Question A: Fill in the blanks:
1. You can define a new class from an existing class. This is known as class ________________.
   The new class is called a ________________.
   The existing class is called a ________________.
2. Every class in Java is descended from the ________________ class. If no super-class is specified when a class is defined, its superclass is ________________.
3. If a method’s parameter type is a superclass (e.g., Object), you may pass an object to this method of any of the parameter’s subclasses (e.g., Circle or String).
   This is known as ________________.

Question B: True/False
1. The constructors of a superclass are not inherited in the subclass  True[ ]   False[ ]
2. A private method can be overridden if we use @Override True[ ]  False[ ]
3. The keyword this can be used to refer to the calling object. It can also be used inside a constructor to invoke another constructor of the same class True[ ]  False[ ]

Question C: Create a class MyDrawing which can draw a line between (0,0) and (20,20). Hint: This class will extend JPanel. Create a Tester class which will be used to show the line in JFrame. In the main method of tester class, create a JFrame, add object of MyDrawing. Set size of frame to be (300,300). Other details and hints given inline below.

```java
import java.awt.Graphics;
import javax.swing.JPanel;

class My_________ extends __________ {
    // Do you remember that you have to write logic to draw line in paintComponent method?
    public void paintComponent(Graphics g) {
        // call superclass method first
    }
}

class Tester {
    public static void main(String[] args) {
        // Create a JFrame object with title “My Frame”
        // Add the above object to the JFrame
    }
}
```
public class Phone {

    // Create instance variables “color” of type String and “cost” of type int

    // Create constructor
    public Phone(String color, int cost) {
        // set the instance variables
    }

    @Override
    public String toString() {
        // this method should return a string which will have both color and cost information
    }
}

public class IPhone extends _______ {

    // create an instance variable model of type String

    // create constructor that takes color, cost and model as arguments
    public IPhone(___________________________) {
        
    }

    @Override
    public String toString() {
        // this method should return a string which will have color, cost and model information
    }

    public static void main(String[] args) {
        IPhone iphone = new IPhone("Black", 600, "5S");
        System.out.println(iphone.toString());
    }
}

Q: What will be printed here?