aware of the concept of list to write a python program
10.	Write a Python program to Reverse a linked list	
11.	Write a Python program to find smallest number in a list, and to find largest number in a list	
12.	Write Python program to perform following operations on Dictionaries:	

	c) Update Dictionary d) Delete Set e) Looping through Dictionary	Students are aware of the concept of dictionary to write a python program
13.	Write a Python script to sort (ascending and descending) a dictionary by value.	
14.	Create a dictionary and apply the following Methods: 1) Print the dictionary items 2) access items 3) use get() 4)change values 5) use len()	
15.	Write a Python program to create a tuple with different data types.: 1) Add items 2) len()	Students are aware of the concept of tuple to write a python program
16.	Write a Python program to create a tuple and perform the following methods: 1) check for item in tuple 2)Access items	
17.	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.	Students are aware of the concept of importing modules to write a python program
18.	Write a program to double a given number and add two numbers using lambda()?	
19.	Write a python program to define a module and import a specific function in that module to another program.	
20.	Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.	Students are aware of the concept to input text file to write a python program
21.	(i) Write a python program to open and write “hello world” into a file? (ii) Write a python program to write the content “hi python programming” for the existing file.	
22.	Write a Python class to convert an integer to a roman numeral.	Students are aware of the concept of Class to write a python program
23.	(i) Write a python Program to display welcome to MRCET by using classes and objects. (ii) Write a python Program to call data member and function using classes and objects. (iii) Write a program to find sum of two numbers using class and methods. (iv) Write a program to read 3 subject marks and display pass or failed using class and object.	
24.	Write a python program using polymorphism, inheritance concepts	Students are aware of the concept of OOPs to write a python program.
25.	Implement single and Multiple inheritance concept	

26.	Implement data-abstraction in python, try to create object of abstract class.	
27.	Write Python program for addition of two matrix using numpy library and find transpose of matrix.	Students are aware of the standard libraries in python.
28.	Using a numpy module create an array and check the following: 1. Type of array 2. Axes of array 3. Shape of array 4. Type of elements in array	
29.	Write Python program using panda library for adding new column admission no. to data frame. Initially, there are 3 column respectively name, marks and have some dummy data for each columns.	
30.	Use Matplotlib to write python program for draw a graph of two lines intersecting each other while points for line1 at x axis- (1,2,3) and at y-axis at (2,4,1) and for line2 points are- (1,2,3) at x axis and (4,1,3) at y-axis.	

Text Books

1. Mark Lutz," Programming Python", Prentice Hall India, 7th Edition, 2017

2. Python Programming- A Problem solving approach" by Reema Thareja

3. Allen Downey, "Think Python", O'Reilly Media, 1st Edition, 2012

Reference Books

1. Mark Lutz, "Learning Python", McGraw-Hill publication, 2nd Edition, 2010

2. Luciano Ramalho, "Fluent Python", O'Reilly Media, 1st Edition, 2015

3. Brett Slatkin , "Effective Python: 59 Specific Ways to Write Better Python", Pearson Education, Inc, 1st Edition 2015.

SWAYAM/NPTEL/MOOCs :

https://onlinecourses.swayam2.ac.in/cec22_cs20/preview

https://onlinecourses.nptel.ac.in/noc22_cs32/preview

Web Link:

https://docs.python.org/3/tutorial/

PROBLEM-BASED LEARNING: Exercises in Problem-based Learning (Assignments)

S No	Problems
1.	Write a Python program to display the current date and time.
2.	Write a Python program that accepts a sequence of comma-separated numbers from the user and generates a list and a tuple of those numbers
3.	Find the length of the list and simply swap the first element with (n-1)th element.
4.	Given a list in Python and provided the positions of the elements, write a program to swap the two elements in the list.
5.	Write a python program to perform arithmetic, assignment, logical and comparison operators? B) Write a Python program to add two positive integers without using the '+' operator. (use bitwise operator) C) Write a Python program to perform the basic four operators (+, -, *, /)
6.	Accept one integer and one float number from the user and calculate the multiplication of both the numbers.
7.	Python program for file operations such as opening a file, reading from it, writing into it, closing it, renaming a file, deleting a file, and various file methods.
8.	Write a Python program to create a histogram from a given list of integers
9.	Write a Python program to print all even numbers from a given list of numbers in the same order and stop printing any after 237 in the sequence.
10.	Write a Python program that will accept the base and height of a triangle and compute its area
11.	Write a Python program to get the current username
12.	Write a Python program to print without a newline or space.
13.	(i) Write a Python program to demonstrate working # of len(). (ii) Python code to find the length # of list using enumerate function.
14.	(i) Sorting elements in ascending order of the integer value given to them. (ii) Sorting objects on the basis of the string value a variable holds
15.	(i) Write a Python program to find files and skip directories in a given directory. (ii) Write a Python program to extract a single key-value pair from a dictionary into variables. (iii) Write a Python program to convert true to 1 and false to 0
16.	(i) Write a python program to convert decimal to hexadecimal. (ii) Write a Python program to check if every consecutive sequence of zeroes is followed by a consecutive sequence of ones of same length in a given string. Return True/False
17.	(i) Write a Python function to check whether a number is divisible by another number. Accept two integer values from the use.

	(ii) Write a Python function to find the maximum and minimum numbers from a sequence of numbers.

18.	(i) What will be the output of the following Python code?
	`n = [-2, -4]`
	`m = map(lambda x:x*2, n)`
	`print(m)`
	
	(ii) How many keyword arguments can be passed to a function in a single function call?

19.	Write a program that implements method overriding following the below steps: Define class Animal Use constructor to set the name with a default value = "This Animal" Define a method eat with a parameter food with a default value = "Grass" Inside the method print (self.name, " eats", food) Define a class Mammal, inherit from Animal Inside the class, override eat method to print(self.name, " does not eat. It only drinks") Define class WingedAnimal, inherit from Animal Override eat method to print(self.name," eats anything and everything") Define a class called Bat, inherit from WingedAnimal, Mammal Define method smell, which prints "This Animal Stinks" Define a class called FruitBat, inherit from Mammal, WingedAnimal (Notice the Order) rabbit1 = Animal("Rabbit") print("Rabbit1 is an instance of Animal")
20.	Write a simple example using this *: Follow the given instructions while writing the program Use the Module_Imp3 which contains functions that can be imported. Use from Module_Imp3 import * Take an integer as input from user and store it in the variable side. Call the function calculatearea(side,side) Call the function calculatediameter(side) Call the function pivalue() print shapes[1:2]

COURSE-BASED PROJECT (Psychomotor skills):

Sl No	Suggested Projects
1.	Create a code generator.
2.	Build a countdown calculator.

3.	Write a sorting method.
4.	Build an interactive quiz.
5.	Tic-Tac-Toe by Text.
6.	Make a temperature/measurement converter.
7.	Build a counter app.
8.	Build a number-guessing game.
9.	Build an alarm clock.
10.	Analyze Your Own Netflix Data
11.	Build Password generator
12.	Use Tweepy to create a Twitter bot
13.	Build an Address Book
14.	Create a Crypto App with Python
15.	Build an upgraded code generator.
16.	Make your Tic-Tac-Toe game clickable.
17.	Scrape some data to analyze.
18.	Build a clock website.
19.	Automate some of your job.
20.	Automate your personal habits.
21.	Create a simple web browser.
22.	Write a notes app.
23.	Build a typing tester.
24.	Create a “site updated” notification system.
25.	Recreate your favorite board game in Python.
26.	Build a Wikipedia explorer.
27.	Build Text-based Adventure Game
28.	Dice Rolling Simulator
29.	Build Email Slicer
30.	Build Countdown Timers
31.	Build a Word Guessing Game
32.	Create YouTube video downloader
33.	Make Voice Assistant Advanced Level
34.	Build a stock market prediction app.
35.	Build a chatbot.
36.	Program a robot.
37.	Build an image recognition app.
38.	Make a price prediction model.
39.	Create your own sentiment analysis model.
40.	Create an interactive map.

Link for more project ideas:

1. https://data-flair.training/blogs/python-project-ideas/
2. https://realpython.com/tutorials/projects/