19CS315 PROGRAMMING WITH PYTHON

Source:
https://www.houseofbots.com/images/news/4355/cover.png

Hours Per Week :

L	T	P	C
2	-	2	3

COURSE DESCRIPTION AND OBJECTIVES:

This course offers sufficient knowledge required to understand the fundamental concepts of Python programming language. In addition this course enable the students to use different data structures like lists, dictionaries, functions, exceptions and files. This course also enables the students to create reliable modular and reusable programming and to create applications using Object-Oriented Programming approach.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes
1	Analyse the usage of different data structures for practical and contemporary applications for a given problem.
2	Develop functional, reliable and user friendly Python programs for given problem statement and constraints.
3	Install the python environment and related packages that are required for practical and contemporary applications.
4	Design programs using the concepts of object oriented programming paradigm.
5	Create simple programming solutions to the given problems.

SKILLS:

- ✓ *Identify suitable data types and data structures required for an application.*
- ✓ *Design structured and Object oriented programming solutions.*
- ✓ *Design reliable applications for a given problem.*

UNIT– I **L- 9**

INTRODUCTION: History of Python, Features of Python, Python Installation on Windows & LINUX, Installing python packages via PIP, Running python commands using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation. Basic data types - Integers, Booleans etc.

OPERATORS AND EXPRESSIONS: Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations.

UNIT – II **L- 9**

CONTROL STRUCTURES: Conditional control structures - if, elif, else; Loop control structures - for, while, for... else, while..else, nested lo ops, break, continue, pass.

PYTHON DATA STRUCTURES: Lists, Tuples, Dictionary - creation, accessing, basic operators and methods.

UNIT – III **L- 9**

OTHER DATA STRUCTURES: Strings - creation, accessing, operators, methods; Sets - creation, accessing, operators, methods; List Comprehensions.

FUNCTIONS: Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions (lambda), Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

UNIT – IV **L- 9**

MODULES: Creating modules, import statement, from. Import statement, name spacing.

ERROR AND EXCEPTIONS: Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User defined exceptions.

FILE PROCESSING: Reading and Writing Files - Creating a New File - Writing to a File - Reading Files as Text, Opening and Closing files, reading and writing, tell (), seek(), rename ().

UNIT - V **L- 9**

OBJECT ORIENTED PROGRAMMING IN PYTHON: Classes, ‘self variable’ Methods, Constructor Method, Inheritance, Overriding Methods, Datahiding.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS **TOTAL HOURS: 30**

1. Python program to display all the python keywords and display each of them in separate lines.
2. Python program to find first n Prime Numbers.
3. Program that prints the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/n.
4. Program to find numbers that are divisible by 3 but are not by 6, between 1000 and 2000 (both included).
5. Program to print the list after removing delete even numbers in [5,6,77,45,22,12,24].
6. Program to generate a dictionary that contains (i, i*i), where i varies from 1 to n (both inclusive). Display the generated dictionary.
7. Program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.
8. Python code to achieve the following

   a) to remove vowels in the given string using control transfer statements.

   b) to count number of uppercase and lowercase letters in the given string.

   c) to remove all punctuation characters from given string.
9. Python code to illustrate the following

   a) Keyword arguments.

   b) Default arguments.

   c) Variable length arguments.
10. Function to find nth Fibonacci number.
11. Recursive function to find the factorial of a given number.
12. Functions to compute square of a number (use Lambda function).
13. Python code to handle the following built-in exceptions

    a) ValueError.

    b) ZeroDivisionError.

    c) TypeError.

    d) NameErrror.
14. Python code to handle multiple exceptions.
15. Python code to raise an exception.
16. Python code to read contents of a file and write the contents to another file.
17. Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.
18. Define a class named Rectangle which can be constructed by length and width. The Rectangle class has a method which can compute the area.

19. Define a class named Shape and its subclass Square. The Square class has an init function which takes length as an argument. Both classes have area function which can print the area of the shape where Shape’s area is 0 by default.

20. write assert statements to verify that every number in the list [2,4,6,8] is even.

21. Define a class Person and its two child classes: Male and Female. All classes have a method “getGender” which can print “Male” for Male class and “Female” for Female class

22. Write python code to depict the following oops concepts:

    a) Datahiding.

    b) Inheritance.

    c) Overriding.

TEXT BOOKS:

1. Vamsi Kurama, “Python Programming: A Modern Approach”, Pearson Publishers, 2018.
2. Mark Lutz, “Learning Python”, 5th edition, Orielly, 2013.

REFERENCE BOOKS:

1. Allen Downey, “Think Python”, Green Tea Press, 2016.
2. Ashok Namdev Kamthane and Amith Ashok Kamthane, “Programming and Problem Solving with Python”, 1st edition, McGraw Hill Education, 2016.
3. W.J. Chun, “Core Python Programming”, 3rd edition, Pearson Publishers, 2013.