**Munster Technological University - Kerry**

**Ord/Hons BSc. in Computing with Specialism - Year 1**

**Continuous Assessment #1 (Group X)**

**Date: 19/2/24**

**Time: 9.10 – 10.45 a.m.**

**Structured Programming 2**

**Instructions:** Copy this file to the desktop of your machine. You may also download and refer to the file **PythonSpellChecker.docx** during the CA. Disconnect from the Internet at the start of the CA and close all other Python module materials that you might have access to, before attempting the following question.

Launch IDLE and create an empty Python file called ***FirstnameLastname***.py on the desktop. When you are finished coding, upload your Python file through Canvas. You should keep a copy of your Python file on your OneDrive, unaltered, in case I need to access it at a later stage.

Please don’t logout of Canvas until I have confirmed that I have received your code successfully.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q1.**

A Python program must be written which will play a portion of exactly 3 games of “Aces High”. The program will begin by asking the user, within a user-defined function named **main**(), to enter the names of the two players who will be playing the game on input dialogs. This function takes **no arguments** and returns nothing. The function will also create a **Tk** window and a **text-area** that will be added to the Tk window. In this case, you can give the text-area a width of 50, a height of 5 and it should use a “courier” font with a point-size of 12. The text-area will be displayed on the Tk window. The main() function will also contain a **loop** that iterates three times, allowing for three games to be played.

Each time this loop iterates, a separate, user-defined function named **dealCard**() will be called **twice**, to deal a card for each of the two players.

The function **dealCard**() will take **no arguments**, will generate a card randomly and return this card as a string e.g. “2 of Clubs”, “10 of Diamonds”, “Queen of Spades” etc. In order to generate a random card, the dealCard() function will end up calling the **randint**() function twice. The first call will generate a random number between 2 and 14 inclusive to represent the **rank** of the card - note here that 11 will be “Jack”, 12 will be “Queen”, 13 will be “King” and 14 will be “Ace”. The randint() function will then be called again to generate a random number between 1 and 4 inclusive to represent the **suit** of the card, so 1 will be “Hearts”, 2 will be “Diamonds”, 3 will be “Clubs” and 4 will be “Spades”. When the function has been called twice, then the string created from the values returned by these calls (which describe the card being dealt) can simply be returned from the function back to the main() function and stored in appropriate variables.

When the function dealCard() has been called twice, another function named **displayCards**() will be called by main(). This function will take **six arguments** – these will represent the names of the two players, the cards dealt for each player, the loop counter variable and a reference to the text-area. With this information, the function displayCards() will be able to add text to the text-area, displaying the game number, the names of the two players along with the cards that they were dealt. Note that the data displayed inside the text-area should be **neatly aligned** as indicated in the sample screenshots.

Once the data has been displayed on the text-area, immediately a message-box should appear, which has the effect of holding up the programs execution. This message-box just displays the message "Please hit return to continue ...." and, as soon as the user does this, the message-box disappears, the contents of the text-area is cleared, and the function exits, returning control back to the main() function again, at which point the loop iterates to play another round of the game.

Once the loop has completed its third iteration, the program should just issue a closing message on a message-box as indicated in the sample screenshots below.

For full marks here your program should, along with a logically correct solution for the problem above, include a meaningful **comment** at the top of the program which should briefly explain the purpose of the program.

Your program should run as indicated in the following sample screenshots. Of course, the results from your program will not be exactly the same, as random number generation is involved so the cards dealt will vary.

**Sample Screenshots**

**The main() function is called and the user is asked to enter the names of the two players**

A screenshot of a computer error

Description automatically generated A screenshot of a computer error

Description automatically generated

**Once the user hits the OK button on the second input dialog (or hits return), the loop then begins, dealCard() gets called twice and then displayCards() gets called, displaying a Tk window containing a text-area. The text-area holds details about the game number, players and cards that have been dealt, neatly aligned**

A close up of a name

Description automatically generated

A screenshot of a computer error

Description automatically generated

**When the user hits return on the message-box, the text-area is cleared of its current data and the loop process then repeats for two more iterations**

A close-up of a card

Description automatically generated

A screenshot of a computer error

Description automatically generated

A close-up of a card

Description automatically generated

A screenshot of a computer error

Description automatically generated

**When the user hits return on the message-box at this point, the loop exits and a message-box appears giving the user a farewell message**

A screenshot of a computer error

Description automatically generated