

6.12: Worksheet 6

Just like the previous set of exercises, make sure that you write a separate function, which is then run by the main(). You will need to keep all your functions, so that I can look at them. It will be a good idea if you were to put a comment above stating which question you are answering. For each of these questions, consider creating a new Python file within your project (called something related to the question).

For each of these questions, make sure that (where appropriate), you are validating the entry made by the user.

1. Write a program which simulates the “Magic 8 Ball” (for information on the magic 8 ball, see this website: [en.m.wikipedia.org/wiki/Magic\_8-Ball](file:///C:\Users\kbika\Downloads\en.m.wikipedia.org\wiki\Magic_8-Ball)). The user should enter a yes-no question, then the magic 8 ball will tell them an answer. You should assume that all 20 possible answers according to Wikipedia are options for your implementation.
   1. Identify the key information for the case study
   2. Draw the flow chart(s) for the “magic 8 ball”
   3. Write the code for the program in Python
2. You have been asked to program a single-player rock, paper, scissors game. The user will enter their chosen option and the computer will randomly choose rock, paper, or scissors. The winner will be identified and output. The user will then be asked whether they want to play again. If the answer is yes, the game starts again. If the user decides to stop, their player stats will be displayed – number of wins, number of losses, percentage won.
   1. Identify the key information for the case study – look at the questions in the slides in section 6.8 to help with this
   2. Draw the flow chart(s) for the rock, paper, scissors game
   3. Write the code for the program in Python
3. Write a program which allows a user to play a “Word Jumble” game. Your program should store a list of words (which you as the developer define). When the user plays the game, the computer will pick one of the words at random and then randomizes the letter. The user has to guess what the original word chosen by the machine is.
   1. Identify the key information for the case study
   2. Draw the flow chart(s) for the “word jumble” game
   3. Write the code for the program in Python
4. Create a program that will play the “cows and bulls” game with the user. The game works like this:

*A 4 digit number is randomly generated. 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 the game and tell the user at the end.*

* 1. Identify the key information for the case study
  2. Draw the flow chart(s) for the “cows and bulls” game
  3. Write the code for the program in Python

*Hint: when you are testing the functionality, you might want to get the generated random number to be printed to the screen – then you can check that the bulls and cows are calculated correctly and that it stops when it should* ☺

1. [Challenge] Write a program which allows two players to play Noughts and Crosses (or Tic Tac Toe). You can assume that the two players are on the same machine. You will need to find a way of outputting the board in the Python console.
   1. Identify the key information for the case study
   2. Draw the flow chart(s) for the “cows and bulls” game
   3. Write the code for the program in Python
2. [Challenge] Adapt your answer to the previous question to allow the user to choose whether they play against another person, or a computer player. You can assume that the computer player makes random moves (but only chooses to play their piece in an empty space).