P E S University

Department of Computer Science & Engineering Session : Aug-Dec 2018

Introduction to Computing using Python Laboratory UE18CS102

Mini-project Titles

1. Simulate cal command
2. Solve system of linear equations
3. Develop a small statistical library
4. Plot graphs using trigonometric functions
5. Develop an analog clock
6. Develop a spell checker, may suggest the best possible word can be added to the dict
7. Check a python program for indentation errors and for few syntax errors
8. Address Book: The user wants to create an address book and downloads your program. How would you make it? Create a program that prompts the user for the information in most address books and then stores it in a .txt file!
9. Solve electric circuits
10. Write a program that takes a list of filenames as arguments and prints contents of all those files, like cat command in unix.
11. Write a program that takes one or more filenames as arguments and prints all the lines which are longer than 40 characters.
12. Write a function to find files that recursively descends the directory tree for the specified directory and generates paths of all the files in the tree.
13. Write a function to compute the number of python files (.py extension) in a specified directory recursively.
14. Write a function to compute the total number of lines of code in all python files in the specified directory recursively.
15. Write a function to compute the total number of lines of code, ignoring empty and comment lines, in all python files in the specified directory recursively.
16. Write a program split.py, that takes an integer n and a filename as command line arguments and splits the file into multiple small files with each having n lines.
17. Find inverse of a matrix (make it complex by including few more functionalities – saddle point, trace, norm, sorting & so on)
18. TextBased Adventure Game : A complete text game, the program will let users move through rooms based on user input and get descriptions of each room. To create this, you'll need to establish the directions in which the user can move, a way to track how far the user has moved (and therefore which room he/she is in), and to print out a description. You'll also need to set limits for how far the user can move. In other words, create "walls" around the rooms that tell the user, "You can't move further in this direction."

19. Dice Rolling Simulator: Like the title suggests, this project involves writing a program that simulates rolling dice. When the program runs, it will randomly choose a number between 1 and 6. (Or whatever other integer you prefer — the number of sides on the die is up to you.) The program will print what that number is. It should then ask you if you'd like to roll again. For this project, you'll need to set the min and max number that your dice can produce. For the average die, that means a minimum of 1 and a maximum of 6. You'll also want a function that randomly grabs a number within that range and prints it.
20. Mad Libs Generator : The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You'll need prompts for user input, and to then print out the full story at the end with the input included.
21. Sentence Generator: A series of diferent parts of sentences will be randomly put together to come up with new interesting sentences.
22. . Suduko Puzzle
23. Cross word Puzzle
24. . N queens problem
25. Cannibal missionery problem
26. Create a module to play with files with fns like i)
    rename file
    ii) delete file
    iii) copy file
    iv) disp file
    test these fnsalso take care of errors ; give suitable message
27. Mastermind : player tries to guess a colour and position combination of four coloured pegs, with the computer giving an indication of how close each guess is . . .
28. Shoutbox : a place where multiple people can write short messages which are stored and published.
29. Meme Quiz : the user answers various questions, the answers are then given points and added up, resulting in the user receiving a description/graphic of what their score represents
30. Write a function which gets # of strings using variable # of arguments and Include function to
    - returns a list of strings which are palindromes. -
    find the characters in each string uniquely
    - find all strings which are anagrams of the first string
31. Create a class called MyStack which supports push and pop operations. Implement the stack class using a list. specify the upperbound of the size while creating the stack object. Provide exception handling mechanism for stack overflow and stack underflow.
32. Create a tree and traverse
33. Create a data structure to hold event info.
    Name,detail date,dd mm yy,people
    - create a list of events
    - arrange them in chronological order
    - perform few more operations on the same

34. Create a list where each element of the list contains empid, empname, basic, da, hra.
    - Find the salary for each person.
    - Store the result in a dictionary with empid as the key and total salary as the value.
    - provide additional operations on emp data
35. Store fall of wickets into a list.
    a) find partnership for each wicket - store in a list b) find highest partnership
    c) give details about no. Of balls d) no. of dot balls
    e) no. of 4's & 6's in partnership & individual
36. Create a class called BankAccount with fields acno and name. Write functions to get and set these fields. Create a class called FixedDeposit which inherits from BankAccount and has additional fields duration, amount and rate of interest. Create a class called RecurringDeposit which inherits from BankAccount and has additional fields duration and monthly payment and rate of interest. Create functions to read and display information in each class.
37. Create a module MyMath which contains useful functions like areas of different shapes and volumes of diferent solids. Write a client program to use these functions.
38. Create a class student. Fields required are name, usn, marks for 5 subjects, semester. Write functions to create new student, delete student object. Create a list of students. Search for a particular student in the given list based on usn. Find average of all students. Find all students with average greater than a given value, all students who failed in atleast one subject, all students passed in all subjects.
39. Write a function to traverse a directory. Call this function to print all files with path in each level.
40. Write your own function myfilter and use it to find
    a) list of all regular files; input is the list of all entries in the directory b) list of all non zero numbers ; input is a list of numbers
    c) list of all strings whose length is less than a given number; input is a list of strings
41. Write your own function mymax and use it to find a) biggest string in a list of strings
    b) biggest string in a list of strings ignoring the case (case insensitive) c) biggest string in a list of strings based on the length
42. Write your own function mymap and use it to generate
    a) list of tuples having string and its length, given a list of strings b) list of filename and its size given a list of filenames
    c) list of reversed string given a list of strings
43. Create a dictionary for words and their meanings. Write functions to
    a) search for a particular word
    b) search based on a partial match (in the beginning of the string) c) find words with same meaning. (two keys having the meaning)
44. Create a phone book – entries should come from file. One name can have multiple phone numbers. Write functions to add a new entry (from keyboard), search for an entry, print all entries in sorted order, delete entry. Use exception handling.

45. Write a function which takes a path as argument and an outfile name. It reads all the files in that path and creates a combined file(tar). Write a function to untar such a file into number of constituent files in a given directory.
46. Write a function to return a list all files which have execute permission and whose size exceeds a given number. Write a function to create a directory with a given name and move all these files to that directory.
47. Given a list of strings, for each string find the anagrams in the given list. Find the anagram which has the highest number of strings – if more than one has the same count, display all of them.
48. Given two files of words, compare the two files. Print those words which are in both the files, those in first and those in second.
49. A file contains names of batsmen and their scores in an innings. Name of the batsman can repeat. Write a program to find the total for each batsman. Display
50. Given an English passage,
    - remove all punctuations
    - convert all words to lowercase - find
    frequency of each word
    - count no. of unique
    - find # of words based on the length - create a
    list of list of words
    list[0] is a list of words starting from a
    list[1] is a list of words starting from b & so on
51. Given an English passage,
    - find # of words based on the length, create a dictionary with length as key & list of words with that length as values.
    - create a dictionary where the key is the first letter of the word and the value is set of unique words starting from that letter.
    - create a set of unique words.
    - create a dictionary where the key is the first letter of the word and the value is ordered list of unique words starting from that letter.
52. Write a function which checks whether a given year is prime. Write a function to validate a given date – use the former function to check for prime. Write a function to convert a given date to Julian Date.
    1 st March 2013 => 31 + 28 + 1 => 60 th day of the year 1 st
    March 2016 => 31+ 29 + 1 => 61 st day of the year

53. Character recognition System (Basic one)
54. Object recognition system (Simple one)
55. Blending of two images to create a new image
56. Scraping images from the web
57. Scraping the text from the web page
58. Visualizing Logic Circuits with Pygame
59. Guessing the actual string out of mixed string
60. Scientific calculator
61. Generate strings for the given image /audio/vedio
62. Simulate the data sender and receiver using very good encoding schemes
63. Railway reservation systems
64. Simulate a stack/Queue using list
65. Simulate a Binary search tree using list
66. Removal of stop words from the given text

67. Billing system for any application
68. Automating the feedback system
69. Simulate Super Sprint Game
70. Given a sentence, classify the words in it to different parts of speech. Use files or NLTK 27. Input a sentence from the user. Check it belongs to which type of sentence. Assertive, Interrogative, exclamatory, Imperative
71. Input a sentence from the user, Check whether it is there in passive voice or active voice
72. Generate different waves using python code (sine,cosine,tan)
73. Search for the file in our system using python code
74. Dig information on different new libraries: beets, newsBlur, boto, salt

75. Noughts and crosses with GUI
76. Guess a colour and position combination of four coloured pegs
77. Random quote generator
78. create a GUI where multiple people can write short messages which should be stored and published
79. Sentence Generator: A series of differents words are randomly generated,arrange them semantically and generate interesting sentences
80. Area Calulator: given an 3D geometric shape caluculate its area
81. Use of Opencv to segment text image into lines

82. Cost estimation GUI based for the construction of house
83. Data detailing in a text file
84. Unit Converter (temp, currency, volume, mass and more)
85. Graph coloring having 4 nodes
86. Ceasar Ciphers on files
87. Generating unique QR code
88. Design of SUDOKU ---- 9*9 Game
89. Inventory System
90. Password Generator(generates and identifies the strength with measures to improve)
91. Help desk for IT employees

92. Discussion Forum
93. Block chain
94. Face Recognition
95. Image classification
96. STUDENT MANAGEMENT SYSTEM that manages every detail related to a student and also keeps a record of all their requests and grievances.
97. Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game)
    Remember the rules:
    Rock beats scissors
    Scissors beats paper
    Paper beats rock

98. Create a program that will play the “cows and bulls” game with the user. The game works like this:
    Randomly generate a 4-digit number. Ask the user to guess a 4-digit number. For every digit that the user guessed correctly in the correct place, they have a “cow”. For every digit the user guessed correctly in the wrong place is a “bull.” Every time the user makes a guess, tell them how many “cows” and “bulls” they have.

    Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout teh game and tell the user at the end.

99. Draw a game board.Ask the user what size game board they want to draw, and draw it for them to the screen using Python’s print statement.

100. Build IRC chat robot using python
101. CBT analysis (Reading data from excel files and generating class average, finding highest, lowest and generating reports)

102. No due management using excel files/text files
103. Checking duplicate questions in a question bank (question bank is stored in a text file)
104. find the team who won most of the times at a particular stadium in IPL
105. simulate traveling sales man problem
106. Rat in a Maze using recursive technique
107. Simulate library management system
108. develop a fitness module on
     1)BMI Index(indicate overweight,under weight)
     2)for every 500 steps, burn of calories
     3)BP monitor(Indicate low,high,normal)
     4)pulse

5)sugar
6)Heart Beat

109. develop a fitness module on
1)BMI Index(indicate overweight,under weight)
2)for every 500 steps, burn of calories
3)BP monitor(Indicate low,high,normal)
4)pulse
5)sugar
6)Heart Beat

110. Write a program that takes various kind of content from the user and classifies and stores them.
Example: If the Topic is Programming Languages. Under that you have sub topics like (Java, Python3,C,C++).If suppose you have an article or something of that sort of JAVA .The program should enable you to store it under that sub topic .Like this many subtopics under each subtopics will be created which will give rise to a tree. When each article is stored , there should be an option to add a tag or a keyword to it so that all articles related to a particular keyword can be searched . Like this many filters can be added to every entry so that this will able the user to easily classify information on any number of topics

111. Alarm Clock which launches a YouTube video - Make a program that accepts command line arguments for what time to go off, and when it does it should launch a YouTube video in your browser that will start playing. The program should read in a text file that contains URLs to different YouTube videos and will randomly choose one and launch it.

112. What's The Weather? - Create a program that pulls data from OpenWeatherMap.org that prints out information about the current weather, such as the high, the low, and the amount of rain for wherever you live. Print out data for the next 5-7 days so you have a 5 day/week long forecast. Print the data to another file that you can open up and view at, instead of viewing the information in the command line.

113. Vocabulary Quizzer - Use https://www.randomlists.com/random-vocabulary-words to generate a random list of 5 vocabulary words to test your user with. Scrape the words and definitions from the page source code to create a quiz-type of your choice. Should create a multiple choice quiz with instances of one of the scraped word and a randomized list of the 5 scraped definitions for the user to choose from. (Uses non-standard library BeautifulSoup for web scraping.)

114. Code your own computer generated modern Art