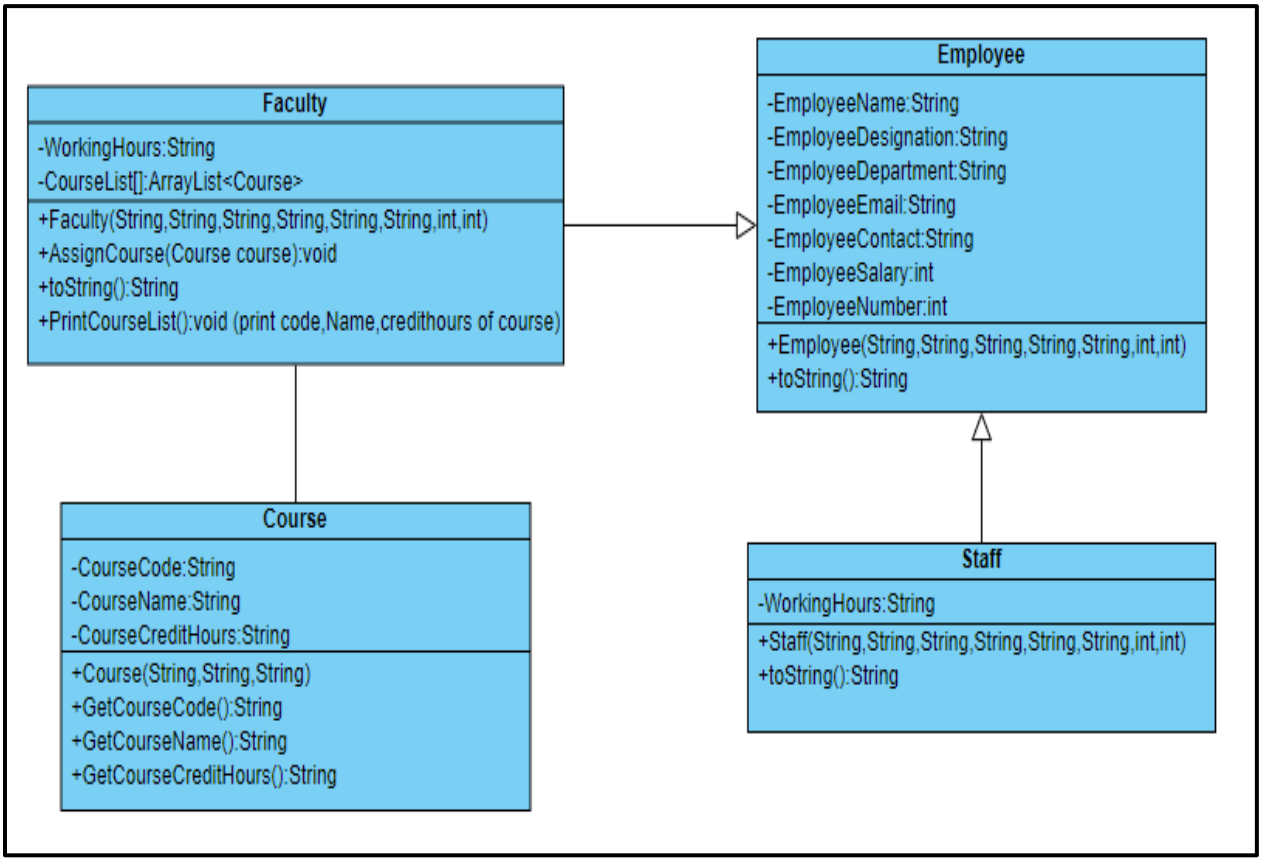
Lab 08 – Inheritance

Task 01:

The UML diagram of a system is given that contains the following classes. Write Java code for these classes.



Implement all classes given in UML diagram.

Create Driver class named as **Inheritance_1**. Create proper Objects of all classes as follows

- Create Two objects of Staff class
- Create Two Objects of Faculty Class
- Create Three Objects of Course class

Assign each Faculty courses of your choice and properly display all the information of Faculty, assigned courses and staff members.

Code:

```
package Task01;

public class Course {
    private String code;
    private String name;
    private int cr;

    // constructor
    public Course(String code, String name, int cr){
        this.code = code;
        this.name = name;
        this.cr = cr;
    }

    public String toString(){
        return code + " : " + name + " : " + cr;
    }

    // getters
    public String getCode(){
        return code;
    }
    public String getName(){
        return name;
    }
    public int getCr(){
        return cr;
    }


}
```

```
package Task01;

public class Employee {
    private int empNo;
    private String name;
    private String designation;
    private String department;
    private String email;
    private int contact;
    private double salary;
```

```
    // constructor
    public Employee(int empNo, String name, String designation, String
departement, String email, int contact, double salary){
        this.empNo = empNo;
        this.name = name;
        this.designation = designation;
        this.department = departement;
        this.email = email;
        this.contact = contact;
        this.salary = salary;
    }

    // method to represent object textually
    public String toString(){
        return empNo + " : " + name + " : " + designation + " : " + department +
" : " + email + " : " + contact + " : " + salary;
    }
}
```

```
package Task01;
import java.util.List;
import java.util.ArrayList;

public class Faculty extends Employee{
    private int workHr;
    private List<Course> courses;

    // constructor
    public Faculty(int empNo, String name, String designation, String department,
String email, int contact, double salary, int workHr){

        super(empNo, name, designation, department, email, contact, salary);

        this.workHr = workHr;

        courses = new ArrayList<>(); // array list of new courses got created
    }

    // method to assign course
    public void assignCourse(Course c){
        this.courses.add(c);
    }
```

```
    // toString method
    public String toString(){
        return super.toString() + " : " + workHr + " : " + courses;
    }

    // method to print course detail (useless but was asked in question)
    public void printDetail(){
        for(Course c : courses){
            System.out.println(c.getCode() + " | " + c.getName() + " | " +
c.getCr());
        }
    }

}
```

```
package Task01;

public class Staff extends Employee {
    private int workHr;

    // constructor
    public Staff(int empNo, String name, String designation, String departement,
String email, int contact, double salary, int workHr){

        // explicitly giving parameters to super() method to call constructor of
parent class
        super(empNo, name, designation, departement, email, contact, salary);

        this.workHr = workHr;
    }

    public String toString(){
        return super.toString() + " : " + workHr;
    }

}
```

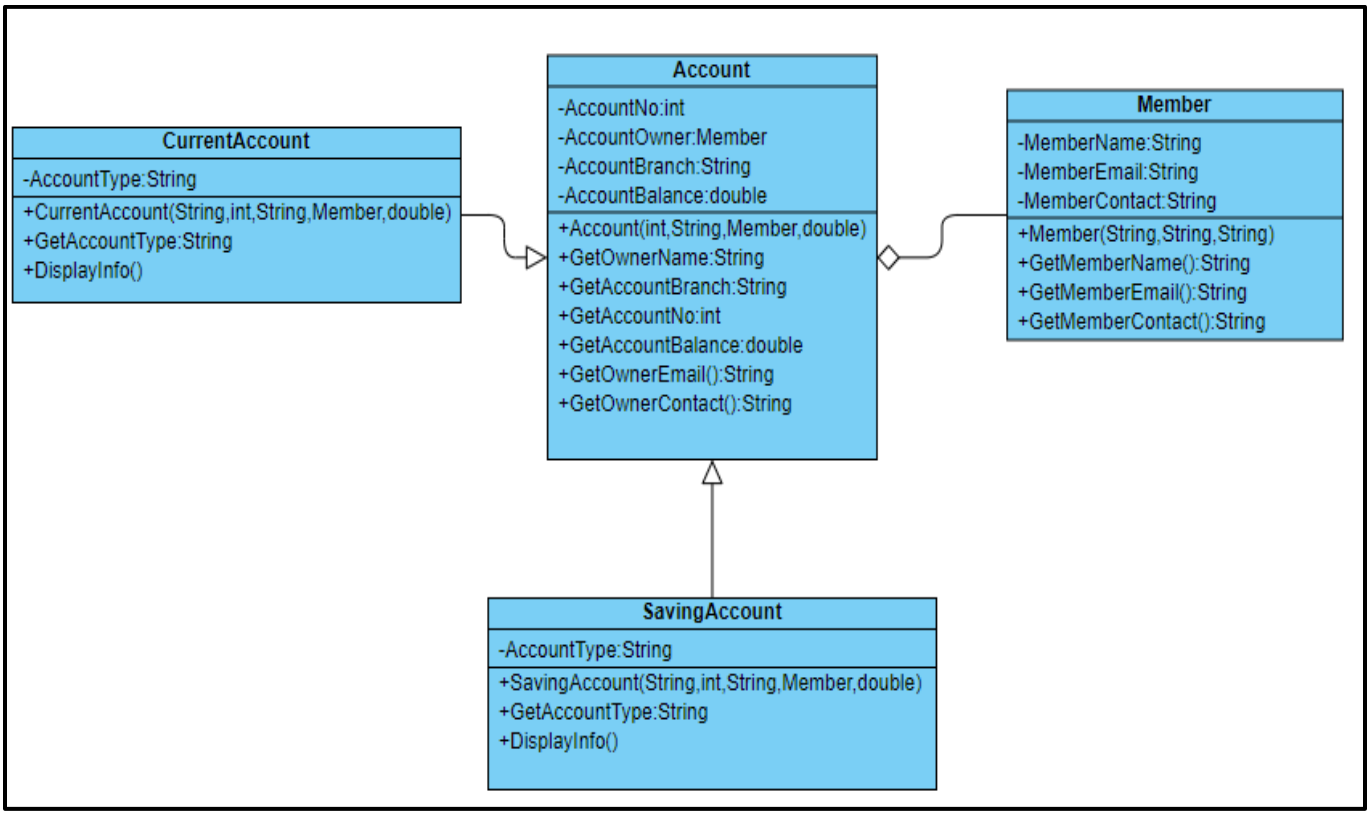
```java
package Task01;

public class Main {
    public static void main(String[] args) {
        // 1. two objects of staff class
        Staff s1 = new Staff(1, "Hasan", "HoD", "CS", "hasan@gmail.com", 12345,
10000000, 6);
        Staff s2 = new Staff(2, "Ghafoor", "DG", "SS", "ghafoor@gmail.com",
98765, 1000000, 5);

        // 2. two object of faculty class
        Faculty f1 = new Faculty(3, "Ishtiaq", "Sr. Professor", "CS",
"ishtiaq@gmail.com", 12345, 20000, 6);
        Faculty f2 = new Faculty(4, "Shabbir", "Assistant", "CS",
"shabbir@gmail.com", 7878, 30000, 7);

        // 3. three objects of courses
        Course c1 = new Course("CS101", "DSA", 4);
        Course c2 = new Course("CS102", "OOP", 3);
        Course c3 = new Course("CE103", "CCN", 4);

        // assigning course to faculty
        f1.assignCourse(c1);
        f1.assignCourse(c2);
        f1.assignCourse(c3);

        f2.assignCourse(c3);


        // displaying all information
        System.out.println("\t\t\t---All Data--- ");

        System.out.println("\n1. Staff Members");
        System.out.println("\t" + s1);
        System.out.println("\t" + s2 + "\n");

        System.out.println("\n2. Faculty Members");
        System.out.println("\t" + f1);
        System.out.println("\t" + f2 + "\n");

    }
}
```

Output:

```
PS D:\Hasan\OOP\University\Lab 08 - Inheritance> javac Task01/Main.java
PS D:\Hasan\OOP\University\Lab 08 - Inheritance> java Task01/Main
                        ---All Data---

1. Staff Members
        1 : Hasan : HoD : CS : hasan@gmail.com : 12345 : 1.0E7 : 6
        2 : Ghafoor : DG : SS : ghafoor@gmail.com : 98765 : 1000000.0 : 5


2. Faculty Members
        3 : Ishtiaq : Sr. Professor : CS : ishtiaq@gmail.com : 12345 : 20000.0 : 6 :
[CS101 : DSA : 4, CS102 : OOP : 3, CE103 : CCN : 4]
        4 : Shabbir : Assistant : CS : shabbir@gmail.com : 7878 : 30000.0 : 7 : [CE10
3 : CCN : 4]

PS D:\Hasan\OOP\University\Lab 08 - Inheritance>
```

Task 02:

The UML diagram of a Bank Account system that contains the following classes. Write Java code for these classes.



Implement all classes given in UML diagram.

Create Driver class named as **Inheritance_2**. Create proper Objects of all classes as follows

- Create three objects of **Member** class. Two of which have **current account** and one has **saving account**

And properly display all the information.

Code:

```
package Task02;

public class Member {
    private String name;
    private String email;
    private String contact;

    // constructor
    public Member(String name, String email, String contact){
        this.name = name;
        this.email = email;
        this.contact = contact;
    }

    // toString
    @Override
    public String toString() {
        return "Member [name=" + name + ", email=" + email + ", conatct=" +
contact + "]";
    }

    // getters
    public String getName(){
        return name;
    }
    public String getEmail(){
        return email;
    }
    public String getContact(){
        return contact;
    }

}
```

```
package Task02;

public class Account {
    private int no;
    private Member owner;
    private String branch;
    private double balance;
```

```
    // constructor
    public Account(int no, Member owner, String branch, double balance){
        this.no = no;
        this.owner = owner;
        this.branch = branch;
        this.balance = balance;
    }

    // toString
    @Override
    public String toString() {
        return "Account [no=" + no + ", owner=" + owner + ", branch=" + branch +
", balance=" + balance + "]";
    }

    // getters
    public int getNo() {
        return no;
    }
    public Member getOwner() {
        return owner;
    }
    public String getBranch() {
        return branch;
    }
    public double getBalance() {
        return balance;
    }

}
```

```
package Task02;

public class CurrentAccount extends Account{
    private String type;

    // constructor
    public CurrentAccount(int no, Member owner, String branch, double balance,
String type) {
        super(no, owner, branch, balance);
        this.type = type;
    }
```

```
    // toString
    @Override
    public String toString() {
        return "CurrentAccount [type=" + type + ", toString()=" +
super.toString() + "]";
    }

    // getter
    public String getType(){
        return type;
    }

    // method to display info
    public void displayInfo() {
        System.out.println("----- Current Account Details -----");
        System.out.println("Account Number : " + getNo());
        System.out.println("Owner Name     : " + getOwner().getName());
        System.out.println("Owner Email    : " + getOwner().getEmail());
        System.out.println("Owner Contact  : " + getOwner().getContact());
        System.out.println("Branch         : " + getBranch());
        System.out.println("Balance        : " + getBalance());
        System.out.println("Account Type   : " + type);
        System.out.println("-----------------------------------");
    }


}



package Task02;

public class SavingAccount extends Account{
    private String type;

    // constructor
    public SavingAccount(int no, Member owner, String branch, double balance,
String type) {
        super(no, owner, branch, balance);
        this.type = type;
    }

    // getter
    public String getType(){
        return type;
```

```
    }

    // toString
    @Override
    public String toString() {
        return "SavingAccount [type=" + type + ", toString()=" + super.toString()
+ "]";
    }

    // method to display info
    public void displayInfo() {
        System.out.println("----- Saving Account Details -----");
        System.out.println("Account Number : " + getNo());
        System.out.println("Owner Name     : " + getOwner().getName());
        System.out.println("Owner Email    : " + getOwner().getEmail());
        System.out.println("Owner Contact  : " + getOwner().getContact());
        System.out.println("Branch         : " + getBranch());
        System.out.println("Balance        : " + getBalance());
        System.out.println("Account Type   : " + type);
        System.out.println("---------------------------------");
    }


}
```

```
package Task02;

public class Main {
    public static void main(String[] args) {

        // three member classes
        Member m1 = new Member("Hasan", "hasan@gmail.com", "02-135251-040");
        Member m2 = new Member("Ghafoor", "ghafoor@gmail.com", "03-135251-040");
        Member m3 = new Member("Ishtiaq", "ishtiaq@gmail.com", "04-135251-040");

        // two members have current account
        CurrentAccount ca1 = new CurrentAccount(101, m1, "Thatta", 5000,
"Current");
        CurrentAccount ca2 = new CurrentAccount(102, m3, "Badin", 2500,
"Current");

        // one has saving
```

```
        SavingAccount sa1 = new SavingAccount(103, m2, "Narowal", 1000,
"Saving");

        // displaying all information
        System.out.println("\n*****Accounts Information*****\n");
        ca1.displayInfo();
        System.out.println();
        ca2.displayInfo();
        System.out.println();
        sa1.displayInfo();


    }
}
```

Output:

```
PS D:\Hasan\OOP\University\Lab 08 - Inheritance> javac Task02/Main.java
PS D:\Hasan\OOP\University\Lab 08 - Inheritance> java Task02/Main

*****Accounts Information*****

----- Current Account Details -----
Account Number : 101
Owner Name     : Hasan
Owner Email    : hasan@gmail.com
Owner Contact  : 02-135251-040
Branch         : Thatta
Balance        : 5000.0
Account Type   : Current
-----------------------------------

----- Current Account Details -----
Account Number : 102
Owner Name     : Ishtiaq
Owner Email    : ishtiaq@gmail.com
Owner Contact  : 04-135251-040
Branch         : Badin
Balance        : 2500.0
Account Type   : Current
-----------------------------------

----- Saving Account Details -----
Account Number : 103
Owner Name     : Ghafoor
Owner Email    : ghafoor@gmail.com
Owner Contact  : 03-135251-040
Branch         : Narowal
Balance        : 1000.0
Account Type   : Saving
-----------------------------------
PS D:\Hasan\OOP\University\Lab 08 - Inheritance>
```