

CSC 110 2.0 Object Oriented Programming Tutorial 09

Instructions:

- All questions must be attempted and answers submitted in a handwritten document, **on or before 8.00am on Tuesday, 22nd October 2019, to the Department Office.**
- You must indicate your **Index Number and the Tutorial Class** to which you belong to (**LCS1/ LCS2/ NFC3.1**) clearly on the front page of your submission.
- Recommended Time Duration: **30 minutes**

Coding in Java

[20 Marks]

Consider this UML class diagram.



In this exercise, Shape shall be defined as an abstract class, which contains:

- Two protected instance variables color(String) and filled(boolean). The protected variables can be accessed by its subclasses and classes in the same package. They are denoted with a '#' sign in the class diagram.
- Getter and setter for all the instance variables, and toString().

- Two abstract methods *getArea()* and *getPerimeter()* (shown in italics in the class diagram).

The subclasses Circle and Rectangle shall *override* the abstract methods *getArea()* and *getPerimeter()* and provide the proper implementation. They also *override* the *toString()*.

Write a test class to test these statements involving polymorphism and explain the outputs. Some statements may trigger compilation errors. Explain the errors, if any.

In this question, Shape shall be defined as an abstract class, which contains:

- Two protected instance variables *color(String)* and *filled(boolean)*. The protected variables can be accessed by its subclasses and classes in the same package. They are denoted with a '#' sign in the class diagram.
- Getter and setter for all the instance variables, and *toString()*.
- Two abstract methods *getArea()* and *getPerimeter()* (shown in italics in the class diagram).

The subclasses Circle and Rectangle shall *override* the abstract methods *getArea()* and *getPerimeter()* and provide the proper implementation. They also *override* the *toString()*.

Write a test class to test these statements involving polymorphism and explain the outputs. Some statements may trigger compilation errors. Explain the errors, if any.
