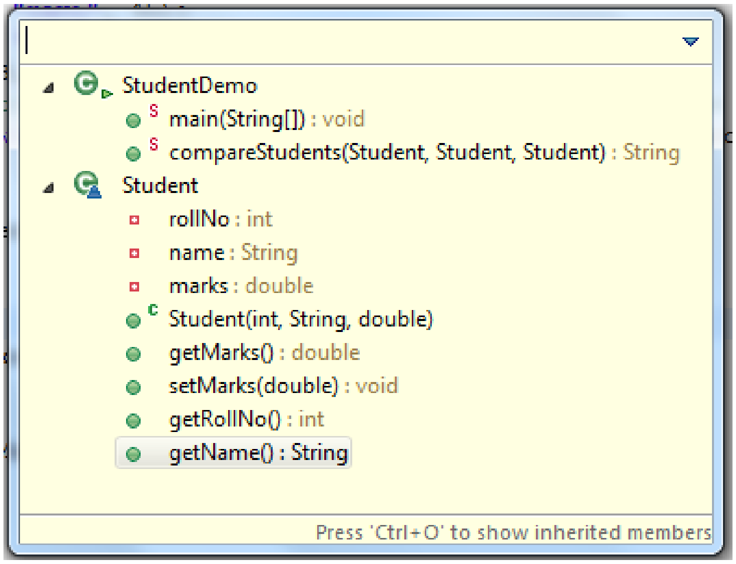
Refer below sample problem with solution to solve remaining problems. Ensure same approach is followed. Use exactly same class names, attribute names and types, constructor, method names and signatures as specified in the problem statement. Any mistake on these instructions may result into invalid submission of the assignments even if logic is 100% correct.

1. Create Class Student with below attributes:
   rollNo
   name
   marks
   Create class StudentDemo with main method. Create three student objects with relevant data in main method. Write separate method in this class which will take these three student objects as parameter and return name of student with highest marks.
   Follow class outline diagram as given below. Ensure class attributes are private and other methods are public. **Use package “com” to build your solution.**

   Class Outline:



```
StudentDemo
    main(String[]) : void
    compareStudents(Student, Student, Student) : String
Student
    rollNo : int
    name : String
    marks : double
    Student(int, String, double)
    getMarks() : double
    setMarks(double) : void
    getRollNo() : int
    getName() : String
```

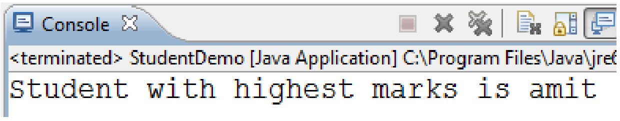
   Code:

   package com;

```
public class StudentDemo {

      public static void main(String[] args) {

            Student one = new Student(1, "ravi", 45);
            Student two = new Student(2, "amit", 65);
            Student three = new Student(3, "pooja", 55);
            //trainees may use scanner object to read above values.
//However, use separate scanner for String and numbers
```

```
        System.out.println("Student with highest marks is " +
        compareStudents(one, two, three));
    }

    public static String compareStudents(Student one, Student two,
Student three)
    {
        Student st = one;

        if(two.getMarks() >st.getMarks())
            st = two;

        if(three.getMarks() > st.getMarks())
            st = three;

        return st.getName();
    }

}
class Student
{
    private int rollNo;
    private String name;
    private double marks;

    public Student(int rollNo, String name, double marks) {
        this.rollNo = rollNo;
        this.name = name;
        this.marks = marks;
    }

    public double getMarks() {
        return marks;
    }

    public void setMarks(double marks) {
        this.marks = marks;
    }

    public int getRollNo() {
        return rollNo;
    }

    public String getName() {
        return name;
    }


    }
```

Output:



```
<terminated> StudentDemo [Java Application] C:\Program Files\Java\jre
Student with highest marks is amit
```

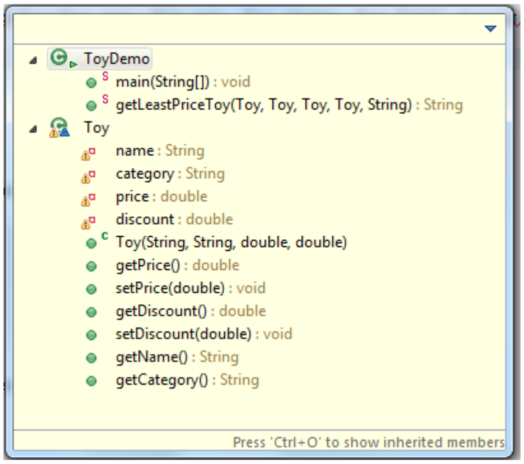
Each trainee is supposed to write above program and try out even if they are aware about Java language. Along with learning Java numeric computations, below points are very important to practice and applicable throughout this ILP training as well as most important for any software code.

- Use exactly same class names as mentioned
- Use exactly same method signature (method name, return type, method parameter type, position of each method parameter)
- Define attributes with same name and data type as given in class outline.
- Define constructors and getter setters as given in the class outline.
- Ensure attributes are private and other methods which will be called from main method, getter-setter methods and constructor is public.
- Use main method only for input and output and testing object creation and object methods.

As mentioned above, any logic which may be 100% correct is not valid if above points are not taken care. Hence, simply building logic does not certify us as project ready. Building exact and complete solution does.

2. Create Class Toy with below attributes
   name
   category
   price
   discount
   Create Class ToyDemo with main method. Create four toy objects with relevant data. Two objects should have category as “fruits” and other two objects should have category as “anmial”. Write a method in this class which will take all four toy objects as input. Additionally, it will also take category name as input. This method should return least price toy name (considering the discount as well) for specified category. If there is no specific category (e.g category as “fruits”, method should return message “no category found”.
   Follow class outline diagram as given below. Ensure class attributes are private and other methods are public. **Use package “com” to build your solution.**
   NOTE: Use .equals for comparing String values instead of == operator. Search on net and find out why. Else discuss among colleagues.

   Class Outline:

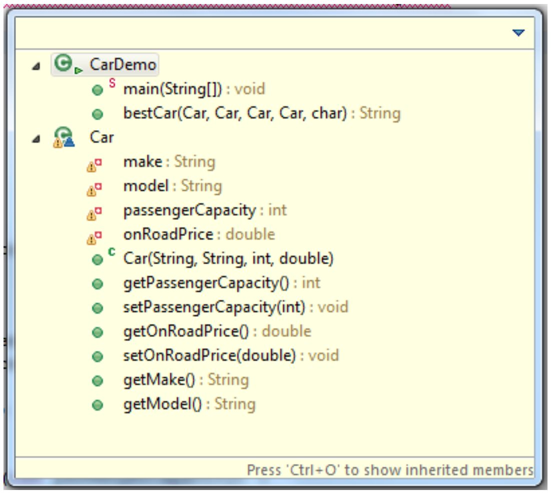


```
ToyDemo
    main(String[]) : void
    getLeastPriceToy(Toy, Toy, Toy, Toy, String) : String
Toy
    name : String
    category : String
    price : double
    discount : double
    Toy(String, String, double, double)
    getPrice() : double
    setPrice(double) : void
    getDiscount() : double
    setDiscount(double) : void
    getName() : String
    getCategory() : String
```

3. Create class Car with below attributes:
   make
   model
   passengerCapacity
   onRoadPrice

   Create class CarDemo which has main method. Create another method - getBestCar in this class which will take four car objects and additional parameter as compareType. Value of this parameter can be ‘c’ or ‘p’, where ‘c’ stands for capacity of passengers and ‘p’ stands for on road price. This method will return make and model name for highest value of ‘c’ or ‘p’ (with – character in between. E.g for make – Hyundai and model – santro, method should return “Hyundai-santro”). Create four objects of Car class in main method and drive it through getBestCar method.
   Follow class outline diagram as given below. Ensure class attributes are private and other methods are public. **Use package “com” to build your solution.**

   Class Outline:

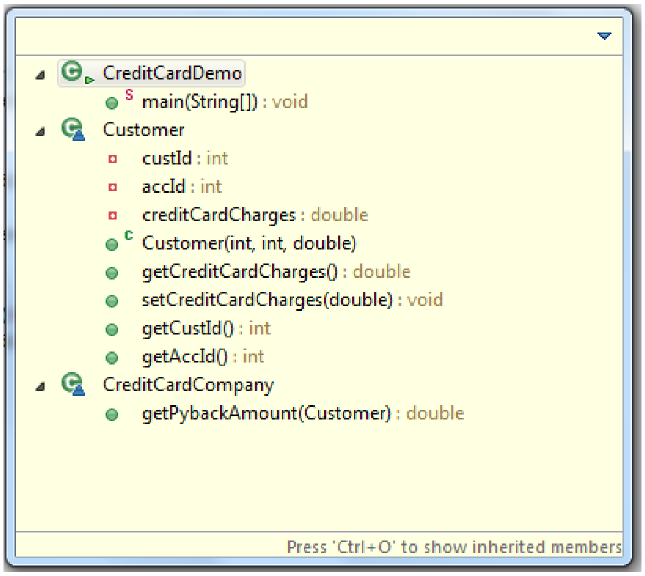


```
CarDemo
    main(String[]) : void
    bestCar(Car, Car, Car, Car, char) : String
Car
    make : String
    model : String
    passengerCapacity : int
    onRoadPrice : double
    Car(String, String, int, double)
    getPassengerCapacity() : int
    setPassengerCapacity(int) : void
    getOnRoadPrice() : double
    setOnRoadPrice(double) : void
    getMake() : String
    getModel() : String
```

4. Create class Customer with below attributes:
   custId
   accountId
   creditCardCharges
   Create another class CreditCardCompany which has a method which takes Customer object as parameter and returns the payback amount for the credit card charges value for that customer. Use below logic to calculate payback amount:

   a) 0.25% for charges up to Rs. 500.
   b) 0.50% for the next Rs.1000 (that is, the portion between Rs. 500 and Rs. 1500),
   c) 0.75% for the next Rs.1000 (that is, the portion between Rs. 1500 and Rs. 2500),
   d) 1.0% for charges above Rs2500.
   Thus, a customer who charges Rs. 400 a year receives Rs.1.00, which is 0.25 • 1/100 • 400, and one who charges Rs1, 400 a year receives Rs. 5.75, which is 1.25 = 0.25 • 1/100 • 500 for the first Rs. 500 and 0.50 • 1/100 • 900 = 4.50 for the next Rs. 900. Determine by hand the pay backs amount for a customer who charged Rs. 2000 and one who charged Rs. 2600.

   Create another class as CreditCardDemo which will have main method. Within main method, create object of CreditCardCompany. Create 5 customer objects with relevant data. Call the method to return payback amount for each customer and display the same
   Follow class outline diagram as given below. Ensure class attributes are private and other methods are public. **Use package "com" to build your solution.**
   Class Outline:



```
CreditCardDemo
    main(String[]) : void
Customer
    custId : int
    accId : int
    creditCardCharges : double
    Customer(int, int, double)
    getCreditCardCharges() : double
    setCreditCardCharges(double) : void
    getCustId() : int
    getAccId() : int
CreditCardCompany
    getPybackAmount(Customer) : double
```