# Hands-on Experiment # 12: Worksheet

Section\_\_\_1\_\_\_\_\_\_\_\_\_\_ Date\_\_\_23 Apr 2019\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Student ID \_\_\_6131874021\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_Sanphat Chanthanuraks\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_6131835621\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_Patthanat Thanintantrakun\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_6131898121\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_Anon Durongpisitkul\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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, RightTriangle, Triangle) using standards keyboard characters. In order to draw a figure, there are 2 input parameters: character and the number of rows. Assume *rows* is 5,

* For Square, the number of characters in each row and column must be 5.
* For RightTriangle and Triangle, the number of characters is increased by 1 every row (up to 5).

|  |  |  |
| --- | --- | --- |
| \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* | % %%  %%%  %%%% %%%%% | #  # #  # # #  # # # #  # # # # # |
| Square | RightTriangle | Triangle |

Assume the size is 6 rows using a character ‘\*’, **draw** the following shapes and **compute** their perimeters and areas.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Square | RightTriangle | Triangle |
| Draw | \*\*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\* | \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*  \*\*\*\*\*\* | \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \* |
| Perimeter | 24 | 20.49 | 19.4 |
| Area | 36 | 18 | 18 |

**Draw** the above RightTriangle when it is vertical flip and draw the above Triangle when it is horizontal flip.

|  |  |  |
| --- | --- | --- |
|  | RightTriangle (Vertical Flip) | Triangle (Horizontal Flip) |
| Draw | \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*  \*\*\*\*\*\* | \* \* \* \* \* \*  \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* |

Assume we can draw each shape at a position (x, y), where x is an indent (the number of spaces) and y is the starting row. Please draw a rectangle at the position (5, 2) when *rows*=6 and *character*=’\*’. From this example, there are 5 indents (x) and the starting row is 2 (y).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Square | RightTriangle | Triangle |
| 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 2 *protected* variables (rows and character) – represented by the “#” symbol.



Class “Shape”

* There are two properties (variables): *rows* and *character*
* There are 2 constructors.
* There are getter & setter methods for all properties (variables).
* toString() shows all variables’ value; e.g., “rows=5 and character=\*”

Class “Square”

* There are 2 constructors.
* draw(): to draw a square without indent and starting row.
* draw(int x, int y): to draw a square with *x* indents and starting row at *y*.
* getArea() and getPerimeter() to compute area and perimeter of the object.
* toString() shows object’s information; e.g., “Square: rows=5 and character=\*”.

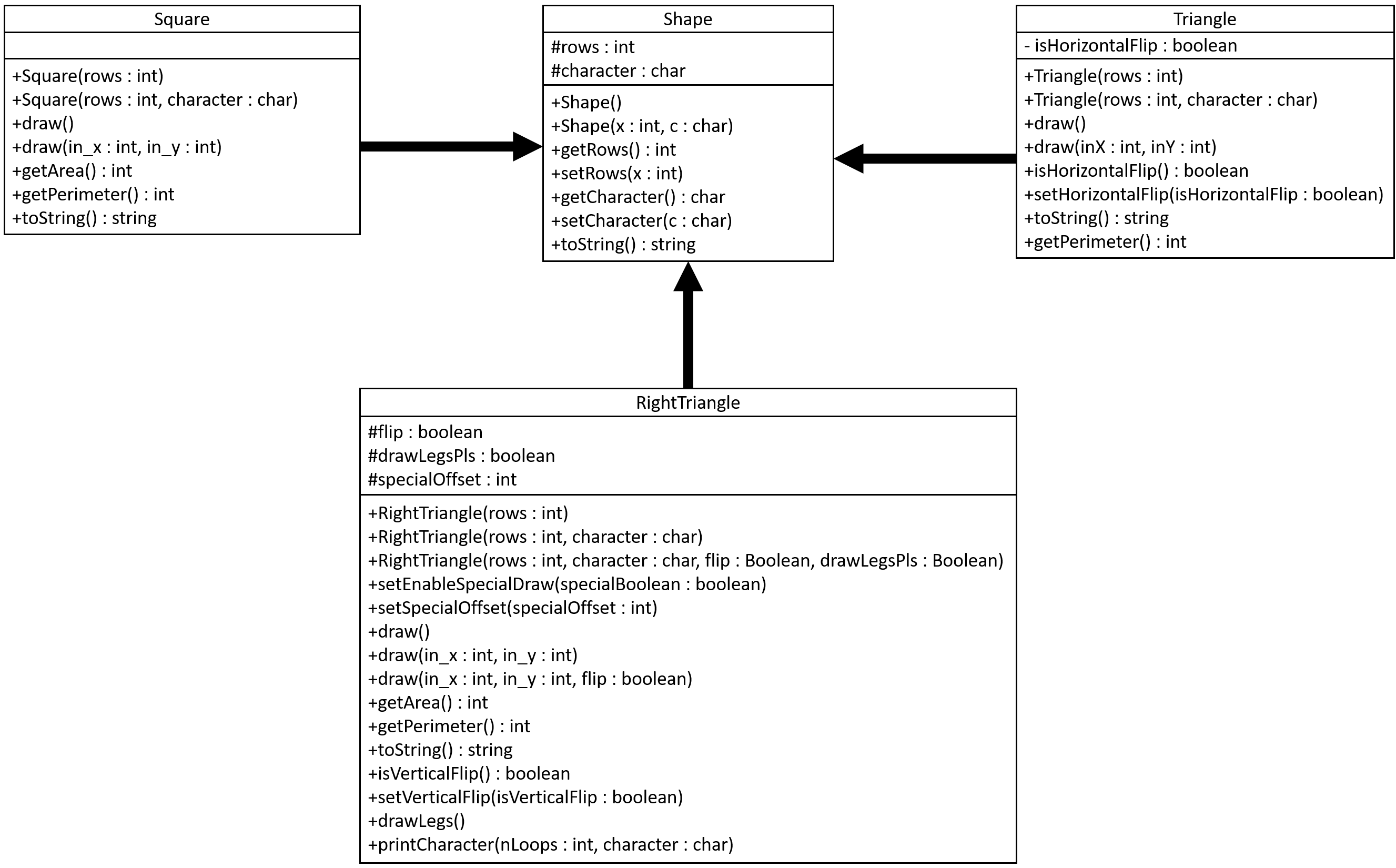
If the variables (*rows* and *character*) in Shape are *private*, can the following code inside Square still be able to compile? If not, why?

|  |
| --- |
| // Inside the Square class  **public** **void** test1(){  **int** side = rows;  } |

No, since the variables are private, those variables can only be accessed in the Shape class.

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

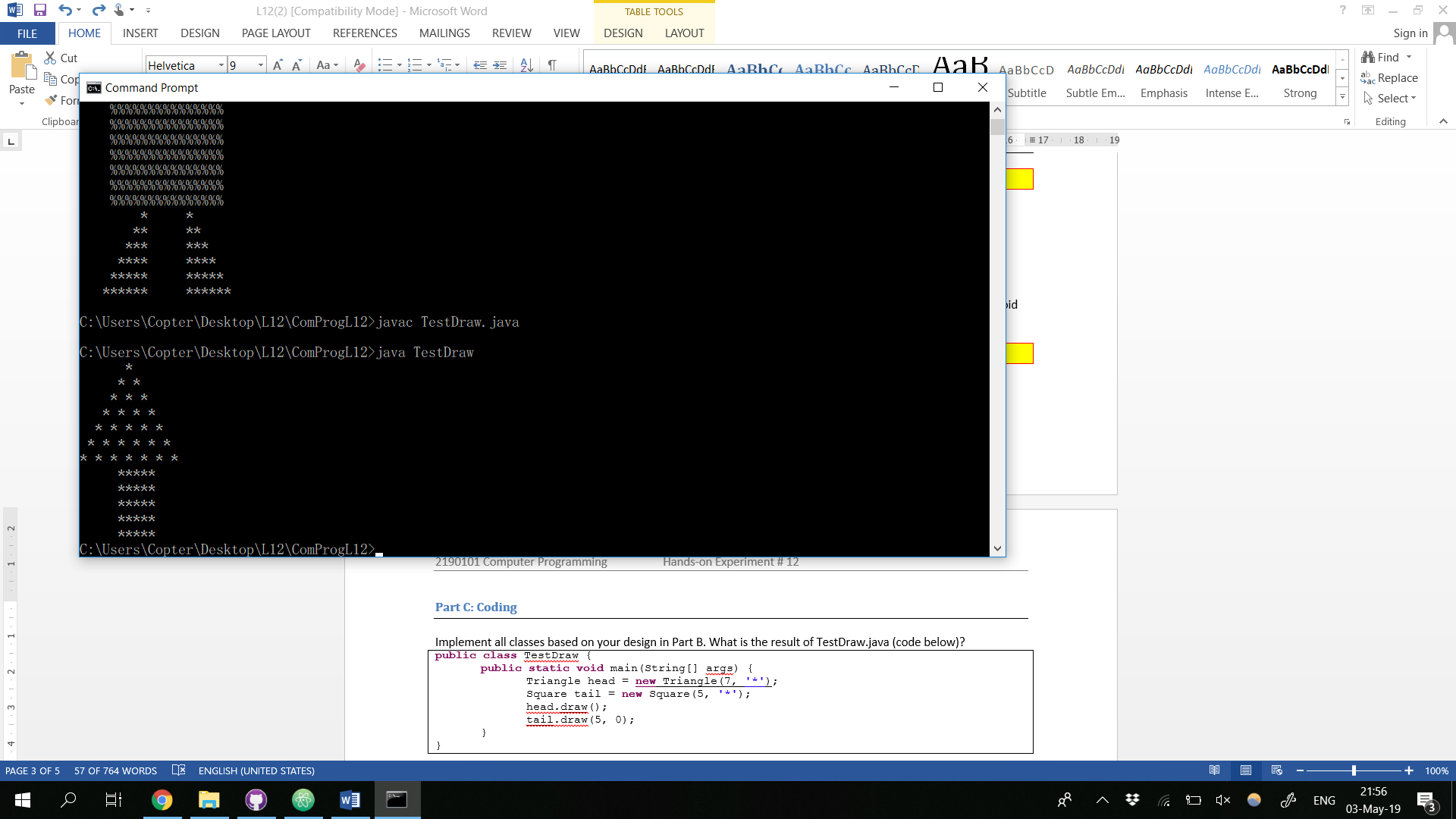
* In Triangle, there is a variable called “isHorizontalFlip”. If it is true, the figure is horizontal flipped.
  + In order to get and set this variable, there are 2 extra methods: boolean isHorizontalFlip() and void setHorizontalFlip(boolean isHorizontalFlip).
* In RightTriangle, there is a variable called “isVerticalFlip”. If it is true, the figure is vertical flipped.
  + In order to get and set this variable, there are 2 extra methods: boolean isVerticalFlip() and void setVerticalFlip(boolean is VerticalFlip).



## Part C: Coding

Implement all classes based on your design in Part B. What is the result of TestDraw.java (code below)?

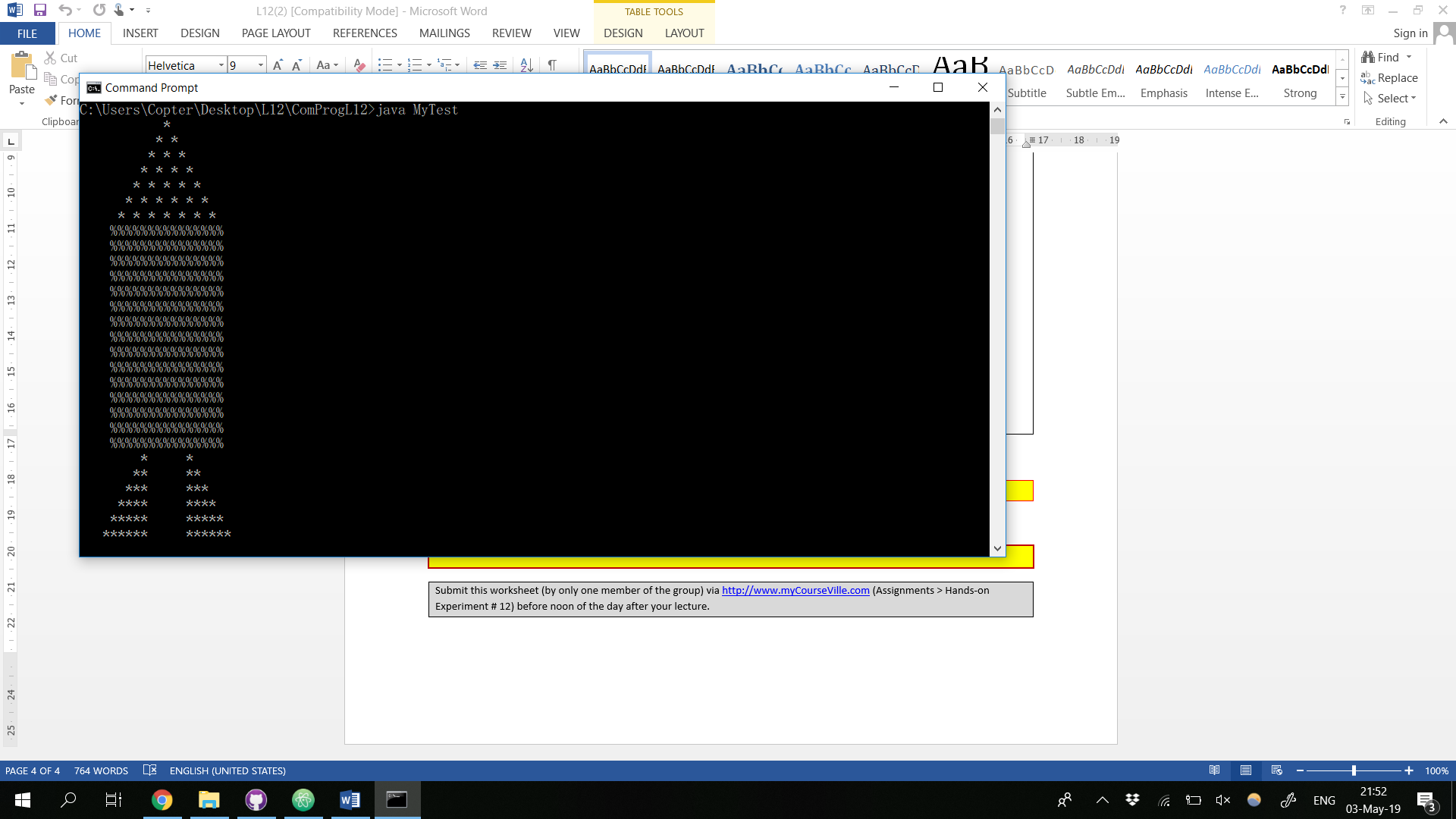
|  |
| --- |
| **public** **class** TestDraw {  **public** **static** **void** main(String[] args) {  Triangle head = **new** Triangle(7, '\*');  Square tail = **new** Square(5, '\*');  head.draw();  tail.draw(5, 0);  }  } |



Modify TestDraw.java to draw the following figure.

|  |
| --- |
| #  # #  # # #  # # # #  # # # # #  # # # # # #  # # # # # # #  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  %%%%%%%%%%%%%%%  & &  && &&  &&& &&&  &&&& &&&&  &&&&& &&&&&  &&&&&& &&&&&&  &&&&&&& &&&&&&& |

Include the screenshots below.



List all your source code here.

Shape.java

public class Shape{

protected int rows;

protected char character;

public Shape(){

this.rows = 0;

this.character = ' ';

}

public Shape(int x, char c){

this.rows = x;

this.character = c;

}

public int getRows(){

return this.rows;

}

public char getCharacter(){

return this.character;

}

public void setRows(int x){

this.rows = x;

}

public void setCharacter(char c){

this.character = c;

}

public String toString(){

return "Rows=" + rows + " Character=" + character;

}

}

Square.java

public class Square extends Shape{

public Square(int rows){

super(rows, '\*');

}

public Square(int rows, char character){

super(rows, character);

}

public void draw(){

// for(int i=0; i<this.rows; i++){

// for(int j=0; j<this.rows; j++){

// System.out.print(this.character);

// }

// System.out.print("\n");

// }

draw(0, 0);

}

public void draw(int in\_x, int in\_y){

for(int y0=0; y0<in\_y; y0++){

System.out.print("\n");

}

for(int i=0; i<this.rows; i++){

for(int x0=0; x0<in\_x; x0++){

System.out.print(" ");

}

for(int j=0; j<this.rows; j++){

System.out.print(this.character);

}

if (i != this.rows - 1) {

System.out.print("\n");

}

}

}

public int getArea(){

return this.rows\*this.rows;

}

public int getPerimeter(){

return 4\*this.rows;

}

public String toString() {

return "Rows: " + rows + ", Character: " + character;

}

}

Triangle.java

public class Triangle extends Shape {

private boolean isHorizontalFlip = false;

// private int rows = 0;

// private char character = "\*";

public Triangle(int rows){

super(rows, ' ');

}

public Triangle(int rows, char character){

super(rows, character);

}

public void draw() {

draw(0,0);

}

public void draw(int inX, int inY) {

int rows = super.rows;

char character = super.character;

/\*\*

\* # -> offset = 6

\* # # -> = 5

\* # # #

\* # # # #

\* # # # # #

# # # # # #

# # # # # # #

\*/

if (isHorizontalFlip) {

for(int i = 0; i < inY; i++) {

System.out.println("");

}

for (int row = rows; row > 0; row--) {

int leftOffset = (rows) - row + inX;

// Print offset

for (int offset = 0; offset < leftOffset; offset++) {

System.out.print(" ");

}

for (int r = 0; r <= row -1; r++) {

System.out.print(character + " ");

}

System.out.println();

}

} else {

for (int row = 0; row < rows; row++) {

int leftOffset = (rows - 1) - row + inX;

// Print offset

for (int offset = 0; offset < leftOffset; offset++) {

System.out.print(" ");

}

for (int r = 0; r <= row; r++) {

System.out.print(character + " ");

}

System.out.println();

}

}

}

public boolean isHorizontalFlip() {

return isHorizontalFlip;

}

public void setHorizontalFlip(boolean isHorizontalFlip) {

this.isHorizontalFlip = isHorizontalFlip;

}

public String toString() {

return "Rows: " + rows + ", Character: " + character;

}

public int getArea() {

return (rows \* (rows/2));

}

public int getPerimeter() {

return 3\*rows;

}

}

RightTriangle.java

public class RightTriangle extends Shape{

protected boolean flip;

protected boolean drawLegsPls;

protected int specialOffset = 0;

public RightTriangle(int rows){

super(rows, '\*');

this.flip = false;

}

public RightTriangle(int rows, char character){

super(rows, character);

this.flip = false;

}

public RightTriangle(int rows, char character, boolean flip, boolean drawLegsPls){

super(rows, character);

this.flip = flip;

this.drawLegsPls = drawLegsPls;

}

public void setEnableSpecialDraw(boolean specialBoolean) {

this.drawLegsPls = specialBoolean;

}

public void setSpecialOffset(int specialOffset) {

this.specialOffset = specialOffset;

}

public void draw(){

if (this.drawLegsPls) {

this.drawLegs();

return;

}

if(this.flip){

for(int i=0; i<this.rows; i++){

for(int blank=this.rows-i; blank>1; blank--){

System.out.print(" ");

}

for(int j=0; j<i+1; j++){

System.out.print(this.character);

}

System.out.print("\n");

}

}

else{

for(int i=0; i<this.rows; i++){

for(int j=0; j<i+1; j++){

System.out.print(this.character);

}

System.out.print("\n");

}

}

}

public void draw(int in\_x, int in\_y){

if (this.drawLegsPls) {

this.drawLegs();

return;

}

if(this.flip){

for(int y0=0; y0<in\_y; y0++){

System.out.print("\n");

}

for(int i=0; i<this.rows; i++){

for(int x0=0; x0<in\_x; x0++){

System.out.print(" ");

}

for(int blank=this.rows-i; blank>1; blank--){

System.out.print(" ");

}

for(int j=0; j<i+1; j++){

System.out.print(this.character);

}

System.out.print("\n");

}

}

else{

for(int y0=0; y0<in\_y; y0++){

System.out.print("\n");

}

for(int i=0; i<this.rows; i++){

for(int x0=0; x0<in\_x; x0++){

System.out.print(" ");

}

for(int j=0; j<i+1; j++){

System.out.print(this.character);

}

System.out.print("\n");

}

}

}

public void draw(int in\_x, int in\_y, boolean flip){

if (this.drawLegsPls) {

this.drawLegs();

return;

}

if(flip){

for(int y0=0; y0<in\_y; y0++){

System.out.print("\n");

}

for(int i=0; i<this.rows; i++){

for(int x0=0; x0<in\_x; x0++){

System.out.print(" ");

}

for(int j=0; j<i+1; j++){

System.out.print(this.character);

}

System.out.print("\n");

}

}

else draw();

}

public int getArea(){

return this.rows\*this.rows/2;

}

public int getPerimeter(){

return 4\*this.rows;

}

public String toString() {

return "Rows: " + rows + ", Character: " + character;

}

public boolean isVerticalFlip(){

return this.flip;

}

public void setVerticalFlip(boolean isVerticalFlip){

this.flip = isVerticalFlip;

}

public void drawLegs() {

int nRows = this.rows;

int centerOffset = 5;

char character = this.character;

for (int row = 0; row < nRows; row++) {

int leftOffset = (nRows - 1) - row;

this.printCharacter(this.specialOffset, ' ');

this.printCharacter(leftOffset, ' ');

this.printCharacter(row, character);

this.printCharacter(centerOffset, ' ');

this.printCharacter(row, character);

System.out.println();

}

}

public void printCharacter(int nLoops, char character) {

for (int i = 0; i < nLoops; i++) {

System.out.print(character);

}

}

}

MyTest.java

public class MyTest{

public static void main(String[] args) {

Triangle head = new Triangle(7, '\*');

Square tail = new Square(15, '%');

RightTriangle feet = new RightTriangle(7, '\*');

head.draw(5,0);

tail.draw(4, 0);

feet.setEnableSpecialDraw(true);

feet.setSpecialOffset(3);

feet.draw(14, 0);

}

}

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.