CS2513 Semester 1 2016/17
Class Exam

50 Minutes

Instructions: Answer all questions for full marks (10). Question 1 is worth 6 marks (each part is worth 2 marks), and Question 2 is worth 4 marks.

Question 1. (6 marks)
Part a. (2 marks)
From the perspective of a development team, what are two benefits of Object Oriented Programming? Briefly, explain your answer.

Part b. (2 marks)
Briefly, explain the relationship between a class and an object.

Part c. (2 marks)
Referring to the SuperHero class, described in Appendix 1, provide a fragment of code that defines a property for the *name* instance variable (you may use either style of property declaration). Give one advantage of defining properties for instance variable access.

Question 2. (4 marks)
Implement a Python class called Superman. The Superman class inherits from the SuperHero class (described on the next page in Appendix 1).

The SuperHero class is described by a name, which represents the name of the super-hero, and morality, which is a boolean that represents whether the super-hero is a goody or a baddy. The SuperHero class also implements a method called defendYourSelf() which prints a message "defending by using my superpowers".

The Superman class should provide a state to indicate if Superman is flying. The initial value for this state is false. The Superman class should provide a specialisation of the defendYourSelf() method that prints "Zapping you with my x-ray vision". The class should be able to describe its state.

Provide an implementation of the Superman class, including constructor, getters/setters and other specified behaviour. Also provide a fragment of code that creates an instance of Superman, changes his flying state, prints his state and calls his defendYourSelf method.

There is no need to declare properties for the instance variable. There is no need to provide comments - except where you feel they would provide clarity.

Appendix 1. The SuperHero Class.

```
class SuperHero():
    def __init__(self, name, morality):
        self._name = name
        self._morality = morality

    def getName(self):
        return self._name

    def setName(self, name):
        self._name = name

    def getMorality(self, morality):
        return self._morality

    def setMorality(self, morality):
        self._morality = morality

    def defendYourSelf(self):
        print("Defending using super powers")

    def __str__(self):
        if self._morality:
            return ("%s is a good guy" % (self._name))
        else:
            return ("%s is a bad guy" % (self._name))
```