|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Application Development with Python** | | **L** | **T** | **P** | **C** |
| **20A05305** | **1** | **0** | **2** | **2** |
| **Pre-requisite** | **NIL** | **Semester** | **III** | | | |
|  | | | | | | |
| **Course Objectives:** | | | | | | |
| * To learn the basic concepts of software engineering and life cycle models * To explore the importance of Databases in application Development * Acquire programming skills in core Python * To understand the importance of Object-oriented Programming | | | | | | |
| **Course Outcomes (CO):** | | | | | | |
| Students should be able to   * Identify the issues in software requirements specification and enable to write SRS documents for software development problems * Explore the use of Object oriented concepts to solve Real-life problems * Design database for any real-world problem * Solve mathematical problems using Python programming language | | | | | | |
|  | | | | | | |
| **Module 1. Basic concepts in software engineering and software project management**  Basic concepts: abstraction versus decomposition, the evolution of software engineering techniques, Software development life cycle  Software project management: project planning and project scheduling  Task:  1. [Identifying the Requirements from Problem Statements](http://vlabs.iitkgp.ernet.in/se/1/)  **Module 2. Basic Concepts of Databases**  Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Relational Databases, [Data Definition Language(DDL) Statements: (Create table, Alter table, Drop](http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/exp1/index.php) [table)](http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/exp1/index.php), [Data Manipulation Language(DML) Statements](http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/exp2/index.php)  Task:   1. Implement [Data Definition Language(DDL) Statements: (Create table, Alter table, Drop table)](http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/exp1/index.php) 2. Implement [Data Manipulation Language(DML) Statements](http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/labs/exp2/index.php)   **Module 3. Python Programming:**  **Introduction to Python:** Features of Python, Data types, Operators, Input and output, Control Statements, Looping statements  **Python Data Structures:** Lists, Dictionaries, Tuples.  **Strings:** Creating strings and basic operations on strings, string testing methods.  **Functions:** Defining a function- Calling a function- Types of functions-Function Arguments- Anonymous functions- Global and local variables  **OOPS Concepts;** Classes and objects- Attributes- Inheritance- Overloading- Overriding- Data hiding  **Modules and Packages:** Standard modules-Importing own module as well as external modules Understanding Packages Powerful Lamda function in python Programming using functions, modules and external packages | | | | | | |

|  |
| --- |
| **Working with Data in Python:** Printing on screen- Reading data from keyboard- Opening and closing file- Reading and writing files- Functions-Loading Data with Pandas-Numpy  Tasks:   1. **OPERATORS**    1. Read a list of numbers and write a program to check whether a particular element is present or not using membership operators.    2. Read your name and age and write a program to display the year in which you will turn 100 years old.    3. Read radius and height of a cone and write a program to find the volume of a cone.    4. Write a program to compute distance between two points taking input from the user (Hint: use Pythagorean theorem) 2. **CONTROL STRUCTURES**    1. Read your email id and write a program to display the no of vowels, consonants, digits and white spaces in it using if…elif…else statement.    2. Write a program to create and display a dictionary by storing the antonyms of words. Find the antonym of a particular word given by the user from the dictionary using while loop.    3. Write a Program to find the sum of a Series 1/1! + 2/2! + 3/3! + 4/4! +…….+ n/n!. (Input :n = 5, Output : 2.70833)    4. In number theory, an abundant number or excessive number is a number for which the sum of its proper divisors is greater than the number itself. Write a program to find out, if the given number is abundant. (Input: 12, Sum of divisors of 12 = 1 + 2 + 3 + 4 + 6 = 16, sum of divisors 16 > original number 12)   **3: LIST**   1. Read a list of numbers and print the numbers divisible by x but not by y (Assume x = 4 and y = 5). 2. Read a list of numbers and print the sum of odd integers and even integers from the list.(Ex: [23, 10, 15, 14, 63], odd numbers sum = 101, even numbers sum = 24) 3. Read a list of numbers and print numbers present in odd index position. (Ex: [10, 25, 30, 47, 56, 84, 96], The numbers in odd index position: 25 47 84). 4. Read a list of numbers and remove the duplicate numbers from it. (Ex: Enter a list with duplicate elements: 10 20 40 10 50 30 20 10 80, The unique list is: [10, 20, 30, 40, 50, 80])   **4: TUPLE**   1. Given a list of tuples. Write a program to find tuples which have all elements divisible by K from a list of tuples. test\_list = [(6, 24, 12), (60, 12, 6), (12, 18, 21)], K = 6, Output : [(6, 24, 12), (60, 12, 6)] 2. Given a list of tuples. Write a program to filter all uppercase characters tuples from given list of tuples. (Input: test\_list = [(“GFG”, “IS”, “BEST”), (“GFg”, “AVERAGE”), (“GfG”, ), (“Gfg”, “CS”)], Output : [(„GFG‟, „IS‟, „BEST‟)]). 3. Given a tuple and a list as input, write a program to count the occurrences of all items of the list in the tuple. (Input : tuple = ('a', 'a', 'c', 'b', 'd'), list = ['a', 'b'], Output : 3)   **5: SET**   1. Write a program to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x). 2. Write a program to perform union, intersection and difference using Set A and Set B. 3. Write a program to count number of vowels using sets in given string (Input : “Hello World”, Output: No. of vowels : 3) 4. Write a program to form concatenated string by taking uncommon characters from two strings using set concept (Input : S1 = "aacdb", S2 = "gafd", Output : "cbgf"). |

|  |
| --- |
| **6: DICTIONARY**  a. Write a program to do the following operations:   1. Create a empty dictionary with dict() method 2. Add elements one at a time 3. Update existing key‟s value 4. Access an element using a key and also get() method 5. Deleting a key value using del() method   b. Write a program to create a dictionary and apply the following methods:   1. pop() method 2. popitem() method 3. clear() method   c. Given a dictionary, write a program to find the sum of all items in the dictionary.  d. Write a program to merge two dictionaries using update() method.  **7: STRINGS**   1. Given a string, write a program to check if the string is symmetrical and palindrome or not. A string is said to be symmetrical if both the halves of the string are the same and a string is said to be a palindrome string if one half of the string is the reverse of the other half or if a string appears same when read forward or backward. 2. Write a program to read a string and count the number of vowel letters and print all letters except 'e' and 's'. 3. Write a program to read a line of text and remove the initial word from given text. (Hint: Use split() method, Input : India is my country. Output : is my country) 4. Write a program to read a string and count how many times each letter appears. (Histogram).   **8: USER DEFINED FUNCTIONS**   1. A generator is a function that produces a sequence of results instead of a single value. Write a generator function for Fibonacci numbers up to n. 2. Write a function merge\_dict(dict1, dict2) to merge two Python dictionaries. 3. Write a fact() function to compute the factorial of a given positive number. 4. Given a list of n elements, write a linear\_search() function to search a given element x in a list.   **9: BUILT-IN FUNCTIONS**   1. Write a program to demonstrate the working of built-in statistical functions mean(), mode(), median() by importing statistics library. 2. Write a program to demonstrate the working of built-in trignometric functions sin(), cos(), tan(), hypot(), degrees(), radians() by importing math module. 3. Write a program to demonstrate the working of built-in Logarithmic and Power functions exp(), log(), log2(), log10(), pow() by importing math module. 4. Write a program to demonstrate the working of built-in numeric functions ceil(), floor(), fabs(), factorial(), gcd() by importing math module.   **10. CLASS AND OBJECTS**  a. Write a program to create a BankAccount class. Your class should support the following methods for   1. Deposit 2. Withdraw 3. GetBalanace 4. PinChange   b. Create a SavingsAccount class that behaves just like a BankAccount, but also has an interest rate and a method that increases the balance by the appropriate amount of interest (Hint:use Inheritance). |

|  |
| --- |
| c. Write a program to create an employee class and store the employee name, id, age, and salary using the constructor. Display the employee details by invoking employee\_info() method and also using dictionary ( dict ).  d. Access modifiers in Python are used to modify the default scope of variables. Write a program to demonstrate the 3 types of access modifiers: public, private and protected.  **11. FILE HANDLING**   1. . Write a program to read a filename from the user, open the file (say firstFile.txt) and then perform the following operations:    1. Count the sentences in the file.    2. Count the words in the file.    3. Count the characters in the file. 2. . Create a new file (Hello.txt) and copy the text to other file called target.txt. The target.txt file should store only lower case alphabets and display the number of lines copied. 3. Write a Python program to store N student‟s records containing name, roll number and branch. Print the given branch student‟s details only. |
| **References:** |
| 1. Rajib Mall, “Fundamentals of Software Engineering”, 5th Edition, PHI, 2018. 2. RamezElmasri, Shamkant, B. Navathe, “Database Systems”, Pearson Education, 6th Edition, 2013. 3.Reema Thareja, “Python Programming - Using Problem Solving Approach”, Oxford Press, 1st Edition, 2017.   4. Larry Lutz, “Python for Beginners: Step-By-Step Guide to Learning Python Programming”, CreateSpace Independent Publishing Platform, First edition, 2018 |
| **Online Learning Resources/Virtual Labs:** |
| 1. <http://vlabs.iitkgp.ernet.in/se/> 2. <http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/index.php> 3. [https://python-iitk.vlabs.ac.in](https://python-iitk.vlabs.ac.in/) |