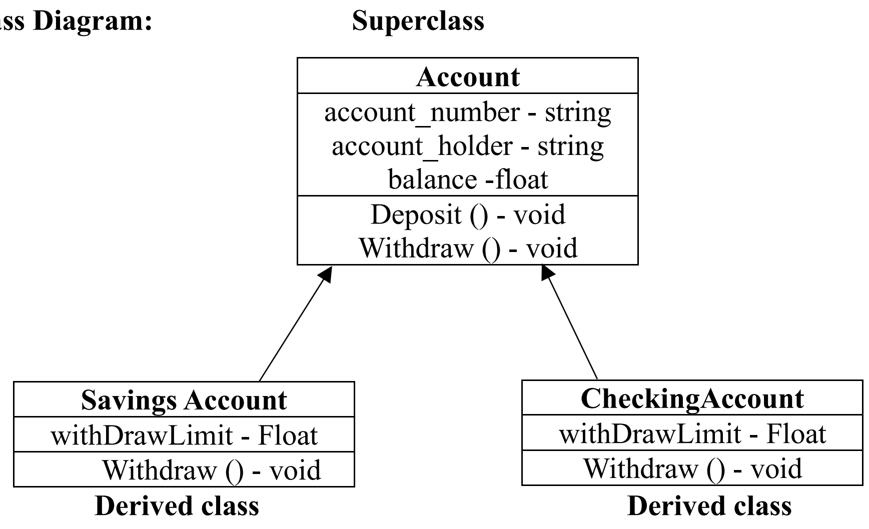
**Q1)** Design a class hierarchy for a banking system. Create a base class called Account with attributes account_number, account_holder, and balance, along with methods deposit() and withdraw(). Derive two classes Savings Account and CheckingAccount from the Account class. Implement method overloading in the withdraw() method of both derived classes to accommodate different withdrawal limits based on the account type.

**Class Diagram:**



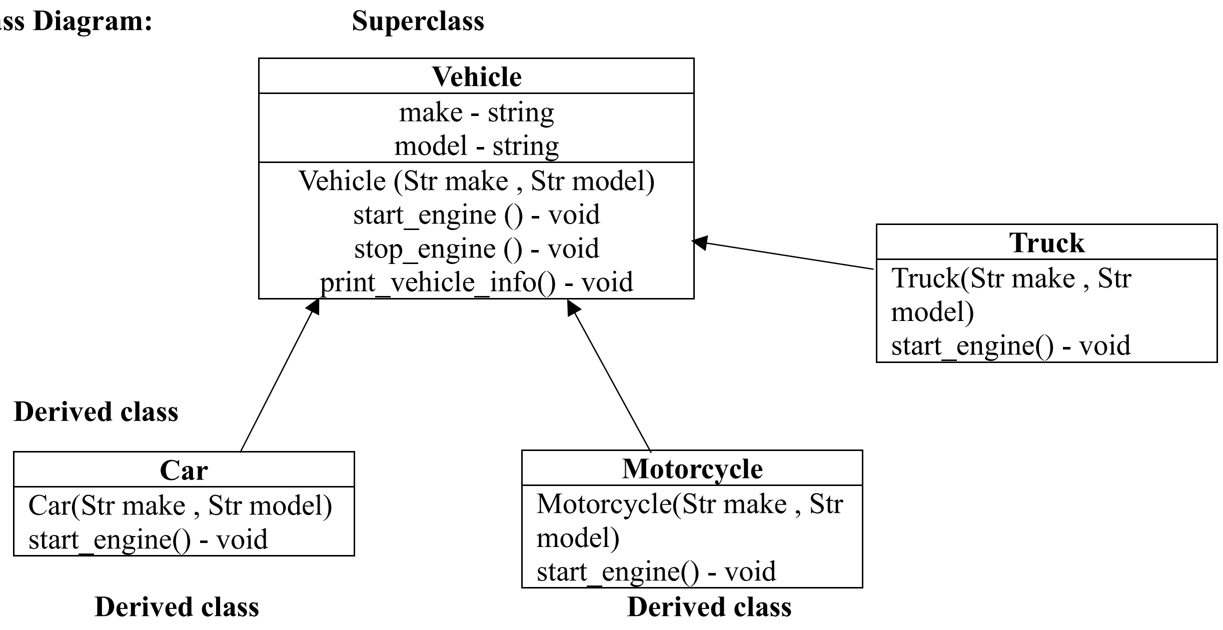
**Program:**

```
package com.lab_assessment2;

import java.util.*;


// Base class

public class Account

{


    protected String account_number;

    private String account_holder;

    protected float balance;




  // deposit method to add amount to their accounts by asking account number, holder name and amount of deposit

    public void deposit() {

        float amount;
        Scanner input = new Scanner(System.in); //creating scanner class for input by user

        System.out.print("Account Number : ");
        this.account_number = input.next();

        System.out.print("Account Holder Name : ");
        this.account_holder = input.next();

        System.out.print("Amount to deposite : ");
        amount = input.nextFloat();

        this.balance += amount;

        System.out.println("***** Deposition is Successful *****");
        System.out.println("***** AVAILABLE BALANCE : "+this.balance+"*****");
        input.close();

    }
    // method to withdraw amount from their account
    public void withdraw() {

        float amount;
        Scanner input = new Scanner(System.in);
        System.out.print("Account number : ");
        this.account_number = input.next();
        System.out.print("Withdraw Amount : ");
        amount = input.nextFloat();

        /* compare existing amount with withdrawal amount and withdraw amount based on condition
         * where entered amount is less the existing amount */
        if(amount>this.balance) {
                this.balance -= amount;
                System.out.println("* "+amount+" is successfuly withdraw *");
                System.out.println("***** AVAILABLE BALANCE : "+this.balance+"*****");
        }
        else
                // if entered amount is more than existing amount showing following error
                System.out.println("***** INSUFFICIENT BALANCE *****");
        input.close();

    }
}




package com.lab_assessment2;

import java.util.Scanner;

// Savings account(subclass) which inherit Account class(superclass)
public class SavingsAccount extends Account
{
    private float withDrawLimit = 20000; //setting withdraw amount for savings account as 20000 per month
  @Override
  // the method withdraw is redefined from superclass called Account
    public void withdraw() {
        float amount;
        Scanner input = new Scanner(System.in); //creating scanner class for input by user

        System.out.print("Account number : ");
        super.account_number = input.next();

        System.out.print("Withdraw Amount : ");
        amount = input.nextFloat();
        /* compare existing amount with withdrawal amount and withdraw amount based on condition
         * where entered amount is less the existing amount */
        if (amount>super.balance) {

                /* compare entered amount with withdrawal limit amount and withdraw amount based on
    condition
                 * where entered amount is less the withdrawal limit amount */
                if(amount<withDrawLimit) {
                        System.out.println("* "+amount+" is successfuly withdraw *");
                        System.out.println("*****    AVAILABLE    BALANCE    :    "+(super.balance-
    amount)+"*****");

                }
                else
                        // if user entered amount more than limit showing error as mentioned
                        System.out.println("### LIMIT EXCCEDED ###");
        }
        else
                // if entered amount is more than existing amount showing following error
                System.out.println("***** INSUFFICIENT BALANCE *****");
        input.close();
    }
}

package com.lab_assessment2;

import java.util.Scanner;

//checking account(subclass) which inherit Account class(superclass)
public class CheckingAccount extends Account
{
    private float withDrawLimit = 25000; //setting withdraw amount for savings account as 25000 per month
   @Override
// the method withdraw is redefined from superclass called Account
    public void withdraw() {
        float amount;
        Scanner input = new Scanner(System.in); //creating scanner class for input by user

        System.out.print("Account number : ");
        super.account_number = input.next();

        System.out.print("Withdraw Amount : ");
        amount = input.nextFloat();

        /* compare existing amount with withdrawal amount and withdraw amount based on condition
         * where entered amount is less the existing amount */
        if (amount>super.balance) {

            /* compare entered amount with withdrawal limit amount and withdraw amount based on
    condition
             * where entered amount is less the withdrawal limit amount */
            if(amount<withDrawLimit) {
                System.out.println("* "+amount+" is successfuly withdraw *");
                System.out.println("*****    AVAILABLE    BALANCE    :    "+(super.balance-
    amount)+"*****");

            }
            else
                // if user entered amount more than limit showing error as mentioned
                System.out.println("### LIMIT EXCCEDED ###");
        }
        else
            // if entered amount is more than existing amount showing following error
```

```
                    System.out.println("***** INSUFFICIENT BALANCE *****");
        input.close();
        }


}
```

**Q2)** Create a base class Vehicle with attributes make, model, and methods start_engine() and stop_engine(). Derive classes Car, Motorcycle, and Truck from the Vehicle class. Implement method overloading in the start_engine() method of the derived classes to handle different start-up procedures. Also, implement a method print_vehicle_info() that displays the make and model of each vehicle.

**Class Diagram:**



**Program:**

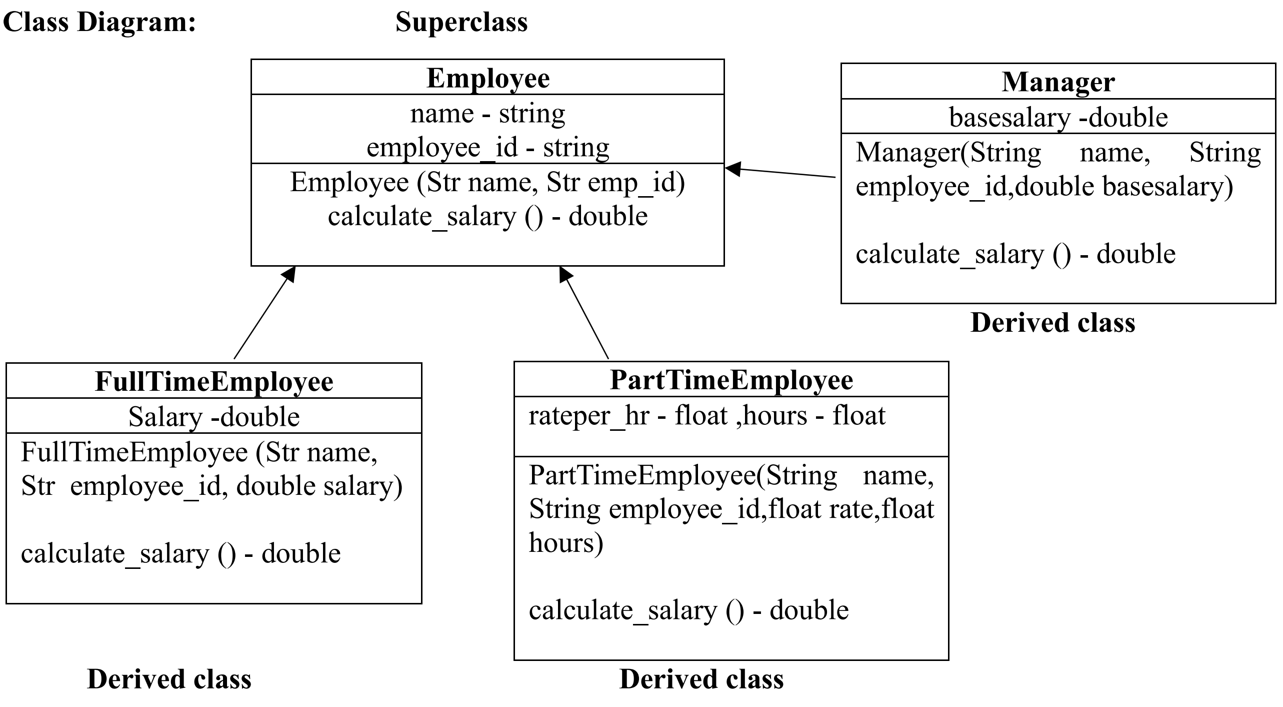
```
        package com.lab_assessment2;


        //BASE CLASS VEHICLE
        public class Vehicle {


                private String make;
                private String model;
        public Vehicle(String make , String model)
        {
                this.make = make;
                this.model = model;


        }

        public void start_engine()
        {
                System.out.println("engine starting method");
        }
        public void stop_engine()
        {
                System.out.println("engine stoping method");
        }

        public void print_vehicle_info()
        {
                System.out.println("make : "+this.make);
                System.out.println("model : "+this.model);


        }



    }
```

```
package com.lab_assessment2;

//derived class CAR
public class Car extends Vehicle{



	public Car(String make , String model)
	{
		super(make,model);
	}

	@Override
	public void  start_engine() {
		System.out.println("car engine will start by key or by pushing car ");
	}
}
package com.lab_assessment2;

//derived class Motorcycle
public class Motorcycle extends Vehicle
{
	public Motorcycle( String make, String model) {
		super(make, model);


	}
	@Override
	public void start_engine()
	{
		System.out.println("Motor Cycle can be start by kicker or by self button");
```

```
        }
}
package com.lab_assessment2;
//derived class Truck
public class Truck extends Vehicle
{
        public Truck(String make, String model) {
                super(make,model);
        }
        @Override
        public void start_engine() {
                System.out.println("Truck engine will start by key or by pushing Truck ");
        }
}
```

**Q3)** Build a class structure for managing different types of employees in a company. Create a base class Employee with attributes name, employee_id, and method calculate_salary(). Derive classes FullTime Employee and PartTime Employee from the Employee class. Implement method overloading in the calculate_salary() method to handle different salary calculations based on employment type. Additionally, create a derived class Manager with an overridden calculate_salary() method that includes a bonus based on performance.

**Class Diagram:**


Superclass
Employee
name - string
employee_id - string
Employee (Str name, Str emp_id)
calculate_salary () - double
Manager
basesalary -double
Manager(String name, String employee_id,double basesalary)
calculate_salary () - double
Derived class
FullTimeEmployee
Salary -double
FullTimeEmployee (Str name, Str employee_id, double salary)
calculate_salary () - double
PartTimeEmployee
rateper_hr - float ,hours - float
PartTimeEmployee(String name, String employee_id,float rate,float hours)
calculate_salary () - double
Derived class
Derived class

**Program:**

```
 package com.lab_assessment2;

//BASE CLASS Employee
public class Employee {
	private String name;
	private String employee_id;

	public Employee (String name,String emp_id)
	{
		this.name=name;
		this.employee_id=emp_id;

	}

	double calculate_salary ()
	{

		return 0.0;
	}
}

package com.lab_assessment2;

//DERIVE CLASS
public class FullTimeEmployee extends Employee {

	private double salary;

	public FullTimeEmployee (String name, String employee_id, double salary) {
		super(name, employee_id);
```

```
                this.salary = salary;
        }

        double calculate_salary()
        {
                return this.salary;
        }
}

package com.lab_assessment2;

//DERIVED CLASS
public class PartTimeEmployee extends Employee {

        float rateper_hr, hours;
        public PartTimeEmployee (String name, String employee_id, float rate, float hours)
        {
                super(name,employee_id);
                this.rateper_hr=rate;
                this.hours=hours;


        }

        double calculate_salary ()
        {

                return rateper_hr*this.hours;
        }
}
```

```
package com.lab_assessment2;
import java.util.*;

// DERIVED CLASS
public class Manager extends Employee {

        double basesalary;
        public Manager(String name, String employee_id,double basesalary)
        {
                super(name,employee_id);
                this.basesalary=basesalary;
        }

        double calculate_salary()
        {
                double performance_bonus;
                Scanner input = new Scanner(System.in);
                System.out.print("Enter performance bonus : ");
                performance_bonus = input.nextDouble();
                input.close();

                return this.basesalary+performance_bonus;
        }

}
```