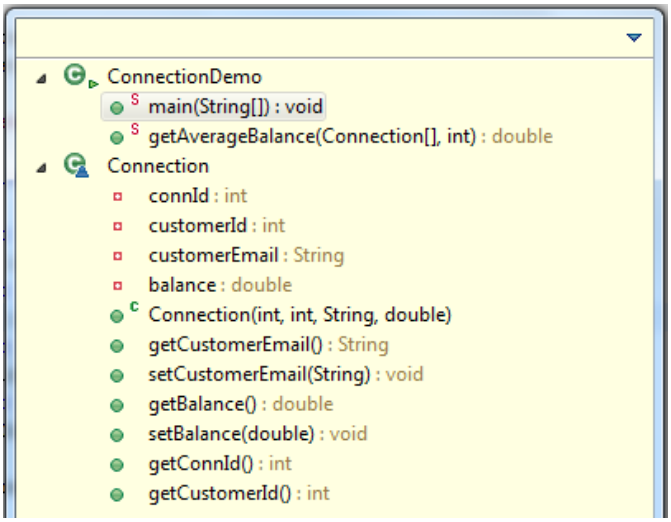
Refer below sample problem with solution to solve remaining problems. Ensure same approach is followed. Use exactly same class names, attribute names and types, constructor, method names and signatures as specified in the problem statement. Any mistake on these instructions may result into invalid submission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

1. Create class Connection with below attributes
   connId
   customerId
   customerEmail
   balance

   Create class ConnectionDemo with main method. Declare array of 3 Connection objects in main method. Initialize this array where few objects have same customer id. Declare another method in this class – getAverageBalance. This method will take the customer array and customer id as input and return average balance for that customer. Display this average balance from main 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.**



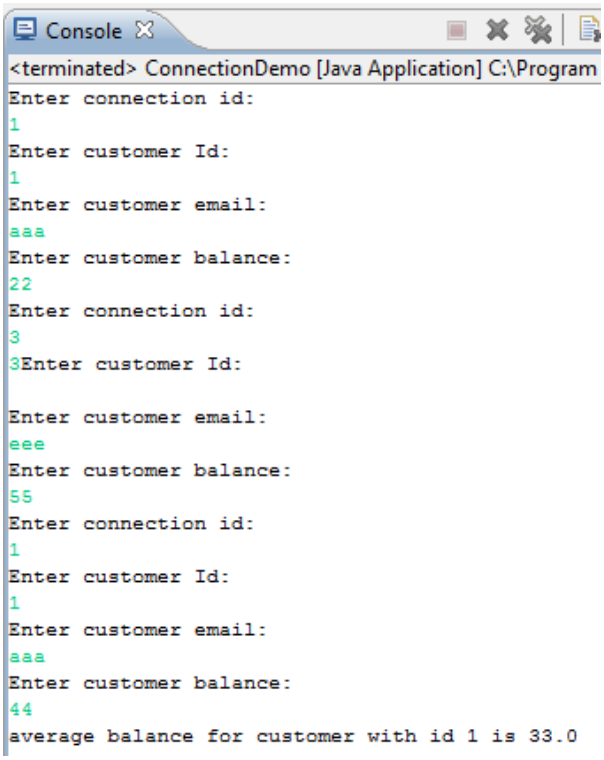
Sample code (Also refer use of Scanner to read numeric and string values):

```
package com;
import java.util.Scanner;

public class ConnectionDemo {

	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);
		Scanner sc1 = new Scanner(System.in);
		Connection[] connections = new Connection[3];

		for(int i = 0;i<connections.length;i++)
```

```
		{
			System.out.println("Enter connection id:");
			int connId = sc.nextInt();
			System.out.println("Enter customer Id:");
			int custId = sc.nextInt();
			System.out.println("Enter customer email:");
			String email = sc1.nextLine();
			System.out.println("Enter customer balance:");
			double balance = sc.nextDouble();

			connections[i] = new Connection(connId, custId,
email, balance);

		}
		double avgBalance = getAverageBalance(connections,1);
		System.out.println("average balance for customer with
id 1 is " + avgBalance);

	}

	public static double getAverageBalance(Connection[]
connections, int custId)
	{
		double balance = 0;
		int custCount = 0;

		for(int i = 0;i<connections.length;i++)
		{
			if(custId == connections[i].getCustomerId())
			{
				custCount++;
				balance = balance +
connections[i].getBalance();
			}
		}

		balance = balance/custCount;

		return balance;
	}


}
class Connection
{
	private int connId;
	private int customerId;
	private String customerEmail;
	private double balance;

	public Connection(int connId, int customerId, String
customerEmail,
			double balance) {
		this.connId = connId;
		this.customerId = customerId;
		this.customerEmail = customerEmail;
		this.balance = balance;
```

```
}

public String getCustomerEmail() {
	return customerEmail;
}

public void setCustomerEmail(String customerEmail) {
	this.customerEmail = customerEmail;
}

public double getBalance() {
	return balance;
}

public void setBalance(double balance) {
	this.balance = balance;
}

public int getConnId() {
	return connId;
}

public int getCustomerId() {
	return customerId;
}



}
```

Output:



Once again repeating below instructions which are very important for a developer…..and hence for your career. Read, understand and follow these points throughout in IT field.

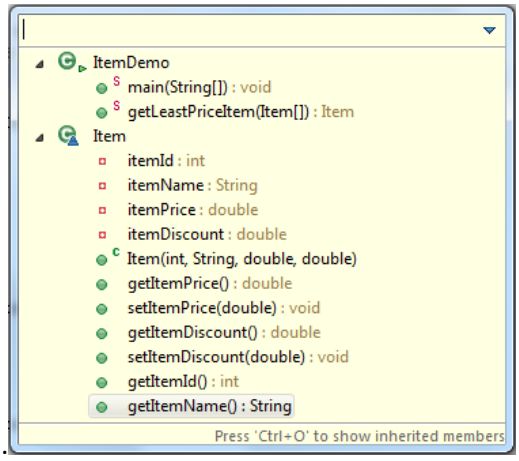
Each trainee is supposed to write above program and try out even if they are aware about Java language. Along with learning Java numeric computations, below points are very important to practice and applicable throughout this ILP training as well as most important for any software code.

- 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.

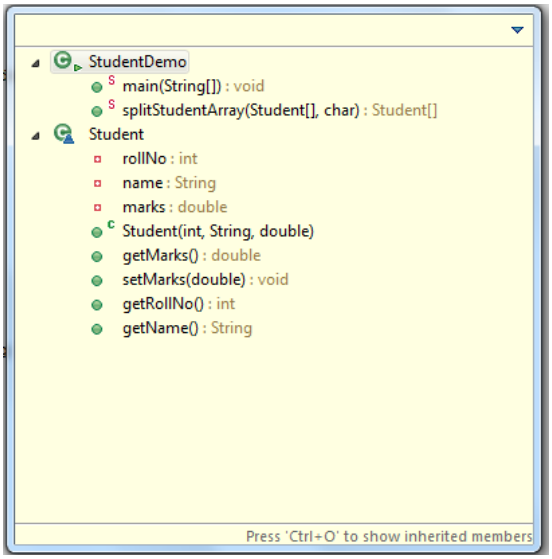
As mentioned above, any logic which may be 100% correct is not valid if above points are not taken care. Hence, simply building logic does not certify us as project ready. Building exact and complete solution does.

2. Create class Item with below attributes
   itemId
   itemName
   itemPrice
   itemDiscount

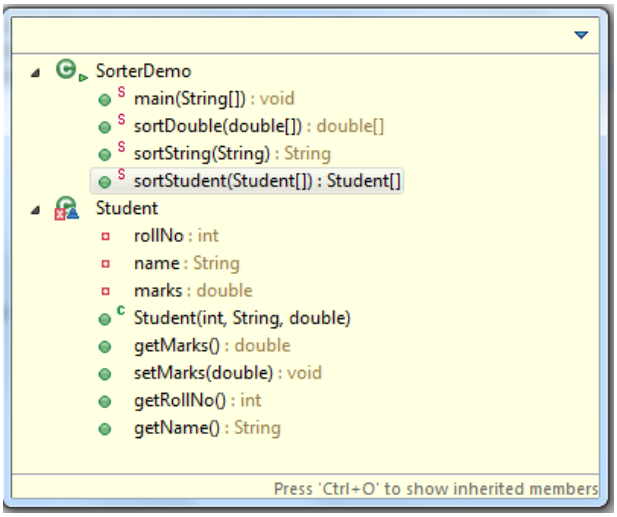
   Create class ItemDemo with main method. Declare array if 5 item objects in main method. Initialize this array. Declare another method in this class – getLeastPriceItem which will take item array as input and return item with least price (considering discount as well). Display item name in main method. Follow class outline diagram as given below. Ensure class attributes are private and other methods are public. **Use package "com" to build your solution.**



3. Create class Student with below attributes:
   rollNo
   Name
   Marks
   Create class StudentDemo with main method. Declare array of 5 student objects in main method. Initialize this array. Declare another method in this class – splitStudentArray. This method will take the student array and a character as input parameters. If the input character is 'o' this method will return array of students with odd value of marks. If the input character is 'e' then this method will return array of students with even value or marks. It will return null array if there is any other character specified. Display name and marks for this returned array from main 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.**



4. Create class SorterDemo with different methods as given in the class outline. Implement sortDouble method which will sort array of double variables. Implement sortString method which will sort given String value in alphabetical manner. Implement sortStudent method which will sort array of Student objects (referring Student class created in above problem). Create array of double variables, declare one String variable and create array of Student objects in main method. Display the output of each array as resulted out of the three sort methods in main 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.**


SorterDemo
main(String[]) : void
sortDouble(double[]) : double[]
sortString(String) : String
sortStudent(Student[]) : Student[]
Student
rollNo : int
name : String
marks : double
Student(int, String, double)
getMarks() : double
setMarks(double) : void
getRollNo() : int
getName() : String
Press 'Ctrl+O' to show inherited members