Object and Classes

1. Create a class `Rectangle` that represents a rectangular region of the plane. A rectangle should be described using four integers: two represent the coordinates of the upper left corner of the rectangle, giving its location; one for the width; and one for the height. Note that this class has nothing to do with AWT, Swing, or JavaFX – it is a class for your own use. Your rectangle should include:
   a) Appropriate constructors;
   b) A method translate() that takes two integers, `deltaX` and `deltaY`, used to translate the location of the rectangle;
   c) A method contains() that takes two integers, `xCoord` and `yCoord`, and returns true if the point given by these two values lies within the rectangle.

```
class Rectangle {
    private int width;
    private int height;
    private int x;
    privet int y;
    public Rectangle (int width; int height ...){
        this.width = width;
        this.height = height;
        this.x = x;
        this.y = y;
    }
    public void translate (int deltaX, int deltaY){
        this.x = this.x + deltaX;
        this.y = this.y + deltaY;
    }
    public boolean contains (int x, int y){
        return
```

this.x < x < this.x + this.width

0 < x - this.x - (this.x + this.width) < 0

Interfaces

2. State which of the following statements is True or False

Statement	True	False
All methods in an interface must be declared public	X	
When casting object types you take a risk of causing an exception	X	
When the compiler encounters one class inside another class, it generates an error		X
Every class in Java is descended from the Object class	X	
Instance methods can be called without creating an instance of the class.		X

3. In the box below explain why you might want to override the toString method in a class you are writing:

✓

4. Consider the interface and classes below. There is one error. In the box below list the line number which contains an error.

```
public interface A {
}
public class B implements A {
}
public class C extends B{
      public static void main(String[] args){
            A b1 = new B();
            A c1 = new C();
            B temp = b1;
            b1 = c1;
            c1 = temp;
      }
}
```

5. What is wrong with the code below. Assume that `transctCount` is inherited from `BankAccount`:

```
public class CheckingAccount extends BankAccount
{  public void deposit(double amount)
   { transctCount++;
     deposit(amount);
   }
   ……..……
}
```

6. What will be printed by the main method of class `NewCounter?`

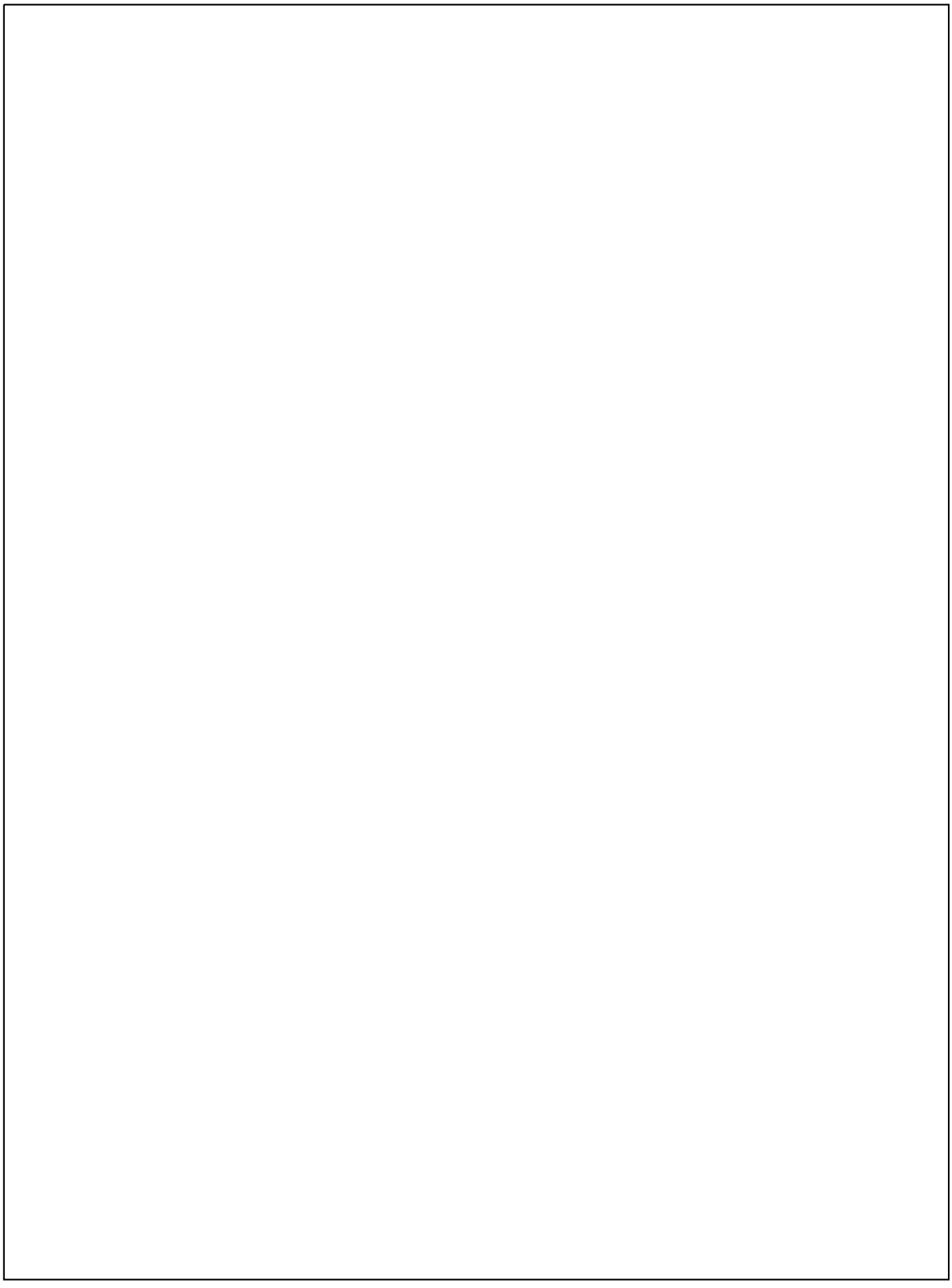
```
class Counter
{   public Counter()
    { value = 0;}
    public int get()
    { return value;}
    public void click()
    { value++;}
    private  static int value;
}

class NewCounter extends Counter {

  public static void main(String[] args)
  {  Counter c1 = new Counter();
     Counter c2 = new newCounter();
     c1.click();
     c2.click();
     c1.click();
     c2.click();
     System.out.println(c1.get() + " "+ c2.get() );
  }
}
```

GUI

7. Write an application that opens a window and draws a Rectangle (of some arbitrary size) in the mouse press position.



8. What does the following code print out:

```
public class B {
    public static void main(String[] args) {
        int a = 1;
        int b = 2;
        int c = 3;
        modify(a, b);
        modify(b, c);
        modify(c, a);
        System.out.println( a +":"+   b +":"+  c);
    }

    public static void modify(int a, int b) {
        int sum = a + b;
        a = sum;
        b = sum - a;
    }
}
```

Exceptions

9. Given the code segment below, what will be printed if no error occurs in the try block?

```
try
{
    ...
}
catch (IOException ex)
{
    System.out.println("I/O error");
}
catch (NumberFormatException ex)
{
    System.out.println("Bad input");
}
System.out.println("Done");
```

Streams

10. Write a method that reads a `String` from a `File`. Handle exceptions or throw them: