**ArrayList Assignment**

**GRN:12010664**

**Name of the Student: Vedant Gokhale**

**Roll No.: 37**

**Class: AI**

**Division: A**

**Batch: B2**

**Problem Statement**

A fashion E-commerce company keeps a track of all the orders using an **ArrayList** and a class Order. Implement class **Order** and retrieve and return the list of items present in all the orders. Implement the logic inside **getItems()** method.

Test the functionalities using the **main()** method of the **Tester** class.

**Sample Input and Output**

|  |  |
| --- | --- |
| **Sample Input** | **Expected Output** |
| orders=[Order(101,itemNames=[Jeans, Shirt, Belt],true), Order(102,itemNames=[Tie,Shirt],true),Order(103,itemNames=[Tshirt,Socks,Tie],true) | [Jeans,Shirt,Belt,Tie,Shirt,Tshirt,Socks,Tie] |
| orders=[Order(311,itemNames=[Sportswear, Dumbbell],true), Order(102,itemNames=[, Jeans],true),Order(103,itemNames=[Smartwatch,Fitnessband,Joggers],true) | Sportswear,Dumbbell,Smartwatch,Fitnessband,Joggers] |

|  |
| --- |
| **Add your code here**  import java.util.ArrayList; import java.util.Arrays; import java.util.Scanner;  class Order {  private int orderId;  private ArrayList<String> itemNames;  private boolean orderStatus;   public Order(int orderId, ArrayList<String> items, boolean orderStatus) {  setOrderId(orderId);  setItemNames(items);  setOrderStatus(orderStatus);  }   public int getOrderId() {  return orderId;  }   public void setOrderId(int orderId) {  this.orderId = orderId;  }   public ArrayList<String> getItemNames() {  return itemNames;  }   public void setItemNames(ArrayList<String> itemNames) {  this.itemNames = itemNames;  }   public void setOrderStatus(boolean orderStatus) {  this.orderStatus = orderStatus;  }   }  public class ArrayListAssignment {  public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  System.*out*.print("Enter number of items: ");  ArrayList<String> arr= new ArrayList<String>();  int numItems = input.nextInt();  for (int i=0;i<numItems;i++){  System.*out*.print("Enter Item "+(i+1)+" ");  String str = input.next();  arr.add(str);  }  Order o1 = new Order(100,arr,true);  System.*out*.print("Enter number of items: ");  numItems = input.nextInt();  for (int i=0;i<numItems;i++){  System.*out*.print("Enter Item "+(i+1)+" ");  String str = input.next();  arr.add(str);  }  Order o2 = new Order(101,arr,true);  System.*out*.print("Enter number of items: ");  numItems = input.nextInt();  for (int i=0;i<numItems;i++){  System.*out*.print("Enter Item "+(i+1)+" ");  String str = input.next();  arr.add(str);  }  Order o3 = new Order(100,arr,true);  System.*out*.println(o1.getItemNames());   } } |
| **Expected Output:** |

**HashMap Assignment**

**GRN:**

**Name of the Student:**

**Roll No.:**

**Class:**

**Division:**

**Batch:**

Given a HashMap of type<String, Double> that stores names and corresponding average marks of students in a class, find out who scored the maximum and minimum marks. Implement the logic inside **findMaxMinScorers()** method. The method should return a **HashMap<String, Double>** which contains the maximum marks and the student(s) who scored the marks followed by minimum marks and the students(s) who scored it.

Test the functionalities using the main() method of the Tester class.

|  |  |
| --- | --- |
| **Sample Input** | **Expected Output** |
| studentMarks={“Leena”=90.0,”Rakesh”=68.0,”Monika”=76.4,  ”Neil”=67.0,”Tom”=92.0} | {“Tom”=92.0,”Neil”=67.0} |
| studentMarks={“Leena”=85.0,”Rakesh”=78.5,”Monika”=86.0,  ”Neil”=72.0,”Tom”=86.0} | {“Tom”=86.0,”Monika”=86.0,”Neil”=72.0} |

**Sample Input and Output**

|  |  |
| --- | --- |
| import java.util.\*;  public class Tester {  public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  HashMap<String,Double> hm = new HashMap<String, Double>();  System.*out*.print("Enter number of students: ");  int n = input.nextInt();  for(int i=0;i<n;i++){  System.*out*.print("Enter Name of Student: ");  String name = input.next();  System.*out*.print("Enter marks: ");  double marks = input.nextInt();  hm.put(name,marks);  System.*out*.println();  }   System.*out*.println(HashMapLab.*findMaxMinScores*(hm));  } }  class HashMapLab {   public static HashMap<String,Double> findMaxMinScores(HashMap<String,Double> student){  double minMap = Collections.*min*(student.values());  Double maxMap = Collections.*max*(student.values());   HashMap<String,Double> ans = new HashMap<String,Double>();   for (Map.Entry<String,Double> entry : student.entrySet()){  if(Objects.*equals*(entry.getValue(),minMap)){  ans.put(entry.getKey(),minMap);  }if(Objects.*equals*(entry.getValue(),maxMap)){  ans.put(entry.getKey(),maxMap);  }   }  return ans;  }   } | |
| **Expected Output:** |

**Set Assignment**

**GRN:**

**Name of the Student:**

**Roll No.:**

**Class:**

**Division:**

**Batch:**

**Problem Statement**

A VIT college has declared the end-semester exam results, after which, few students applied for re-evaluation of their answer scripts. The college rule is that a student can apply for re-evaluation only in one subject. The students were not aware of this rule and they applied for re-evaluation in more than one subject. The management observed that the students were able to apply for re-evaluation in more than one subject.

You need to fetch and return the details(studentId and studentName) of the students who have applied for re-evaluation in more than one subject using a **HashSet**. Implement the Student class and also write logic in **findDuplicateEntries()** method of Tester class given below.

**Note:** Two students are considered to be the same if their **studentIds** are the same.

Test the functionalities using the **main()** method of the **Tester** class

**Sample Input and Output**

|  |  |
| --- | --- |
| **Sample Input** | **Expected Output** |
| students=[Student(1001,”Ajay”,111), Student(1002,”Rahul”,112),Student(1003,”Julie”,113), Student(1005,”Nikhil”,113), Student(1100,”Akshay”,112), Student(1003,”Julie”,111), Student(1001,”Ajay”,114), | [Student(1001,”Ajay”), Student(1003,”Julie”),] |
| students=[ Student(4534,”John”,111), Student(2347,”Rosy”,112), Student(4534,”John”,112), Student(4538,”John”,113),Student( 4534,”John”,114),] | [Student(4534,”john”)] |

|  |  |
| --- | --- |
| import java.util.HashSet; import java.util.Scanner;  class Student{  private int studentId;  private String studentName;  private int num;   Student(int studentId,String studentName, int num){  setStudentId(studentId);  setStudentName(studentName);  setNum(num);  }   public void setNum(int num) {  this.num = num;  }   public void setStudentId(int studentId) {  this.studentId = studentId;  }   public void setStudentName(String studentName) {  this.studentName = studentName;  }   @Override  public int hashCode(){  int hashCode = 0;  hashCode+=10;  return hashCode;  }   @Override  public boolean equals(Object obj){  Student other = (Student)obj;  if (studentId== other.studentId){  System.*out*.println("Duplicate Entry is ");  System.*out*.println("ID: "+other.studentId+" Name: "+other.studentName);  System.*out*.println();  return true;  }else{  return false;  }  } }  public class HashSetLab {  public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  HashSet<Student> s = new HashSet<>();  System.*out*.print("Enter number of records: ");  int n = input.nextInt();  Student[] arr =new Student[n];   for (int i=0;i<n;i++){  System.*out*.print("Enter Student ID: ");  int id = input.nextInt();  System.*out*.print("Enter Student Name: ");  String name = input.next();  System.*out*.print("Enter Subject Code: ");  int code = input.nextInt();  arr[i] = new Student(id,name,code);  System.*out*.println();  }  for (int i=0;i<n;i++){  s.add(arr[1]);  }   } } | |
| **Expected Output:** |