****



U T T A R P R A D E S H

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **L** | **T** | **P/S** | **SW/FW** | **TOTAL CREDIT UNITS** | **No. of PSDA** |
| 2 | 0 | 2 | 2 | 4 | 5 |

**Course Title: PYTHON PROGRAMMING**

**Credit Units: 3**

**Course Code: CSIT232**

**Course Level:** UG

**Course Objectives:** This course aims at

* Provide in-depth knowledge of developing and debugging Python Programs.
* Illustrate and manipulate core data structures like Lists, Dictionaries, Tuples, and Strings.
* Understand the concept of files and exception handling.

**Pre-requisites:** Experience using a web browser and email.

**Course Contents/Syllabus:**

|  |  |
| --- | --- |
|  | **Weightage (%)** |
| **Module I: Introduction** | **15** |
| **Basic concepts:** Functional Programming, OOPS and Data Structures  **Getting Started:** Running Code in the Interactive Shell, Input, Processing and Output, Editing, Saving and Running a Script, Working of Python.  **Variables, Expressions and Statements:** Values and Data Types, Variables, Keywords, String Literals, Escape Sequences, Operators and Operands, Expressions and Statements, Interactive mode and Script mode, Order of Operations, Comments |
| **Module II: Conditional Statements and Loops** | **15** |
| Modulus Operator, Boolean Expressions, Logical Operators, Conditional Execution “if statement”, Alternative Execution “else clause”, Chained Conditionals “elif clause”, Nested Conditionals, while statement, For loop, Break and Continue Statement |
| **Module III: Functions and Recursion** | **25** |
| Function Calls, Type Conversion Functions, Math Functions, Composition, Adding new functions, Parameters and 25  Arguments, Stack Diagrams, Importing modules with “from”, Recursion, Stack Diagram for Recursive Functions,  Infinite Recursion  **String Functions:** Traversal, Comparison, Searching, Counting, Pre-defined String Functions, In Operator |
| **Module IV: Lists, Dictionaries and Tuples** | **25** |
| Lists: List as a Sequence, Traversing a list, List Operations, List Slices, List Methods, Map, filter and Reduce, Deleting Elements, Lists and Strings, Objects and Values, Aliasing, List Arguments Dictionaries: Dictionary as a set of counters, Looping and Dictionaries, Reverse Look Up, Dictionaries and Lists, Memos, Global Variables, Long Integers Tuples: Tuple Assignment, Tuples as return values, Variable Length argument tuples, Lists and Tuples, Dictionaries and Tuples, Comparing Tuples, Sequences of sequences |
| **Module V: Files** | **20** |
| Text files and their Formats, Reading from a file, Writing to a file, Accessing and Manipulating Files and Directories on the Disk, Format Operator, Filenames and paths Exception Handling: Errors, Exceptions, Handling Exceptions, Raising Exceptions, Try. Finally, The with Statement, Catching Exceptions, Databases, Pickling, Pipes |

**Course Learning Outcomes:**After completion of this course, student will be able to:  
• Apply basic Python programs.  
• Analyze basic Python decisions and iterations.  
• Create custom functions and call built-in Python functions.  
• Analyze data structures of Python.  
• Create exceptions and document code.

**Pedagogy for Course Delivery (Blended mix mode):** Subject will be taught based on classroom lectures, practical, self-works, and activities.

**Lab/Practical details:  
List of Experiments:**

1. Start the Python interpreter and type help(‘print’) to get information about the print statement.
2. If you run a 10-kilometer race in 43 minutes 30 seconds, calculate your average time per mile and your average speed in miles per hour using Python  
   Calculator. (Hint: there are 1.61 kilometers in a mile).
3. Write a function to calculate the square of first n natural numbers.
4. Write a function that draws a grid like the following:  
   + - - - -+ - - - -+  
   | | |  
   | | |  
   | | |  
   | | |  
   + - - - -+ - - - -+  
   | | |  
   | | |  
   | | |  
   | | |  
   + - - - -+ - - - -+
5. Write a function that takes four parameters—a, b, c and n—and then checks to see if Fermat’s theorem, an + bn = cn, holds. If n is greater than 2 and it  
   turns out to be true then the program should print, “Holy smokes, Fermat was wrong!” Otherwise the program should print, “No, that doesn’t work.”
6. Write a function that takes a string argument and returns true if it is a palindrome and False otherwise.
7. A number, a, is a power of b if it is divisible by b and a/b is a power of b. Write a function that takes parameters a and b and returns True if a is a power  
   of b.
8. Write a recursive function to calculate the factorial of a given number.
9. Write a function that takes a string as a parameter. Calculate the length of a string without using len function. Print the length concatenated with the  
   string and aligned towards the extreme right of the output screen.
10. ROT13 is a weak form of encryption that involves “rotating” each letter in a word by 13 places. To rotate a letter means to shift it through the alphabet,  
    wrapping around to the beginning if necessary, so ’A’ shifted by 3 is ’D’ and ’Z’ shifted by 1 is ’A’. Write a function that takes a string and an integer  
    as parameters, and then returns a new string that contains the letters from the original string “rotated” by the given amount. Use the built-in functions  
    ord, which converts a character to a numeric code, and chr, which converts numeric codes to characters.
11. Write a function that takes a nested list of integers and add up the elements from all of the nested lists.
12. Write a function called middle that takes a list and returns a new list that contains all but the first and last elements. So middle ([1, 2, 3, 4]) should return  
    [2, 3].
13. Write a program to print the keys of the dictionary and their values in an alphabetical order.
14. Write a function that takes any number of arguments and returns their sum.
15. Write a program that reads words.txt and prints only the words with more than 20characters (not counting white space).

**Self-Work/ Professional Skill Development activities**

Self-Work/ Professional Skill Development activities can be conducted by dividing the class into group of 7-8 students and same will be evaluated by board of faculty members along with the group report. Some of the suggested activities are:

|  |  |
| --- | --- |
| **PSDA 1** | 1. Learn how to write the code on interactive shell prompt and how to run the code 2. Python and its basics functional understanding 3. Write a function that prints all the prime numbers between 0 and limit where limit is a parameter |
| **PSDA 2** | 1. Learn the operator and conditional statements 2. Apply conditional statement and loop to create simple calculator 3. Extract the number that represents the month 4. How the nested condition works and write a program to compare three numbers and find the which one is the greatest number among three input numbers |
| **PSDA 3** | 1. Define the functions and type of functions 2. Learn the recursive function with stack diagram 3. How the infinite recursion works and process 4. What is different kind of string functions? 5. Define the Traversal, Comparison, Searching, Counting, Pre-defined String Functions, In Operator |
| **PSDA 4** | 1. Define the list, dictionaries and tuples 2. Apply the list, dictionaries and tuples with proper example 3. Write a program to implement the list, dictionaries and tuples |
| **PSDA 5** | 1. Describe and implement file handling in different modes 2. Accessing and Manipulating Files and Directories on 3. Apply the exception handling in python by using try, catch and finally |

**Assessment/ Examination Scheme:**

|  |  |  |
| --- | --- | --- |
| **Theory L/T (%)** | **Lab/Practical/Studio (%)** | **End Term Examination** |
| **75** | **25** | **100** |

**Theory Assessment (L&T):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Continuous Assessment/Internal Assessment** | | | | | **End Term Examination** |
| **Components (Drop down)** | Mid Term | Self-work /PSDA | Viva | Attendance | **EE** |
| **Weightage (%)** | 10 | 20 | 5 | 5 | 60 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Continuous Assessment/Internal Assessment** | | | | | | **End Term Examination** | |
| **Components (Drop down)** | | Lab Record | **Mid Term** | **Performance** | **Viva** | **Attendance** | **Experiment** | **Viva** |
| **Weightage (%)** | | 10 | 10 | 10 | 5 | 5 | 40 | 20 |

**Textbooks and References:**

* Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning, ISBN: 978-1111822705.
* Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd Edition) Author: Eric Matthes.
* Head-First Python: A Brain-Friendly Guide (2nd Edition)
* Learn Python the Hard Way: 3rd Edition.
* Python Programming: An Introduction to Computer Science (3rd Edition)