ASSIGMENT JAVA DAY5

Harshit Kushmakar| 16896

1.Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call method of parent class by object of parent class method of child class by object of child class.

```
package assignment5;

public class Child extends Parent {
    public void method() {
        System.out.println("This is a child class");
    }
}
```

```
package assignment5;

public class Parent {
    public void method() {
        System.out.println("This is parent class");

    }

}
```

Main:

```
package assignment5;

public class ParentChildMain {
    public static void main(String[] args) {
        Parent p = new Parent();
        Child c = new Child();
    p.method();
    c.method();
    }

}
```

2.In the above example, declare the method of the parent class as private and then repeat the operations.

```
package assignment5;

public class Parent {
    Parent(){
```

```
        System.out.println("Calling the parent Class Constructor");
    }
   private void method() {////We Cannot call this private parent
class method from ParentChildMain's main() method.


System.out.println("This is parent class");

    }


}
```

3.With the help of a demo code, explain the constructor calling sequence in multi-level inheritance.

```
package assignment5;

public class Parent {
    Parent(){
        System.out.println("Calling the Parent Class Constructor");
    }
    public void method(){
        System.out.println("This is Parent class");
    }


}
```

```
package assignment5;

public class Child extends Parent {
    Child(){
        System.out.println("Calling the Child Class Constructor");
    }
    public void method(){
        System.out.println("This is a Child Class");
    }
}
```

```
package assignment5;

public class SubChild extends Child {

    SubChild(){
        System.out.println("Calling the SubChild Constructor");
    }
}
```

4.Create a class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

```
1. package assignment5;

   public class Parent {

       public void message(){
           System.out.println("This is Parent class");
       }


   }
```

```
package assignment5;

public class Child extends Parent {
    public void message() {
        System.out.println("This is subclass");
    }
}
```

```
package assignment5;

public class SubChild extends Child {

   public void message(){
        System.out.println("This is second subclass");
    }
}
```

5. Make the class ‘Parent’ as an abstract class and the method ‘message’ as abstract. Override the method in both subclasses. Call the methods ‘message’ by creating the reference of Parent class and storing the object of each subclass in it. Comment the findings which method got invoked when.

```java
 package assignment5;

abstract class Parent {

    abstract void message();


}
```

```java
package assignment5;

public class Child extends Parent {
    public void message() {
        System.out.println("This is First subclass");
    }
}
```

```java
package assignment5;
import assignment5.Child;

public class SubChild extends Parent {

        public void message() {
            System.out.println("This is second subclass");
        }
    }
```

```java
package assignment5;

public class ParentChildMain {
    public static void main(String[] args) {
        Parent p1 = new Child();
        Parent p2 = new SubChild();
    p1.message();
    p2.message();
    }

}
```

```
"C:\Program Files\Java\jdk-11\bin\java.e
This is First subclass
This is second subclass

Process finished with exit code 0
```

6. Write the classes as shown in the following class diagram.

a. Mark all the overridden methods with annotation @Override.

b. Call the constructor of Person class from Student & Staff class constructor using super keyword.

c. In the overridden method, use super keyword to call the base class method.

```java
package assignment5;
import assignment5.Staff.*;
public class PersonStudentStaffMain {
    public static void main(String[] args) {
        Person p = new Person("Harshit", "Jhumri teliya");
        Student s = new Student("Harshit", "Jhumri teliya","IT", 4, 40600);
        Staff st = new Staff("Kushmakar", "Jhumri teliya", "SHS", 400);
        System.out.println(p);
        System.out.println(s);
        System.out.println(st);
    }
}
```

```java
package assignment5;

public class Staff extends Person{
    private String school;
    private double pay;
    Staff(String name, String address, String school, double pay){
        super(name, address);
        this.school = school;
        this.pay = pay;
    }
    public String getSchool() {
        return school;
    }
    public void setSchool(String school) {
        this.school = school;
    }
    public double getPay() {
        return pay;
    }
```

```
    public void setPay(double pay) {
        this.pay = pay;
    }
    @Override
    public String toString(){
        return "Staff[" + super.toString() + ", school=" + getSchool() + ",
pay=" + getPay();
    }
}
```

```
package assignment5;

public class Student extends Person {
    private String program;
    private int year;
    private double fee;
    Student(){
    }
    Student(String name, String address, String program,int year, double
fee) {
        super(name, address);
        this.program = program;
        this.year = year;
        this.fee = fee;
    }
    public String getProgram() {
        return program;
    }
    public int getYear() {
        return year;
    }
    public double getFee() {
        return fee;
    }
    public void setProgram(String program) {
        this.program = program;
    }
    public void setYear(int year) {
        this.year = year;
    }
    public void setFee(double fee) {
        this.fee = fee;
    }
    @Override
    public String toString(){
        return "Student[" + super.toString() +", program = " + getProgram()
+ ", year = " + getYear() + ", fee = " + getFee();
    }
}
```

```
package assignment5;

public class Person {
    private String name;
    private String address;
    Person(){
    }
```

```
    Person(String name, String address){
        this.name = name;
        this.address = address;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String toString(){
        return "Person[name = " + name + ", address = " + address + "]";
    }
}
```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Co
Person[name = Harshit, address = Jhumri teliya]
Student[Person[name = Harshit, address = Jhumri teliya], program = IT, year = 4, fee = 40600.0
Staff[Person[name = Kushmakar, address = Jhumri teliya], school=SHS, pay=400.0

Process finished with exit code 0
```

7. Create the LoanProduct class hierarchy according to the given Class Diagram.

Create the required constructor and write a test class to test the functionalities.

a. The method 'LTVCalculationAsPerCollatoralType' need to be overridden in all three classes. The formula to be used is given in Assignment-1. Pass the 'LoanAmountAsked' value in the method.

b. Use upcasting to call this method.

8. Make the base class – LoanProduct as an abstract class and the method as an abstract method. Use upcasting to call the abstract method.

```
package assignment5;

public class ConsumerVehiclesLoan extends LoanProduct{
    private String assetCategory;
    private String assetVariant;
    private String assetModel;
    private String manufacture;
    private int yearOfManufacture;
    private double assetCost;
```

```
    private double downPayment;



    @Override
    public double LTVCalculationAsPerCollateralType(double LoanAmountAsked,
double collateral) {
        return 12;
    }
}


package assignment5;

public class EducationLoan extends LoanProduct{
  private String CourseName;
  private String CollegeName;
  private String CourseType;
  private String DegreeType;
  private String educationStream;
  private double totalFees;


    @Override
    public double LTVCalculationAsPerCollateralType(double LoanAmountAsked,
double collateral) {
        return 0;
    }


}


package assignment5;

public class HomeLoan extends LoanProduct {
    private String PropertyType;
    private String NatureOfProperty;
    private String propertyPurpose;
    private String propertyOwnership;
    private double marketValue;
    private double builtUpArea;
    private double carpetArea;

    private int propertyAge;

    public HomeLoan(String propertyType, String natureOfProperty, String
propertyPurpose, String propertyOwnership,
                    double marketValue, double builtUpArea, double
carpetArea, int propertyAge) {
        PropertyType = propertyType;
        NatureOfProperty = natureOfProperty;
        this.propertyPurpose = propertyPurpose;
        this.propertyOwnership = propertyOwnership;
        this.marketValue = marketValue;
        this.builtUpArea = builtUpArea;
        this.carpetArea = carpetArea;
        this.propertyAge = propertyAge;
    }
```

```
    @Override
    public double LTVCalculationAsPerCollateralType(double LoanAmountAsked,
double collateral) {
        return 14;
    }
}
```

```
package assignment5;

public abstract class LoanProduct {
    private String loanProductCode;
    private String loanProductName;
    private boolean assetBased;
    private String loanSecurityType;
    private double minTenure;
    private double maxTenure;
    private double minLoanAmount;
    private double maxLoanAmount;
    private double roi;
    private double ltv;
    //We Must use upcasting to call this
LTVCalculationAsPerCollatoralType() Method.
    public abstract double LTVCalculationAsPerCollateralType(double
LoanAmountAsked, double collateral);

}
```

```
package assignment5;

public class Main {
    public static void main(String[] args) {
        LoanProduct lp = new
HomeLoan("Private","privatePurpose","selling","kushmakar",100000,300000,
20000,20);
        LoanProduct lp1 = new EducationLoan();
        System.out.println(lp);
    }
}
```