|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Topic** | **Program Assignment** |
| 41 | Dictionary  and Date & Time | Using calendar module perform following operations. a) Print the 2016 calendar with space between months as 10 characters. b) How many leap days between the years 1980 to 2025. c) Check given year is leap year or not.  d) print calendar of any specified month of the year 2016. |
| 42 | Functions | Write a program to generate a Fibonacci series using a function called fib(n),  a) Where ‘n’ is user specified argument specifies number of elements in the series. |
| 43 | Functions | Write a program to search given element from the list. Use your own function to search an element from list. **Note**: Function should receive variable length arguments and search each of these arguments present in the list. |
| 44 | Functions | Write a program with lambda function to perform following. a) Perform all the operations of basic calculator (Add, Sub, Multiply, Divide, Modulus, Floor division) |
| 45 | Functions | Write a program to check given string is Palindrome or not. (Use function Concepts and Required keyword, Default parameter concepts) i.e Reverse the given string and check whether it is same as original string, if so then it is palindrome. Example: String "malayalam" when reversed will be "malayalam" hence palindrome. |
| 46 | Functions | Write a function to find the biggest of 4 numbers. a) All numbers are passed as arguments separately (Required argument) b) use default values for arguments (Default arguments) |
| 47 | Functions | Write function to extend the tuple with elements of list. Pass list and Tuple as parameter to the function. |
| 48 | Functions | Create a Calculator with the following functions. a) Addition/subtraction/multiplication and division of two numbers (Note: Create separate function for each operation) b) Find square root of a given number. (Use keyword arguments in your function) c) Create a list of sub strings from a given string, such that sub strings are created with given character. i.e. String = "Pack: My: Box: With: Good: Food" Create sub strings with the delimiter character ":" such that the following sub strings are created. substrlist=[Pack, My, Box, With, Good, Food] Note : Function should take at least 2 parameters ( Main string and delimiter character) return value from function will be list of substring. |
| 49 | File I/O Operations | Write a program to perform following file operations a) Open the file in read mode and read all its contents on to STDOUT. b) Open the file in write mode and enter 5 new lines of strings in to the new file. c) Open file in Append mode and add 5 lines of text into it. |
| 50 | File I/O Operations | Write a program to open the existing file in read mode and perform following tasks, a) Rad 10 character at a time and then print its current position of file object. Repeat this operation till the EOF. b) Reset the file pointer after reading 100 Character from file (Use Seek function to reset) c) Open the file in read mode and start printing the contents from 5th line onwards. |
| 51 | File I/O Operations | In a given directory search all text files for the pattern "Treasure".  a) Find how many text files has the pattern. b) Count how many times pattern repeats in each file **Note**: Create at least 4 text files in a directory and keep the pattern in at least 2 files. Repeat the pattern in the file many times. |
| 52 | File I/O Operations | Open existing text file and reverse its contents. i.e a) print the last line as first line and first line as last line (Reverse the lines of the file) b) print characters of file from last character of file till the first character of the file.(Reverse entire contents of  file ) |
| 53 | File I/O Operations | Open the file is read & write mode and apply following functions a) All 13 functions mentioned in Tutorial File object table. |
| 54 | Exception Handling | Write a program to handle the following exceptions in you program. a) KeyboardInterrupt b) NameError  c) ArithmeticError **Note**: Make use of try, except, else: block statements. |
| 55 | Exception Handling | Write a program for converting weight from Pound to Kilo grams. a) Use assertion for the negative weight. b) Use assertion to weight more than 100 KG |
| 56 | Exception Handling | Write a program to handle following exceptions using Try block. a) IO Error while you try writing contents into the file that is opened in read mode only. b) ValueError |
| 57 | Exception Handling | Try implementing atleast any 5 exceptions in you program. |
| 58 | Modules & Packages | Create file called  "calc.py" which has following functions i) functions to add 2 numbers ii) function to find diff of 2 numbers iii) function to multiply 2 numbers iv) all maths operations ( sqrt, div, floor div, modulus, primenumber) v) Fibonacci series          a) Write a new program in file "maths.py" such that you import functions of file "calc.py" to your new program b) Use From <module> import <function> statement to import only few function  from calc module. |
| 59 | Modules & Packages | Create file called "stringop.py" which has following functions i) functions to sort numbers (Use loops for  sorting, do not use built in function) ii) function to search given element through binary search method.(Refer to net for the Binary search algorithm) iii) function to reverse the given string  Write new program in file strpackage.py such that you import functions of file "stringop.py" to your new program |
| 60 | Modules & Packages | Create a package of all programs you have done in earlier.         a. All programs related to strings - Stringpackage         b. All programs related to Lists -ListPackage         c. All programs related to Tuple - TuplePackage         d. All programs related to Dictionary -DictionaryPackage         e. All programs related to Functions - FunctionPackage         f. All programs related to Files  -- FilePackage |