PUCIT

PYTHON PROGRAMMING

Subject Code: STPY201

Total Hours: 40 Credits: 4

Course Learning Objectives (CLO)

The objective of this course is to make students to

- Master the basic programming constructs (branches, loops, functions).
- Work with control structures such as branches, loops and functions.
- Understand how string and classes support the sequence methods in Python.
- Explore advanced topics such as inheritance, exceptions.
- Understand the key data structures and how they are used to represent sparse matrices.

UNIT 1: Python Basics, Branch, Loops and Strings [12 hours]

Programming using Python, Basic input and output, Errors, Development environment, Variables and assignments, Identifiers, Objects, Numeric types: Floating-point, Arithmetic expressions, Python expressions, Module basics, Math module, representing text, String basics, List and Set as basics, Common data types summary, Type conversions, Binary numbers, String formatting, Using the shell as a calculator, If-else statement, Equality and relational operators, Boolean operators and expressions, Order of evaluation, Membership and identity operators, Code blocks and indentation, Conditional expressions Loops, while loops, Counting, For loops, Counting using the range () function, while vs. for loops, Nested loops, Break and continue, String slicing, Advanced string formatting, String methods, Splitting and joining strings, the string format method, regular expressions.

UNIT 2: Data Structures, Functions and Recursion [08 hours]

Lists, Dictionaries, Mutability and References, Tuples, Sets and Collections, User-defined function basics, returning values from functions, Reasons for defining functions, Function stubs, Functions: Common errors, Function arguments, First class functions, anonymous functions, map, filter and reduce, list comprehensions, cartesian product, itertools, eval, exec, enumerate, zip, copy, Recursive functions, Recursive algorithm: Search, adding output statements for debugging, creating a recursive function, Recursive math functions, Recursive exploration of all possibilities.

UNIT 3: Classes and Exceptions **[12 hours]**

Classes: Grouping data, Class methods, Class and instance object types, Class constructors, Class interfaces, Class customization, Classes as numeric types, Memory allocation and garbage collection, Decorators, Generators, Handling exceptions using try and except, Multiple exception handlers, raising exceptions, Exceptions with functions, using finally to clean up, Custom exception types.

UNIT 4: Modules, Files and Inheritance **[08 hours]**

Modules, finding modules, importing specific names from a module, executing modules as scripts, Reloading modules, Packages and Standard library, reading files, Writing files, Interacting with file systems, Binary data, Command-line arguments and files, the 'with' statement, Comma separated values files, Derived classes, accessing base class attributes, overriding class methods, Is-a versus has-a relationships, Mixing classes and multiple inheritance, Data and Times, Sound, Getting files from the Internet.

SKILL BASED EXERCISE (SBE):

Note: - These Projects/activities are only indicative; the faculty member can innovate

Case studies: -

Python use cases in Finance

(https://www.botreetechnologies.com/blog/top-10-python-use-cases-and-applications/)

Python use cases for automation

(https://www.botreetechnologies.com/blog/top-10-python-use-cases-and-applications/)

Assignment on: -

- Object Orientation & Data Structures,
- Recursion & Random Number Generation

Mini projects: -

- Tic tac Toe
  - Draw the Tic Tac Toe game board
  - Checking whether a game board has a winner
  - Handle a player move from user input

Contact Book: The main objective of this project is to generate a contact book using python where users can add a new contact, edit, or delete existing contacts and view the details of all their contacts

Labs Exercises: -

- Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list and makes a new list that has only the even elements of this list in it.
- Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game)

Remember the rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock
- Write a program that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order.
- Create a program that will play the "cows and bulls" game with the user.
- Write a program that creates
    - a student class that initializes attributes name, age and grade while creating an object
    - a teacher class that initializes parameters name and age while creating an object and inherits the attributes, from its parent class Staff, the attributes' role, department and salary
    - a cohort class consisting an array of students and provides methods for adding students to a cohort and removing students from a cohort
    - overload the + and > operators of the cohort class so that two cohorts can be merged and we can compare if one cohort is larger than the other.
    - Create a method in the cohort that returns students of a particular age
- Write a Python program to calculate the geometric sum of n-1
- Write a Python program to find the greatest common divisor (gcd) of two integers
- Finding the factorial of a number

Coding Challenge: - Problem Set

TEXTBOOKS:

- Brian Heinold, "A Practical Introduction to Python Programming", 2022

REFERENCES:

- Fluent Python – by Luciano Ramalh, O'Reilly Publications
- Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming
- Beginning Programming with Python for Dummies, 2ed by John Paul Mueller

Course Outcomes:

On completion of this course, students are able to

- Discuss both the principles and the practice of programming, using Python.
- Use iterative functions and loops in Python for managing and transforming data.
- Demonstrate the working of basic String operations
- Demonstrate the programming skill using class operations.
- Implement the writing and reading files in Python.