1005ICT Object Oriented Programming 2015-2
Laboratory 5

School of Information and Communication Technology
Griffith University
August 31, 2015

<table>
<thead>
<tr>
<th>When</th>
<th>Teaching week 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td>In this laboratory you will create a program with classes related by inheritance.</td>
</tr>
<tr>
<td>Marks</td>
<td>4</td>
</tr>
</tbody>
</table>

1 Preparation

Before your lab class:

- Print these lab notes. You need to refer to them a lot before the lab class and during it.
- Read up to section 10 of the lecture notes.
- You can start work before your lab class

2 Pre-laboratory questions (0.5 marks)

Answer the following questions in the space provided, before your laboratory class.

1. What makes an abstract class different to other classes?

2. How is an abstract class indicated in a UML class diagram?

3. Consult the Java API, and describe what %s does in a printf format string?
3 Activities

3.1 Design 1 (1 mark)

We are going to build a program a bit like last week’s rectangle tool. This time the program will only report areas (not perimeters), but it will do it for the following shapes:

- base
- height
- radius
- width
- height
- side

Draw a UML class diagram (on paper is preferred) that shows the inheritance relationship between the following classes:

- `java.lang.Object`, which defines the `toString()` method;
- a class that represents a triangle with a base and a height;
- a class that represents a rectangle with a width and a height;
- a class that represents a square with a side;
- a class that represents a circle with a radius; and
- a class that represents an unspecified `Shape`. All shapes have an `area()` function and this class will define it.

The diagram does not need to show the members of the classes. Note that:

- All classes are subclasses of `java.lang.Object`.
- A triangle is a shape.
- A rectangle is a shape.
- A square is a shape, but it is not a rectangle.
- Class `Shape` has already been implemented as an abstract class, as follows:

    /*
    ** file:  Shape.java
    ** author: Andrew Rock
    ** purpose: This class represents a shape. Every shape has an area.
    */

    public abstract class Shape {
        // area() returns the area of this shape.
        public abstract double area();
    }

It declares that the `area()` function should be a public instance method that returns a `double` and that it has no arguments. Note that the method is marked `abstract` and has no block.
3.2 Program 1 (2.5 marks)

- Implement all of the new above, as follows:
  - Each class must have private instance fields that store the specific shapes dimensions.
  - Each class must have a constructor that supplies the dimensions.
  - Each class must have an area() function.
  - Each class must have a toString() function that returns strings like these, as appropriate:
    * triangle (base = 1.00, height = 2.00)
    * rectangle (width = 1.00, height = 1.00)
    * square (side = 1.00)
    * circle (radius = 1.00)
  All numbers are rounded to two decimal places.

- Write a Main class that works like this:

  ```java
  $ java Main
  ? triangle 1.0 2.0
  Area of triangle (base = 1.00, height = 2.00) = 1.00
  ? rectangle 1.0 2.0
  Area of rectangle (width = 1.00, height = 2.00) = 2.00
  ? square 1.0
  Area of square (side = 1.00) = 1.00
  ? circle 1.0
  Area of circle (radius = 1.00) = 3.14
  ? quit
  $
  ```

  It does not need to handle bad user input.

- Hint: Your Main class will be simpler if you exploit inclusion polymorphism. Have a single variable of type Shape, instead of many variables with different types.

- Use MaSH Online Judge problem-id: 0507-shapeTool to check your answer.

3.3 Program 2 (No marks, just kudos)

- Add pentagons (with a side) to your tool.

4 After the Laboratory

- Organise the work you have done into folders on your network drive.

- Please answer these feedback questions.
  - What was the most difficult aspect of this laboratory?
  - Did you find an error in these lab notes?