**LESSONPLAN**

NAME OF FACULTY : Dr Ravinder Rathee DISCIPLINE :Computer. Engg.

SEMESTER : 5th

SUBJECT : Programming in Python

**(15 Weeks Lession Plan)**

**WORKLOAD(LECTURE/ PRACTICAL):LECTURES-3,PRACTICALS-4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **WEEK** | **THEORY** | | **PRACTICAL** | |
| **1st** | **DAY** | **TOPIC** | **PRACTI CALDA Y/PERI**  **OD** | **TOPIC** |
| **1** | **UNIT1The way of the program**  The Python programming language, What is a  program? What is debugging? | 1  - 4 | * Introduction and python installation |
| **2** | Syntax errors, Runtime errors, Semantic errors,  experimental debugging. |
| **3** | **Variables, Expressions and Statements:** Values and data types, Variables, Variable names and  keywords, Statements, Evaluating expressions |
| **2nd** | **1** | Operators and operands, Type converter functions,Order of operations | 1-4 | * Let list1 and list2 be two lists of integers. Implement function sublist() that takes as input lists list1 and list2 and returns True if list1 is a sublist of list2, and False otherwise |
| **2** | Operations on strings, Input, Composition, The modulus operator. |
| **3** | **Conditionals:** Boolean values and expressions, Logical operators Simplifying Boolean Expressions |
| **3rd** | **1** | Conditional execution, Chained conditionals, Nested conditionals, The return statement, Logical opposites. | 1-4 | * Write function vowelCount() that takes a string as input and counts and prints the numberof occurrences of vowels in the string. |
| **2** | **UNIT IIIteration**: Assignment, Updating variables, The for loop, The while statement, |
| **3** | The Collatz 3n + 1 sequence, Tables, Two- dimensional tables |
| **4th** | **1** | Paired Data, Nested Loops for Nested Data. | 1-4 | * The cryptography function crypto() takes as input a string (i.e., the name of a file in the current directory). The function should print the file on the screen with this modification: Every occurrence of string 'secret' in the file should be replaced with string 'xxxxxx'. * Write a function stats() that takes one input argument: the name of a text file. The function should print, on the screen, the number of lines, words, and characters in the file;your function should open   the file only once. |
| **2** | **Strings**: Working with strings as single things, Working with the parts of a string, |
| **3** | Length, Traversal and the for loop, Slices, String comparison |
| **5th** | **1** | Strings are immutable, The in and not in operators,  A find function, Looping and counting, Optional parameters,. | 1-4 | * Implement function distribution () that takes as input the name of a file (as a string). This oneline file will contain letter   grades separated by blanks. Your function should print |
| **2** | The built-in find method, The split method, Cleaning up your strings, The string format method |
| **3** | **Test 1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | thedistribution of grades   * The function censor () takes the name of a file (a string) as input. The function should open the file, read it, and then write it into file censored.txt with this modification: Every occurrence of a four-letter word in the file should be replaced with string 'xxxx' * Create a dictionary for phones and their prices. Write functions to add a new entry (phone:price)   ,search for a particular phone and retrieve it’s price, given price findphones with same price ,  remove an entry, display all phones sorted according to price |
| **6th** | **1** | **Tuples:** Tuples are used for grouping data, Tuple assignment | 1-4 | * Write a Python program that prompts the user to enter a list of first names and stores them in a list. The program should display how many times the letter 'a' appears within the list * Write a Python program that prompts the user to enter integer values for each of two lists.It then should displays whether the lists are of the same length, whether the elements in each list sum to the same value, and whether there are any values that occur in both lists * Implement and test a Python program that determines if all parentheses in an entered line of code form matching pairs. Note: Pairs of   parentheses may be nested. |
| **2** | Tuples as return values,Composability of Data  Structures. |
| **3** | **Lists**: List values, Accessing elements, List length, List membership, List operations, List slices, |
| **7th** | **1** | Lists are mutable, List deletion, Objects and  references, Aliasing, Cloning lists | 1-4 | * Suppose variable s has been assigned in this way: s = '''It was the best of times, it was the worst of times; it was the age of wisdom, it was the age of foolishness; it was the epoch of belief, it was the epoch of incredulity; it was ...''' Then do the following, in order, each time: (a) Write a sequence of statements that produce a copy of s, named newS, in which characters ., ,, ;,   and \n have been replaced by blank spaces listS |
| **2** | Lists and for loops, List parameters, List methods, Pure functions and modifiers, |
| **3** | Functions that produce lists, Strings  and lists, list and range, Nested lists, Matrices. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **8th** | **1** | **Functions**: Functions with arguments and return values. | 1-4 | * The function avgavg() takes as input a list whose items are lists of three numbers. Each three-number list represents the three grades a particular student received for a course. * Implement function names () that takes no input and repeatedly asks the user to enter the first name of a student in a class. * Implement six classes to tree hierarchy model taxonomy with Python inheritance. |
| **2** | **UNIT III Modules:** Random numbers, The time module, The math module |
| **3** | Creating your own modules, |
| **9th** | **1** | Namespaces, Scope and lookup rules,  Attributes and the dot operator. | 1-4 | * Write a program that calculates the numeric value of a single name provided as input * Expand your solution to the previous problem to allow the calculation of a complete name |
| **2** | **Files**: About files, Writing our first file, Reading a file line-at-a-time |
| **3** | Turning a file into a list of lines Reading the whole file at once, |
| **10th** | **1** | Working with binary files, Directories, fetching something from the web | 1-4 | * Write a python program with function inner\_product(x,y) that computes the inner product of two (same   length) lists. |
| **2** | **List Algorithms:** Linear search, Binary search, Merging two sorted lists. |
| **3** | **Test 2** |
| **11th** | **1** | **UNIT IV Object oriented programming:** Classes and Objects- The Basics, Attributes, Adding  methods to our class | 1-4 | * The Sieve of Eratosthenes is an elegant algorithm for finding all of the prime numbers up to some limit n. |
| **2** | Instances as arguments and parameters, Converting an instance to a string |
| **3** | Instances as return values Objects are mutable, Sameness, Copying. |
| **12th** | **1** | **Exceptions**: Catching exceptions, raising our own exceptions, the finally clause of the try statement | 1-4 | * Write a function that returns the index of the smallest element in a list of integers. If the number of such elements is greater than 1, return the smallest index. Use the following header: def index Of Smallest Element (lst): Write a program that prompts the user to enter a list of numbers, invokes this function to return the index of the   smallest element, and displays the index |
| **2** | **Inheritance**: Polymorphism, Generalization, Pure functions |
| **3** | **UNIT V GUI:** Creating Graphical User Interfaces, Using Module Tkinter |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **13th** | **1** | Building a Basic GUI, Models, Views, and Controllers | 1-4 | * Write a program that reads an unspecified number of integers and finds the ones that have the most occurrences. |
| **2** | Customizing the Visual Style, Few More  Widgets |
| **3** | **Revision** |
| **14th** | **1** | **Databases:** Overview, Creating and Populating,  Retrieving Data, updating and Deleting | 1-4 | * Develop and test a Python program that allows a user to type in a message and have it converted into Morse code, and also enter Morse code and have it converted back to the   original message. |
| **2** | Using NULL for Missing Data, Using Joins to  Combine Tables |
| **3** | Keys and Constraints, Advanced Features. |
| **15th** | **1** | **Test 3** | 1-4 | **Revision** |
| **2** | Revision |
| **3** | Revision |