Question 1:

What is the output of running the class C in (a)? What problem arises in compiling the program in (b)?

```java
class A {
    public A() {
        System.out.println("A's no-arg constructor is invoked");
    }
}

class B extends A {
    public C {
        public static void main(String[] args) {
            B b = new B();
        }
    }
}
```

```java
class A {
    public A(int x) {
        
    }
}

class B extends A {
    public B() {
        
    }
}

class C {
    public static void main(String[] args) {
        B b = new B();
    }
}
```

(a) (b)

Question 2:

Show the output of the following code:

```java
public class Test {
    public static void main(String[] args) {
        new Person().printPerson();
        new Student().printPerson();
    }
}

class Student extends Person {
    @Override
    public String getInfo() {
        return "Student";
    }
}

class Person {
    public String getInfo() {
        return "Person";
    }

    public void printPerson() {
        System.out.println(getInfo());
    }
}
```

```java
public class Test {
    public static void main(String[] args) {
        new Person().printPerson();
        new Student().printPerson();
    }
}

class Student extends Person {
    private String getInfo() {
        return "Student";
    }
}

class Person {
    private String getInfo() {
        return "Person";
    }

    public void printPerson() {
        System.out.println(getInfo());
    }
}
```

(a) (b)
Question – 3:

Identify the problems in the following code:

```java
public class Circle {
    private double radius;

    public Circle(double radius) {
        radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public double getArea() {
        return radius * radius * Math.PI;
    }
}

class B extends Circle {
    private double length;

    B(double radius, double length) {
        Circle(radius);
        length = length;
    }

    @Override
    public double getArea() {
        return getArea() * length;
    }
}
```
Question-4:

Suppose that Fruit, Apple, Orange, GoldenDelicious, and McIntosh are defined in the following inheritance hierarchy:

```
Fruit
  ↑
Apple  Orange
  ↑
GoldenDelicious  McIntosh
```

Assume that the following code is given:

```java
Fruit fruit = new GoldenDelicious();
Orange orange = new Orange();
```

Answer the following questions:

a. Is fruit instanceof Fruit?
b. Is fruit instanceof Orange?
c. Is fruit instanceof Apple?
d. Is fruit instanceof GoldenDelicious?
e. Is fruit instanceof McIntosh?
f. Is orange instanceof Orange?
Question-5:

(The Triangle class) Design a class named Triangle that extends GeometricObject. The class contains:

- Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
- A no-arg constructor that creates a default triangle.
- A constructor that creates a triangle with the specified side1, side2, and side3.
- The accessor methods for all three data fields.
- A method named getArea() that returns the area of this triangle.
- A method named getPerimeter() that returns the perimeter of this triangle.
- A method named toString() that returns a string description for the triangle.

For the formula to compute the area of a triangle, see Programming Exercise 2.19. The toString() method is implemented as follows:

```java
return "Triangle: side1 = " + side1 + " side2 = " + side2 + " side3 = " + side3;
```

Draw the UML diagrams for the classes Triangle and GeometricObject and implement the classes. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.

Question-6:

(Use ArrayList) Write a program that creates an ArrayList and adds a Loan object, a Date object, a string, and a Circle object to the list, and use a loop to display all the elements in the list by invoking the object's toString() method.

Question-7:

(Implement MyStack using inheritance)

Define a new stack class that extends ArrayList.

Draw the UML diagram for the classes and then implement MyStack. Write a test program that prompts the user to enter five strings and displays them in reverse order.