Q1.

XYZ software Company is working on a Bank Project which follows the below given design. You need to implement this project as per the guidelines mentioned below:

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

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 contracts 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 HourlyWorker & SalariedWorker. Every worker has a name & a salary rate. Write a method computePay(int hours) that computes the weekly pay for every worker. An hourly worker 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 time &half. The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is. Write a method that uses polymorphism to compute the pay of any Worker. Supply a test program that test these classes & methods

Q3.

Assume that a bank maintains 2 types of accounts for its customers, one is savings account & the other is current account. The savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should maintain a minimum balance & if the balance falls below this level, a service charge is imposed. Create a class Account that store customer name, account number, and type of account. From this derive the classes Curr-acc & Sav-acc to make them more specific to their requirements.

Include necessary methods in order to achieve the following tasks.

Accept deposit from a customer & update the balance.

Display the balance.

Compute & deposit interest

Permit withdrawal & update the balance

Check for the minimum balance, impose penalty if necessary & update the balance Do not use constructors. Use methods to initialize the class members.

Q4.

WAP which has an interface IRestaurant with a contract Dish getDish(int dishId) which needs to be implemented by Restaurant Concrete class. Restaurant class should poses a static array menu of Dish type which needs to be initialized via static block with some sample dishes. Dish class should comprise dishId, dishName,dishPrice &dishMakeTime as attributes and also override toString method to display the Dish details. Restaurant class implement the getDish method which prompts the user to enter the dishId of Dish which he/she wants to purchase. If the dish with dishId entered by the user is available in the menu array then it return that Dish object else return null. In the main method a user will be prompted to enter the dish id as per his choice and on the basis of user input a getDish method will be invoked which returns the Dish object which will be printed on console.

Q5.

Create a class named "CollegeCourse" that includes data fields that hold the department (for example,"ENG"), the course number (for example, 101), the credits (for example, 3), and the fee for the course (for example, $360). All of the fields are required as arguments to the constructor, except for the fee, which is calculated at $120 per credit hour. Include a display () method that displays the course data. Create a subclass named "LabCourse" that adds $50 to the course fee. Override the parent class display() method to indicate that the course is a lab course and to display all the data. Write an application named UseCourse that prompts the user for course information. If the user enters a class in any of the following departments, create a LabCourse: BIO, CHM, CIS, or PHY. If the user enters any other department, create a CollegeCourse that does not include the lab fee. Then display the course data. Save the files as CollegeCourse.java, LabCourse.java, and UseCourse.java.

Q6.

Write an application named UseChildren that creates and displays at least two Child objects—one Male and one Female. Child is an abstract class and Male and Female are subclasses. The Child class contains fields that hold the name, gender, and age of a child. The Child class constructor requires a name and a gender. The Child class also

contains two abstract methods named setAge()and display(). The Male and Female subclass constructors require only a name; they pass the name and appropriate gender to the Child. The subclass constructors also prompt the user for an age using the setAge() method, and display the Child's data using the display()method. Save the files as Child.java, Male.java, Female.java, and UseChildren.java.