Question A: Fill in the blanks:
1. You can define a new class from an existing class. This is known as class **inheritance**. The new class is called a **subclass**.
   The existing class is called a **superclass**, **parent class**, or **base class**.
2. Every class in Java is descended from the **java.lang.Object** class. If no super-class is specified when a class is defined, its superclass is **Object**.
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 **polymorphism**.

Question B: Answer: True, False, True
1. **Concept**: A constructor is used to construct an instance of a class. Unlike properties and methods, the constructors of a superclass are not inherited in the subclass. They can be invoked only from the constructors of the subclasses, using the keyword super.
2. **Concept**: An instance method can be overridden only if it is accessible. Thus, a private method cannot be overridden because it is not accessible outside its own class. If a method defined in a subclass is private in its superclass, the two methods are completely unrelated.
3. **Concept**: 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.

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.

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

class MyDrawing extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawLine(0,0,20,20);
    }
}

import javax.swing.*;

public class Tester {
    public static void main(String[] args) {
        JFrame frame = new JFrame("My Frame");
        frame.add(new MyDrawing());
        frame.setSize(300, 300);
        frame.setVisible(true);
    }
}
**Question D**: Create a class called Phone, create one constructor, override toString method. Now create a subclass IPhone which will extend Phone. Create its constructor. Override its toString method which will return a string containing all attributes of Phone as well as IPhone class.

```java
public class Phone {
    private String color;
    private int cost;

    public Phone(String color, int cost) {
        this.color = color;
        this.cost = cost;
    }

    @Override
    public String toString() {
        String result = "Color: " + color + " cost: "+cost ;
        return result;
    }
}
```

```java
public class IPhone extends Phone {
    private String model;

    public IPhone(String color, int cost, String model) {
        super(color, cost);
        this.model = model;
    }

    @Override
    public String toString() {
        String result = super.toString() + " model: "+model;
        return result;
    }
}
```

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

It will print: Color: Red cost: 600 model: 5S