Quiz 7
Question: 3.

```
/*Make a class Employee with a name and salary. Make a class Manager inherit from
        Employee . Add an instance variable, named department , of type String . Supply a method
        toString that prints the manager's name, department, and salary. Make a class Execu¬tive
        inherit from Manager . Supply appropriate toString methods for all classes. Supply
        a test program that tests these classes and methods.*/


public class Employee {
   private String name;
   private double salary;


   public Employee(String name, double salary) {
      this.setName(name);
      this.setSalary(salary);
   }


   public String getName() {
      return name;
   }


   public void setName(String name) {
      this.name = name;
   }


   public double getSalary() {
      return salary;
   }


   public void setSalary(double salary) {
      this.salary = salary;
   }


   public String toString() {
```

```
            return getClass().getName() + " name: " + this.getName() + " salary: " + this.getSalary();
        }
    }
```

```
public class Manager extends Employee {
    private String department;

    public Manager(String name, double salary, String department) {
        super(name, salary);
        this.setDepartment(department);
    }

    public String getDepartment() {
        return this.department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    public String toString() {
        return super.toString() + " department: " + this.getDepartment();
    }
}
```

```
public class Executive extends Manager {
    private double bonus;

    public Executive(String name, double salary, String department, double bonus) {
```

```
        super(name, salary, department);
        setBonus(bonus);
    }

    public double getBonus() {
        return this.bonus;
    }

    public void setBonus(double bonus) {
        this.bonus = bonus;
    }

    public String toString() {
        return super.toString() + " bonus: " + this.getBonus();
    }
}
```

Question: 4

/*The Rectangle class of the standard Java library does not supply a method to com-pute the area or the perimeter of a rectangle. Provide a subclass BetterRectangle of the Rectangle class that has getPerimeter and getArea methods. Do not add any instance variables. In the constructor, call the setLocation and setSize methods of the Rectangle class. Provide a program that tests the methods that you supplied.

Repeat Exercise P9.10, but in the BetterRectangle constructor, invoke the superclass constructor.*/

```
import java.awt.Rectangle;

@SuppressWarnings("serial")
```

```
public class BetterRectangle extends Rectangle {

   public BetterRectangle(int x, int y, int width, int height) {
      super(x, y, width, height);
   }

   /*public BetterRectangle(int x, int y, int width, int height) {
      this.setLocation(x, y);
      this.setSize(width, height);
   }*/

   public double getPerimeter() {
      return 2 * super.getHeight() + 2 * super.getWidth();
   }

   public double getArea() {
      return super.getHeight() * super.getWidth();
   }
}
```

.
.
Question: 5
Ans:
It must implement the Measurable interface and provide a getMeasure method returning the salary.
এরকম আরো কিছু প্রশ্ন এবং তার উত্তর আছে এই লিঙ্কে
https://quizlet.com/196133092/java-ch9-95-96-flash-cards/
.
.
Question: 6
If both BankAccount and Coin implement the Measurable interface, can a Coin reference be converted to a BankAccount reference?
Ans:
No–a Coin reference can be converted to a Measurable reference, but if you attempt to cast that reference to a BankAccount, an exception occurs.

.
.
Question: 7
Why would you use an inner class instead of a regular class?
Ans:
Inner classes are used for logical grouping of classes. In cases where your class B is used only by class A it's better to put Class B as an inner class to Class A. This improves encapsulation and readability of the code. We have different ways to implement inner class a) static nested class b) inner class c) anonymous class.
More: https://www.quora.com/Why-do-we-need-inner-classes-in-Java