SOUTHERN UNIVERSITY COLLEGE

SEMESTER B

YEAR 2024

PRACTICAL

**BTPR3203 PYTHON FOR DATA SCIENCE**

DATE : 09/08/2024 DURATION : 3 HOURS

BACHELOR OF SOFTWARE ENGINEERING (HONOURS)

YEAR THREE

This is a test. It is worth 20% of the overall marks of this course. It is to assess the following learning outcome:

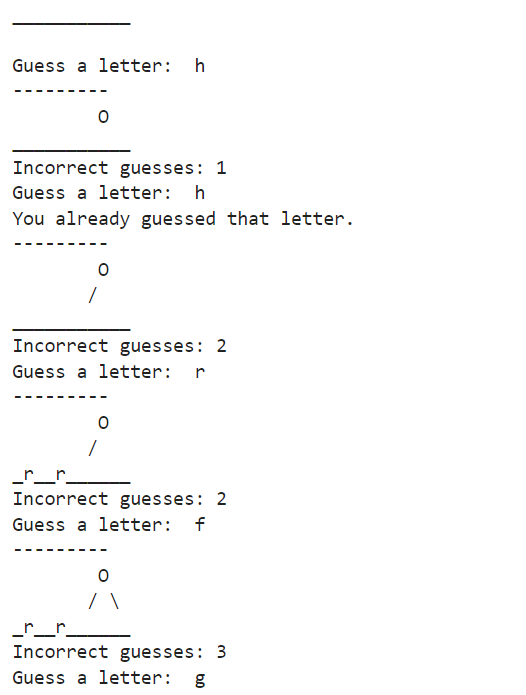
|  |  |
| --- | --- |
| CLO2 | Demonstrate the concepts and techniques of programming with Python by using modularisation, different structured types, file and exception handling (C3, PLO2). |
| CLO3 | Apply the knowledge of programming with Python and Python libraries to develop a solution (C4, PLO8). |

This is an open-book question. Submit the GitHub link of your Jupyter Notebook files via Microsoft Teams.

|  |  |
| --- | --- |
| Student ID | B220173C |
| Name | Tong Lee Qi |

Q1. Write a Python program that simulates a text-based hangman game where the player guesses letters to reveal a hidden word. Setup the game as follows:

* Create a list of words to be used for the game. The length of each word must be more than six.
* Randomly select a word from the list.
* Initialize an empty list to represent the hidden word with underscores. If the word to be guessed is “P R O G R A M M I N G”, the respective underscores would be \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_.
* Set the number of allowed incorrect guesses.
* While the number of incorrect guesses is less than the limit and the word is not fully guessed:
  + Display the current state of the word (underscores and guessed letters).
  + Prompt the user to guess a letter.
  + Check if the letter has already been guessed. If the letter is correct, update the hidden word. If the letter is incorrect or has already been guessed, increment the number of incorrect guesses.
  + Display the updated game state (word, guesses remaining, hangman visual).
* If the player guesses all letters correctly, display a winning message.
* If the player runs out of guesses, display a losing message and reveal the word.



(25 marks)

[Total : 25 marks]

Q2. Download the “spotify.csv” dataset and write a Python program to read the dataset into a DataFrame. From the dataset, find the average loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo and duration (in minutes) for each genre that are released in the year 2017 only. Save the findings into a .csv file.

Additionally, investigates the relationship between danceability and energy, and other factors that may influence danceability and energy. (25 marks)

[Total : 25 marks]

Q3. Download the “coffee.csv” dataset and write a Python program to read the dataset into a DataFrame. From the dataset, find the total order of each product type. In a DataFrame, store the findings as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **category** | **product** | **total\_order** |
| **0** | Coffee | Barista Espresso | … |
| **1** | Bakery | Biscotti | … |
| **…** | … | … | … |

Then, create a plot to visualise the DataFrame above. (25 marks)

[Total : 25 marks]