|  |  |
| --- | --- |
| **Introduction to Python Programming** | |
| **Course Code:PLC142**  **Pre-requisites: Nil**  **Course Coordinator:** | **Credits: 2:0:1**  **Contact Hours:28+14** |

### **Course Contents**

### **Unit I**

**The way of the program**: The Python programming language, what is a program? What is debugging? Syntax errors, Runtime errors, Semantic errors, Experimental debugging.

**Variables, Expressions and Statements**: Values and data types, Variables, Variable names and keywords, Conditional Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator. **Iteration**: Assignment, Updating variables, the for loop, the while statement, The Collatz 3n + 1 sequence, Nested Loops for Nested Data. **Functions:** Functions with arguments and return values, Lambda Functions

|  |  |
| --- | --- |
| * Pedagogy /   Course delivery tools | * Chalk and talk, Power point presentation, Videos |
| * Link |  <https://www.learnbyexample.org/python/>   <https://www.learnpython.org/>  https://pythontutor.com/visualize.html#mode=edit |

### **Unit II**

**Tuples**: Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.

**Lists**: List values, Accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, Cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices.

**Dictionarie**s: Dictionary operations, dictionary methods, aliasing and copying.

|  |  |
| --- | --- |
| * Pedagogy /   Course delivery tools | * Chalk and talk, Power point presentation, Videos |
| * Link |  <https://www.learnbyexample.org/python/>   <https://www.learnpython.org/>  https://pythontutor.com/visualize.html#mode=edit |

### **Unit III**

**Modules**: Random numbers, the time module, the math module, creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot Operator. **Files**: About files, writing our first file, Reading a file line-at-a-time, turning a file into a list of lines, Reading the whole file at once, working with binary files, Directories, fetching something from the web. Algorithms: Linear search, Binary search, merging two sorted lists.

|  |  |
| --- | --- |
| * Pedagogy /   Course delivery tools | * Chalk and talk, Power point presentation, Videos |
| * Link |  <https://www.learnbyexample.org/python/>   <https://www.learnpython.org/>  https://pythontutor.com/visualize.html#mode=edit |

### **Unit IV**

**Object oriented programming:** Classes and Objects — The Basics, Attributes, Adding methods to our class, Instances as arguments and parameters, Converting an instance to a string, Instances as return values, Objects are mutable, Sameness, Copying. **Inheritance:** Polymorphism, Generalization, Pure functions, Operator Overloading.

|  |  |
| --- | --- |
| * Pedagogy /   Course delivery tools | * Chalk and talk, Power point presentation, Videos |
| * Link |  <https://www.learnbyexample.org/python/>   <https://www.learnpython.org/>  https://pythontutor.com/visualize.html#mode=edit |

### **Unit V**

**Exceptions**: Catching exceptions, Raising our own exceptions, the finally clause of the try statement. **Strings**: Working with strings as single things, Working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, The in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method.

|  |  |
| --- | --- |
| * Pedagogy /   Course delivery tools | * Chalk and talk, Power point presentation, Videos |
| * Link |  <https://www.learnbyexample.org/python/>   <https://www.learnpython.org/>  https://pythontutor.com/visualize.html#mode=edit |

**Lab Component:**

|  |  |
| --- | --- |
| **SL. No.** | **QUESTIONS** |
|  | 1. Write a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide) 2. Write a python program to find the factorial of number using while loop. 3. Write a python program to add 10 numbers by inputting each from the keyboard using for loop. |
|  | 1. Write a python function linearSearch() to read an array and search for the key element. Display the appropriate messages. Use the recursive function. 2. Write a python program to define a function max\_of\_three() that takes three numbers as arguments and returns the largest of them using default arguments. 3. Write a python program to define a function generate\_n\_chars() that takes an integer n and a character c and returns a string, n characters long. For example, generate\_n\_chars(5,"x") should return the string "xxxxx“ using keyword only parameters. |
|  | 1. Write a python program to implement a stack and queue using lists 2. Write a python program to create a list of tuples having first element as the strings and the second element as the length of the string. Output the list of tuples sorted based on the length of the string. 3. Write a python program to create a list and perform the following operations  * Inserting an element * Removing an element * Appending an element * Displaying the length of the list * Popping an element * Clearing the list |
|  | 1. Write a tiny Python program numDict.py that makes a dictionary whose keys are the words ‘one’, ‘two’, ‘three’, and ‘four’, and whose corresponding values are the numerical equivalents, 1, 2, 3, and 4 (ints, not strings). 2. Write a Python program to store PROFILE\_DATA(user\_id, name, DOB, qualification, work\_experience) in a dictionary and pretty print the dictionary contents. (import pprint) |
|  | 1. Write a Python program to demonstrate built-in modules (Random,Time, Math, etc,) 2. Create a user defined module using python to execute the following a) area of circle b) area of triangle c) area of rectangle. |
|  | 1. Write a python program to create a text file and ask the user to enter 5-6 lines of text. Display the longest and the shortest word from the file. Display the length of these words. 2. Develop a python 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()]. |
|  | 1. Develop a python 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.] 2. Write a python program for the following: 3. Create a class called time. Its three members all type int should be called hours, minutes and seconds. Write a python program that prompts the user to enter a time values separately. The Program should then store the time in the object and finally printout the total no of seconds represented by this value. Use appropriate member functions. |
|  | 1. Write a python program to create a class called Mylist that shadows a python list: it should overload + operator to append the data to the list. Also provide constructor for your class that takes an existing list. 2. Write a python program to implement the following using Inheritance   Employee  \_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_  | |  Clerk Software Engineer  |  Team Leader |
|  | 1. Write a python program to Build a Number guessing game. When user enters not an integer raise the Exception and print total number of Guesses. 2. Write a python function named DivExp which takes TWO parameters a, b and returns a value c (c=a/b). Write suitable assertion for a>0 in function DivExp and raise an exception for when b=0. Develop a suitable program which reads two values from the console and calls a function DivExp. |
|  | 1. Write a python program to implement the following using strings   The third person singular verb form in English is distinguished by the suffix -s, which is added to the stem of the infinitive form: run -> runs. A simple set of rules can be given as follows:   * If the verb ends in y, remove it and add ies * If the verb ends in o, ch, s, sh, x or z, add es * By default just add s * Test your function with words like try, brush, run and fix. Tip: Check out the string method endswith().  1. Write a python function partition() that splits a list of soccer players into two groups. More precisely, it takes a list of first names (strings) as input and prints the names of those soccer players whose first name starts with a letter between and including A and M.   >>> partition(['Eleanor', 'Evelyn', 'Sammy', 'Owen', 'Gavin'])  Eleanor  Evelyn  Gavin  >>> partition(['Xena', 'Sammy', 'Owen']) |

**Course outcome:**

At the end of the course the student will be able to:

1. Demonstrate proficiency in handling loops and creation of functions
2. Identify the methods to create and manipulate lists, tuples and dictionaries.
3. Develop programs using modules and files.
4. Interpret the concepts of Object-Oriented Programming as used in Python.
5. Demonstrate the use of built-in functions for string processing and exception handling.

### Suggested Learning Resources

**Text Book:**

1. Downey, A., Elkner, J., & Meyers, C. (2012). How to think like a computer scientist: learning with python. Green Tea Press, Wellesley, Massachusetts.

**Reference Books:**

1. Al Sweigart, **“Automate the Boring Stuff with Python”,** 1stEdition, No Starch Press, 2015. (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/](http://www.learnbyexample.org/python-lambda-function/)

1. 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>

(Chapters 13, 15, 16, 17, 18) (Download pdf/html files from the above link)

**Course Assessment and Evaluation:**

|  |  |  |
| --- | --- | --- |
| **Continuous Internal Evaluation (CIE):** 50 Marks | | |
| **Assessment Tool** | **Marks** | **Course outcomes addressed** |
| Internal test-I | 30 | CO1, CO2 |
| Internal test-II | 30 | CO3, CO4, CO5 |
| Average of the two internal tests shall be taken for 30 marks. | | |
| **Other components** | | |
| Lab Component Evaluation | 20 | CO1, CO2, CO3, CO5, CO5 |
| **Semester-End Examination (SEE)** | 100 | CO1, CO2, CO3, CO4, CO5 |