# Hands-on Experiment # 12: Worksheet

Section\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No more than 3 students per one submission of this worksheet.

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part A: Getting Familiar with Problem (Do not code here)

In this lab, we aim to write a program to draw many geometric shapes (Square and Triangle) using standards keyboard characters. In order to draw a figure, there are 3 input parameters: character, the number of rows (size), and indent. Assume *row* is 5,

Assume character is “#”, indent is 0, and the number of rows is 5.

|  |  |
| --- | --- |
| #####  #####  #####  #####  ##### | ######  #######  ######## ­#########  ########## |
| Square | Triangle |

If character is “\*”, indent is 4, and the number of rows is 5.

|  |  |
| --- | --- |
| ####\*\*\*\*\* ####\*\*\*\*\* ####\*\*\*\*\* ####\*\*\*\*\* ####\*\*\*\*\* | ########\*#  #######\*#\*#  ######\*#\*#\*#  #####\*#\*#\*#\*#  ####\*#\*#\*#\*#\*# |
| Square | Triangle |

If the triangle is vertically flipped,

|  |
| --- |
| ##########  #########  ######## #######  ###### |
| Triangle |

Assume the size is 4 rows using a character ‘\*’, **draw** the following shapes and **compute** their perimeters and areas.

|  |  |  |
| --- | --- | --- |
|  | Square | Triangle |
| Draw |  |  |
| Area |  |  |
| Perimeter |  |  |

**Draw** the above triangle when it is vertically flipped.

|  |  |
| --- | --- |
|  | Triangle (Vertical Flip) |
| Draw |  |

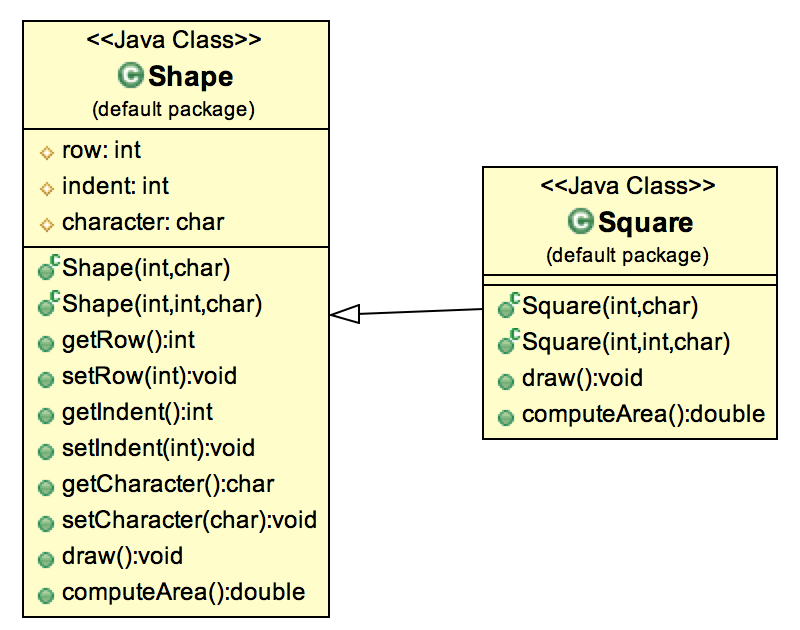
Assume the indent is 3, draw the above flipped triangle.

|  |  |
| --- | --- |
|  | Square |
| Draw |  |

## Part B: Design Your Class (Do not code here)

The below figure shows a part of the program: Shape and Square. Shape is a superclass of any shapes and there are 3 *protected* variables (row, indent, and character).

Note that the “protected” modifier can be represented by  or “#”, the “public” modifier can be represented by  or “+”. Each constructor can be represented by “C as superscript” .



Class “Shape”

* There are three properties (variables): *row, indent, character*
* There are 2 constructors.
* There are getter & setter methods for all properties (variables).
* **The draw method shows a message “Do not know what shape to draw yet!”. The computeArea() method always returns 0 as a default value.**

Class “Square”

* There are 2 constructors.
* draw(): to draw a square
* getArea(): to compute area of the object.

Write UML diagram of all shapes including: Shape, Square, Triangle

* In Triangle, there is a variable called “isVerticallyFlipped”. If it is true, the figure is vertically flipped.
  + In order to get and set this variable, there are 2 extra methods: boolean isVerticallyFlipped() and void setVerticallyFlipped(boolean isVerticallyFlipped).

What is the parent class of Shape?

## Part C: Implement Your Classes

Implement all classes based on your design in Part B and copy your implemented classes to the provided space below. Note that TestDraw1.java, TestDraw2.java, and TestDraw3.java are given.

Source code for class Shape, Square, Triangle

What statement the you have to put in the constructor of Square and Triangle in order to make the class compiled without error? Why those statements are required?

## Part D: Test Your Classes

1) What is the result of TestDraw1.java (code below)?

|  |
| --- |
| **public** **class** TestDraw1 {  **public** **static** **void** main(String[] args) {  Shape shape = **new** Shape(7, 2, '\*');  System.***out***.println(shape.computeArea());  shape.draw();  }  } |

Include the screenshots below.

2) What is the result of TestDraw2.java (code below)?

|  |
| --- |
| **public** **class** TestDraw2 {  **public** **static** **void** main(String[] args) {  Triangle triangle1 = **new** Triangle(7, 2, '\*');  System.***out***.println(triangle1.computeArea());  triangle1.draw();  }  } |

Include the screenshots below.

3) What is the result of TestDraw3.java (code below)?

|  |
| --- |
| **public** **class** TestDraw3 {  **public** **static** **void** main(String[] args) {  Shape triangle1 = **new** Triangle(7, 2, '\*');  System.***out***.println(triangle1.computeArea());  triangle1.draw();  }  } |

Include the screenshots below.

What is the property name that allows you to assign subclass to superclass (on the highlight line in the code above)?

When statement

Shape triangle1 = new Triangle(7, 2, ‘\*’);

Is executed, what constructors need to be called and in what order? (Try to figure out without using System.out.println() in the constructor.)

Run the statement in the previous question with System.out.println() in the constructor, or use “Java Visualizer” to see the execution order. Capture the screen the show the result of execution order. Why the execution is in that order?

## Part E: Apply Implemented Classes

Implement ShapeCollection.java (an initial code is given) to draw “an up and down arrow” along with the total area as shown in the figure below.

|  |  |
| --- | --- |
| The total area is 74.0  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \* \* \*  &&&&&  &&&&&  &&&&&  &&&&&  &&&&&  \* \* \* \* \* \* \*  \* \* \* \* \* \*  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* | To draw this picture, they are comprised of 3 shapes:   1. Triangle with 7 rows and no indent 2. Square with 5 rows and 4 spaces of indent 3. Vertically Flipped Triangle of the first shape   In the class, you must create ArrayList to collect all the shapes. Also, there are 2 static methods including   * public static double computeArea(ArrayList<Shape> shapeList) * public static void draw(ArrayList<Shape> shapeList) |

Include the screenshots below.

List all your source code here.

## Part F: Have fun with coding (Optional for today but will be mandatory later)

Install Robocode from **robocode-1.9.3.0-setup.jar**. If you do not know how to install using .jar file, go to <http://robocode.sourceforge.net>, and read the instruction how to install

Read Robocode API for the class robocode.Robot.

1. What is the purpose of class Robot?
2. Assume that we have a reference name robot1 refers to an instance of a robot, what method(s) that we can use to turn the robot head to left wall. Write a statement that will turn the robot head to left wall.
3. From the previous question, if your robot moving to the left wall, how can your robot moves to the opposite direction when hitting the wall?

Try to run robocode with some example robots in the battle. Have fun!

Read <http://robowiki.net/wiki/Robocode/My_First_Robot> and try to create you own simple robot. Later, you will be ask to implement a robot that can do some tasks like move around the field to cover all area of the field in some pattern (similar to cleaning robot that has many cleaning patterns), or battle with some example robot, ajarn’s robot, or your friend’s robots.

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 12) before noon of the day after your lecture.