**Project Report**

**Introduction:**

Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word and the other tries to guess it by suggesting the letters. The word to guess is represented by a row of dashes, giving the number of letters. If the guessing player suggests a letter which occurs in the word, the program writes it in all its correct positions. If the suggested letter does not occur in the word, the other player draws one element of the hangman diagram as a tally mark. The game is over when the guessing player correctly guesses the whole word.

HOW TO PLAY: Our code will generate a word which has to be guessed by the player. So, at the output screen will exist a set of instructions for the player, an entry box where the user will type a word and the stand from which the stick figure of the hangman will begin to form every time the user guesses wrong. Then the player will guess a letter. If that letter is in the word(s) then the project will write the letter at everyplace it appears, and cross out that letter in the alphabet. If the letter isn't in the word then we cross out the lifelines (which in our project are six chances) from the list. The player will continue guessing the letters until he can either solve the word (or phrase) or he will end up losing all the chances and a prompt will appear saying that the game has been lost.

So, it is basically a two player game. But in our project a single player plays the game and the rules are strictly followed by the code.

**Logic:**

The overall code of the project can be broken down in to the following parts:

1. We formulated a word list and stored them in an Array collection with the name “LOW” (List of words).
2. We used a library known as “import random” which will be used in the code to pick random words from the array.
3. The actual method which does the logical reasoning, whether the letter exists or not, if yes, write it down at all the places else decrease the chance. We did this by using the selection statement of IF-ELSE. This forms the main part of the code.
4. We also displayed a prompt “You have won the game” if the user guessed the word correctly else the prompt “You have lost the game” will be shown.
5. In order to take the input, we put the logic code of the word into a function. We made another function where we called the function of the logic code. The parameter of the logic code is the input.
6. Finally, the GUI coding which forms the user interactive screen will mainly prevail during the code output.

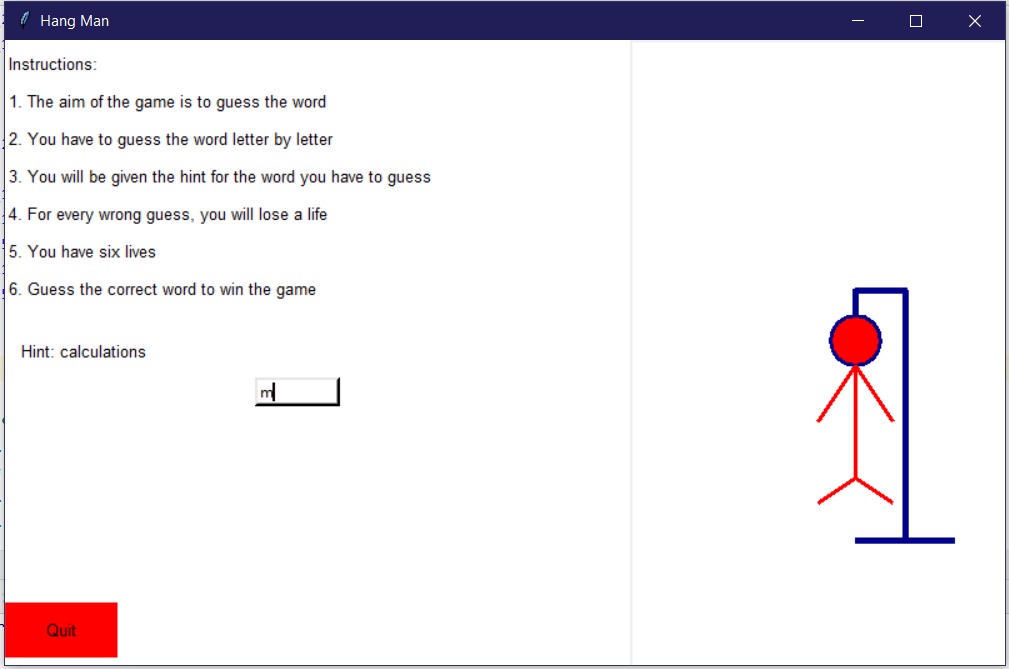
For GUI, we used the Tkinter library and for the code we have used Python language. The code was written on the IDE Pycharm.

**Challenges:**

We faced many challenges and problems throughout formation of this project. The most notable one was we had to learn the syntax for the Python programming language. Before shifting to python, we had to write the whole code in C++ and then convert it to python. Besides that, when writing the code in C++ we had to figure out on how to perform the “random” function which would randomly pick words from the array. We also had to figure out on which graphics library to use and ended up choosing Tkinter. The graphics portion of the project was definitely the toughest as we had to learn all the syntax, the whole process of taking the letter input from the user and then storing it in a variable and processing it was also a challenge.

If we had the chance to do the project again, we would make proper use of functions in order to make the code shorter and easier. Moreover, if we used file handling, we could have stored more words for the user to guess which would also have made our code much shorter and efficient.

**Output:**

****

This is how the final GUI interface will look like.

**Group members:**

1. M.Talal Farrukh

2. M.Uzair Akhtar

3. M.Huma Tahir

4. M.Salman Shaheer