

CS2011 Intermediate programming and problem solving I

Week 4: Lab Assignment

*Due: **Friday October 11th, 5pm (Week 5).***

This lab is worth 5% of your total marks. You will receive a mark out of 5.

PLEASE READ THESE INSTRUCTIONS CAREFULLY.

Completion of lab exercises is **mandatory**. These lab exercises must be completed using Pycharm or repl.it. As well as answering the question provided, you must appropriately comment your code to demonstrate your understanding of the code you wrote. **You will lose marks for uncommented code!**

Following completion, you must upload each solution on the appropriate assignment:

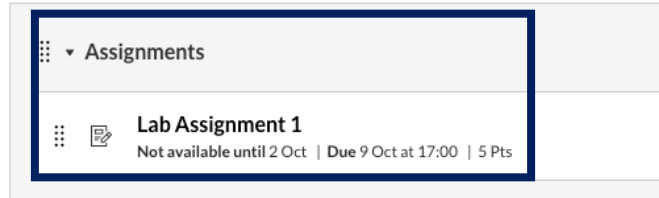
- On canvas (as separate .py files or in .zip format); **and**
- Upload to your repl.it account. On the top of your python code, you should write a comment with the link to that specific repl.it file, and not your repl.it profile link.

You should complete these actions by the due date specified above. Please note the following concerning continuous assessment material:

- No late work will be accepted. The due date is final
- You cannot repeat a continuous assessment

Further submission instructions:

- The submission location for these lab assignments can be found on the CS2011 course page on canvas, under **Assignments -> Assignments -> Lab Assignment 1.**



- For this assignment, you will only need one file. You will be building on this python code: <https://repl.it/@lauramaye/participant>
- Copy/download this file. Rename this file in the format:
 - `ID_Lab1Assignment.py`
 - For example, for student ID number 123456 answering Question 1 A, the file should be named in the format:
`123456_Lab1Assignment.py`

Plagiarism check

The Python code of the practical will be checked for plagiarism following the UCC guidelines: <https://www.ucc.ie/en/media/support/recordsandexaminations/documents/UCCPlagiarismPolicy-November2017V1.0-CLEAN.pdf>

Plagiarism detection in a computer program is a simple process and experienced programmers can easily see if two programs are "modified" versions of the same one.

Question:

For this assignment, you will be building on this code:

<https://repl.it/@lauramaye/participant>

This code prints out a different message, depending on whether the characters 'a', 'b', 'c', or 'd' are pressed. To build on this code, you will be creating a simple gambling game, that is intended to log and display participant character press behavior. You will be creating a class called Participant. In this game:

- Each participant will start with 2,000 euro
- A participant can choose between four characters to input: 'a', 'b', 'c', 'd'. Each time one of these characters is pressed, the participant a) wins a certain amount of money and b) loses a certain amount of money.
- For this assignment, we will assume we only have one participant. We will also be hardcoding the win amount and lose amount (as a tuple) every time the participant presses either 'a', 'b', 'c', or 'd' on the keyboard (see instructions below).

Creating a Participant class and using it	
A	<p>The Participant class contains the following variables. You should decide whether they should be defined as a class variable or an instance variable.</p> <ul style="list-style-type: none">(i) The participant's first name (which is passed as an argument when the object is instantiated)(ii) The participant's last name (which is passed as an argument when the object is instantiated)(iii) A variable that counts how many instances of the Participant class have been created(iv) The participant's number (which is set automatically as new participant objects are created. The first participant's number is 1, the second participant's number is 2, the third participant's number is 3, etc.)(v) The participant's winnings (which is initially set to 2000 upon object instantiation. This variable should be private.)(vi) A dictionary containing the characters the participant presses in the trial as keys, followed by the amount of times the participant has pressed the characters as items. This dictionary should be private.
B	<p>You should ensure that when the object is printed to the screen, it contains reader-friendly information concerning the object's instance. The read friendly information should include the participant's full name and number.</p>
C	<p>Create an instance of the Participant object in an appropriate place in your code.</p>
	<p>In addition to the above, the Participant class should also contain the following functions:</p>
D	<p>getFullName(self)</p> <p>This function returns the <u>full name</u> of the participant.</p> <p>Invoke this function in an appropriate place in your code.</p>

E	<p>setFullName(self, new_full_name)</p> <p>This function sets the name of the participant. Keep in mind that this function passes in the full name of the participant (which is separated by a space). You will need to add code to separate the first and last name appropriately to store the first and last name separately, as above.</p> <p>Invoke this function to test that it works.</p>
F	<p>getWinnings(self)</p> <p>This function returns the amount won by the participant when the participant presses the appropriate character ('a', 'b', 'c', 'd').</p> <p>This function should be invoked every time the user presses either one of those characters ('a', 'b', 'c', 'd').</p>
G	<p>setWinnings(self, winnings_loses)</p> <p>This function accepts one argument: a tuple representing the amount won by the participant and the amount lost by the participant when a either the 'a', 'b', 'c', 'd' characters are pressed. The format of the tuple passed in should be as follows:</p> <p>(amount_won, amount_lost)</p> <p>From this information, the setWinnings() function should calculate how much money the participant has left over once the winning amount and losing amount has been applied to the participant's overall winnings.</p> <p>This function should be invoked whenever the characters 'a', 'b', 'c' or 'd' are pressed in the main code.</p> <p>For the purpose of testing this function, you can hardcode the tuple pair and pass it in to the function. For example, you can call the following function for p1 to set winnings for 100 euro and losings at 50 euro:</p> <p>p1.setWinnings((100,50))</p>
H	<p>getKeyPressInfo(self)</p> <p>This function returns the character presses made by the participant as a list of tuples (i.e. not a dictionary). For example, if the dictionary storing all of the character counts contained the following information::</p> <p>{ 'a': 7, 'b': 5, 'c': 0, 'd':8}</p> <p>Then invoking this function should return:</p> <p>[('a',7),('b',5),('c',0),('d',8)]</p> <p>Invoke this function every time a user presses any character on the keyboard.</p>

I	recordKeyPress(<i>self</i>, char_pressed) <p>This function is intended to count how many times a participant has clicked the characters a, b, c or d. These values should be stored in the dictionary. You should take into account that a user might press other characters (e.g. 'j','k') and plan writing the code in this function accordingly.</p> <p>An appropriate message should appear on the screen if the user has not input the correct characters (i.e. 'a', 'b', 'c', 'd')</p> <p>Invoke this function every time a user inputs a character on the keyboard.</p>
J	getMaxKeyPress(<i>self</i>) <p>This function returns a list of tuples that hold the character(s) with the highest amount of key presses. For example, if the dictionary containing the character counts contained the following:</p> <pre>{ 'a': 7, 'b': 5, 'c': 0, 'd':8}</pre> <p>Then running this function should return the following tuple list:</p> <pre>[('d',8)]</pre> <p>Keep in mind that the function should also account that several characters could have the highest character press count. For example, running the function on the following dictionary:</p> <pre>{'a': 7, 'b': 5, 'c': 8, 'd':8}</pre> <p>Should return the following list of tuples:</p> <pre>[('c':8),('d',8)]</pre> <p>Note that you should not use the max() function for solving this question.</p> <p>This function should be invoked after every character press.</p>
K	getMinKeyPress(<i>self</i>) <p>This function returns a list of tuples that hold the character(s) with the lowest amount of key presses. For example, if the dictionary containing the character counts contained the following:</p> <pre>{ 'a': 7, 'b': 5, 'c': 0, 'd':8}</pre> <p>Then running this function should return the following tuple list:</p> <pre>[('c',8)]</pre> <p>Keep in mind that the function should also account that several characters could have the lowest character press count. For example, running the function on the following dictionary:</p>

	<pre>{'a': 7, 'b': 0, 'c': 0, 'd':8}</pre> <p>Should return the following list of tuples:</p> <pre>[('b':0), ('c',0)]</pre> <p>Note that you should not use the min() function for solving this question.</p> <p>This function should be invoked after every character press.</p>
L	<p>Where appropriate, you should create properties for your getter and setter methods. In your code, replace your function calls to those getter and setter methods with the property variable you created.</p>