Important Instructions:

- At the end of assignment, you will able to implement OOPs concepts.
- Your code will be graded both on correctness and efficiency
- Use comments in your code that explains your assumptions and design decisions.
- You need to submit your assignment solution by end of the day
- Before submitting your assignment makes sure it as per the given requirement.
- Follow java naming conventions

Q1.

XYZ software Company is working on a Project.

You need to implement this project as per the guidelines mentioned below:

Design an interface named as Role which provides the following two methods:

- public String getRoleName();
- public String getResponsibility();

Design two classes CEO and Manager as two Roles which implement the Role interface mentioned above and provide the definition of two methods mentioned in Role interface. Design a main class which instantiate the two object one which is a CEO object another is Manager Object,

display the Role Name and Responsibility of two objects accordingly.

Q2 :-

Write a superclass Worker & subclass ContractualWorker & SalariedWorker. In order to calculate Weekly Salary.

Every worker has a name & a salary rate. Write a method computePay(int hours) that computes the weekly pay for every worker.

For ContractualWorker gets paid the hourly wage for the actual number of hours worked, if hours is at most 40. If the hourly worker worked more than 40 hrs, the excess is paid at the rate of 1.5 times of per hour salary .

The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is.

Where as SalariedWorker has a method increementSalary(int percentage), this method will increase the actual salary of the worker.

Write a method that uses polymorphism to compute the pay of any Worker.

```
3  class Worker
4  {
5      String name;
6      int noOfHoursWorked;// in a week
7      int hourlyWages ; // Per hour Rate
8
9      int computePay()
10     {
11         return 0;
12     }
13 }
14
15 class ContractualWorker extends Worker
16 {
17     // override computePay method
18 }
19
20 class SalariedWorker extends Worker
21 {
22     // override computePay method
23 }
```

Q3

Assume that a bank maintains 2 types of accounts for its customers, one is a savings account & the other is the current account.

- The savings account provides compound interest and withdrawal facilities but no cheque bounce facility.
- The current account provides cheque bounce facility but no interest.
- Current account holders should maintain a minimum balance & if the balance falls below, a service charge is imposed.

Create a class Account that store customer name, account number, and type of account.

Create two sub classes CurrentAccount & SavingAccount

Include necessary methods in order to achieve the following tasks.

1) Accept deposit from a customer & update the balance.
2) Display the balance.
3) Compute & deposit interest .
4) Withdrawal & update the balance
5) Check for the minimum balance
6) impose penalty if necessary & update the balance

Do not use constructors. Use methods to initialize the class members.