OLLSCOIL NA hÉireann, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
University College, Cork

2016/2017

Semester 1 - Winter 2016

CS2513 Intermediate Programming

Dr. Helen Purchase
Professor Cormac Sreenan
Mr. Cathal Hoare

INSTRUCTIONS: Answer **all** questions for full marks. Questions 1 and 2 are worth 25 marks each, while Question 3 is worth 20. There is no need to add comments.

1.5 Hours

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

PLEASE ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

Question 1. (25 marks)

Question 1.a. (6 marks)

Explain the concept of *inheritance*. What do this term mean? From the perspective of a software development team, give two advantages of inheritance.

Question 1.b. (9 marks)

Provide an implementation of a class called *Square* that models the attributes of a square. The constructer for the class takes one attribute, *length of side*; a setter should be provided for this attribute. The class should provide three getters, one each for *length of side*, *length of perimeter* and *area*; the values for perimeter and area should be derived from *length of side*. Provide a __str__ method to report on the state of the object. Declare properties for length of side (getter and setter), and length of perimeter and area (getters only).

Provide a segment of code that declares an instance of the class (there is no need to provide a full test block).

Question 1.c. (10 marks)

Provide an implementation of a class called ColouredSquare that extends the Square class. The class should take an extra parameter, *colour*, containing a string that describes the colour of a square. Provide a property declaration for *colour*. Provide suitable constructor, getter/setter and *__str__* methods to report the state of the class. In addition, provide a method that overloads the '==' operator. This should return true or false based on the equivalence of the two squares; two squares are considered equivalent if they have the same length of side and colour. There is no need to provide a test block.

Question 2. (25 marks)

Question 2.a. (4 marks)

Briefly, explain how can your code might recover from an exception? Name two ways that can trigger an exception in your code.

Question 2.b. (6 marks)

From the perspective of Graphical User Interface (GUI) development, MVC stands for the Model View Controller design pattern. Describe this pattern, explaining the purpose of each of its three components.

Question 2.c. (15 marks)

Write a Python program that presents a graphical user interface. It should contain a label, entry and submit button that allows entry of a length of side value. This value will be used to create an instance of an *AnotherSquare* class (there is no need to implement the AnotherSquare class here). The interface should then update a second label with the area of the square; this label should be positioned under the other entry field and submit button.

An example showing the usage of the AnotherSquare class is shown below:

```
square = AnotherSquare(10) #Instantiates an instance of AnotherSquare
print("Area of the square is: %d" % (square.area)) #Retrieves the area of AnotherSquare instance
```

Question 3. (20 marks)

Question 3.a. (4 marks)

Using a regular expression, explain how you would write code to print "CAPITAL" if and only if an input string starts with an uppercase letter. Explain each step of your process, and illustrate with segments of code.

Question 3.b. (6 marks)

When developing a module, we can add the following lines of code:

```
if __name__ == "__main__":
    #do something
```

Under what circumstances does *__name__* have the value "*__main__*"? Under what circumstances does __name__ contain a value other than "*__main__*"? What is the significance of this difference?

Question 3.c. (10 marks)

Assume that you have a module called *anothersquare.py* containing the class *AnotherSquare* (there is no need to implement AnotherSquare here). Further, assume that you have created and run a script that created and stored an instance of the AnotherSquare class on a shelve called ("mysquaredb") using the key "square1". Write a Python program that updates this stored instance to have length of side 10 and re-saves it to the same shelve.

An example showing the usage of the AnotherSquare class is shown below:

```
square = AnotherSquare(10) #Instantiates an instance of AnotherSquare
print("Len of side the square is: %d" % (square.length_of_side)) #Print the length of side

square.length_of_side = 5 #Set the length of side
print("Side: %d" % (square.length_of_side)) #Print updated value
```