**ASSINGMENT ON COLLECTION (Assignmnet-2)**

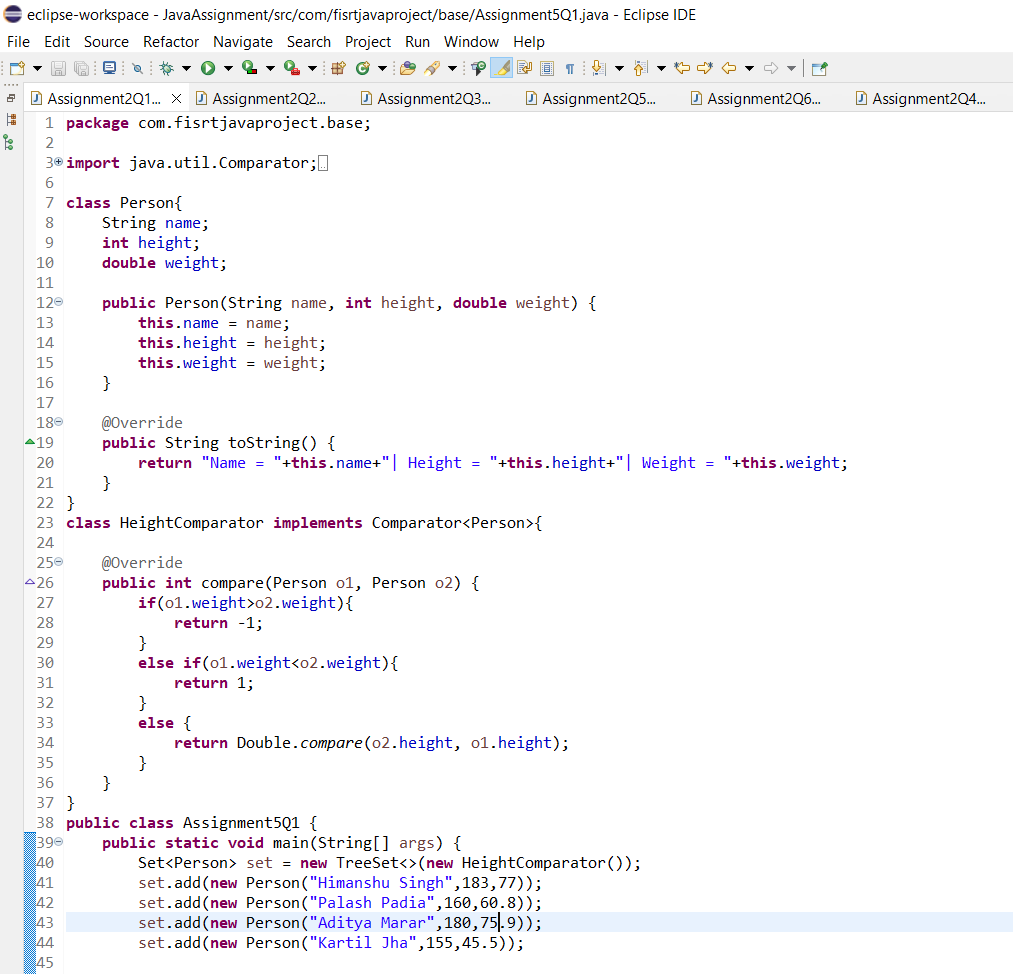
**Q1. Write a class Person having weight, height & name. Create multiple person objects & print them in the sorted order. In the sorting order first sort based upon their weight & it two persons have same weight them sort them based upon their height. Use TreeSet.**

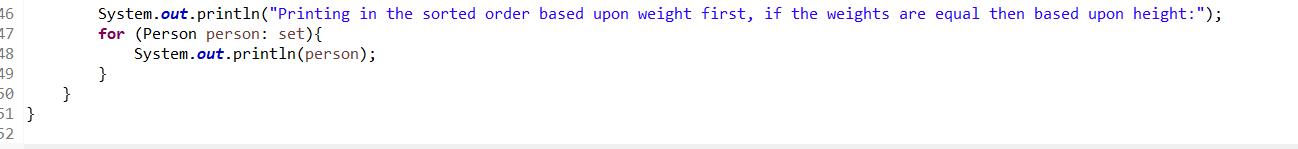
**Description:-**

Create Person class with variables weight and height.Create multiple person objects & print them in the sorted order based upon weight first if the weights are equal then based upon height. Use TreeSet.

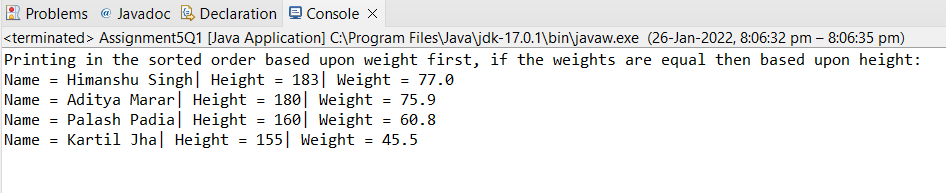
}

**Code:**





**Output:**

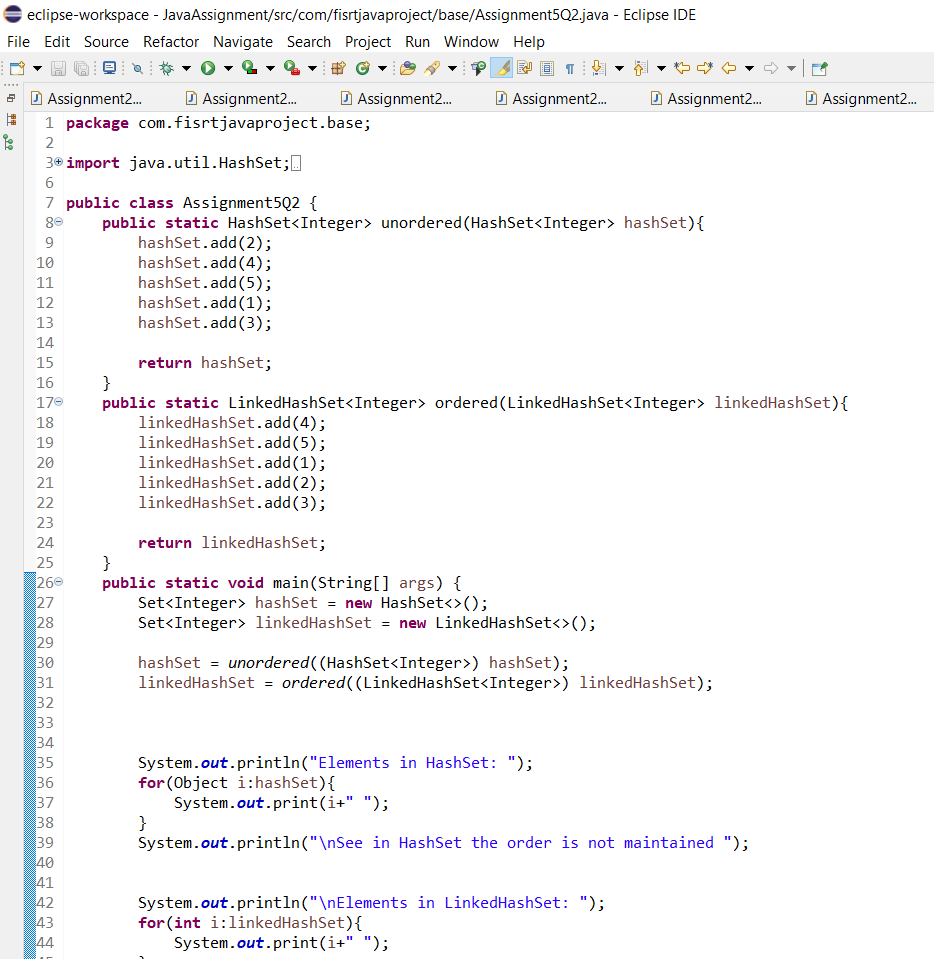


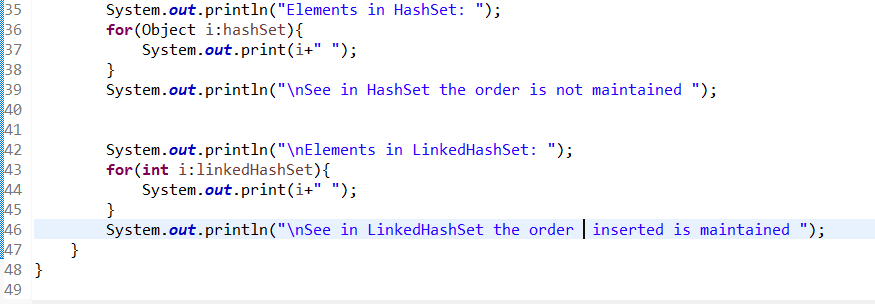
**Q2.  Prove that Hash Set is unordered & Linked Hash Set is ordered.**

**Specifications:**

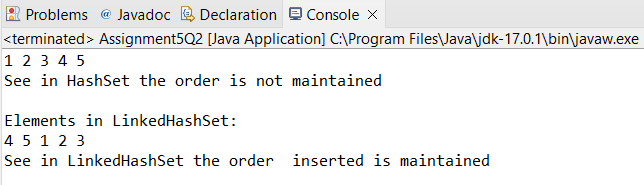
public class Assignment3Q2 {  
    public static void main(String[] args) { }  
    public static LinkedHashSet ordered(LinkedHashSet linkedHashSet){ }  
    public static HashSet unordered(HashSet hashSet){}  
}

**Code:**





**Output:**

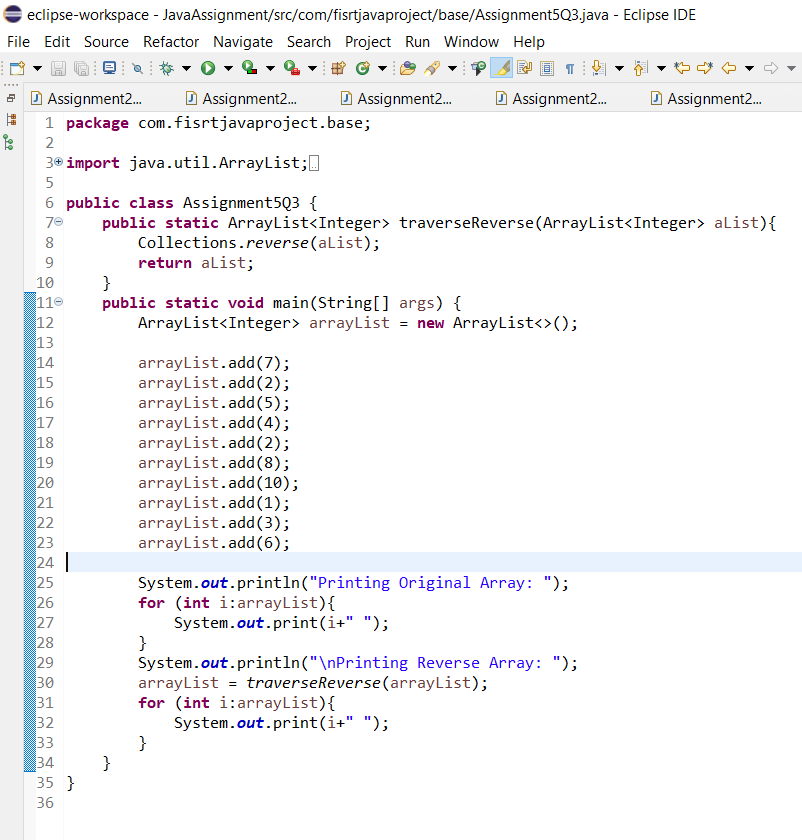


**Q3. Create a ArrayList with few elements & print it in backward direction. Use ListIterator.**

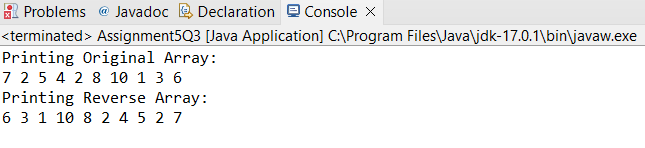
**Description:**

Write a program which consists of ArrayList which has some elements and print them in reverse direction.

**Code:**



**Output:**

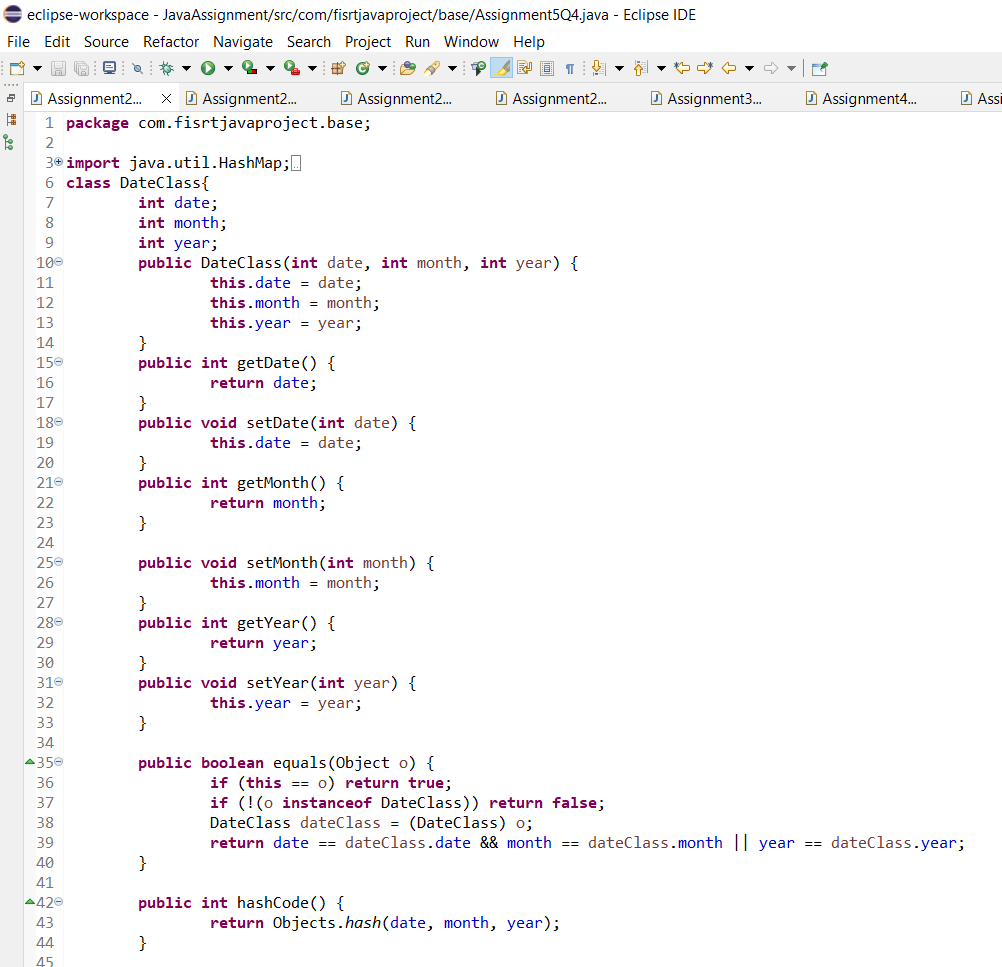


**Q4. Write a program using Hashtable or HashMap where Date of birth is a key & Employee name as value. Design the class Date is such a way where the get method fails if two employees have same day & month of birth but birth year is different**.

**Description:-**

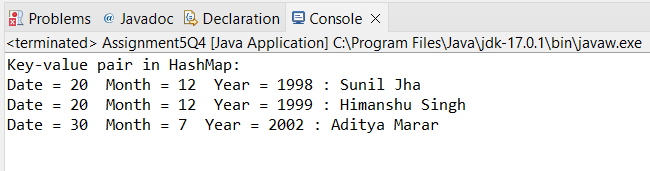
Using hash table or hash map write a program where key is date of birth and employee name is value and also the condition in the question should be satisfied.

**Code:**





**Output:**



**Q5. Write a user defined class say Employee that overrides equals() & hashCode() methods. Equals() always returns true & hashCode() always returns a fixed number. Make such a class as key of you Hashtable. Observe the behavior while calling put & get methods.**

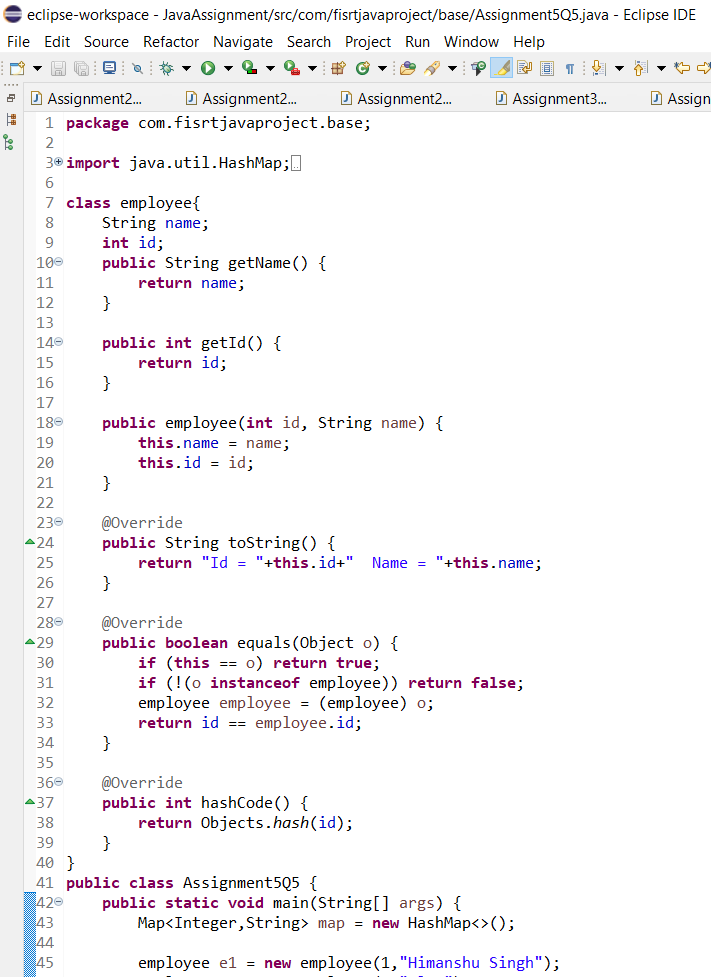
**Description:-**

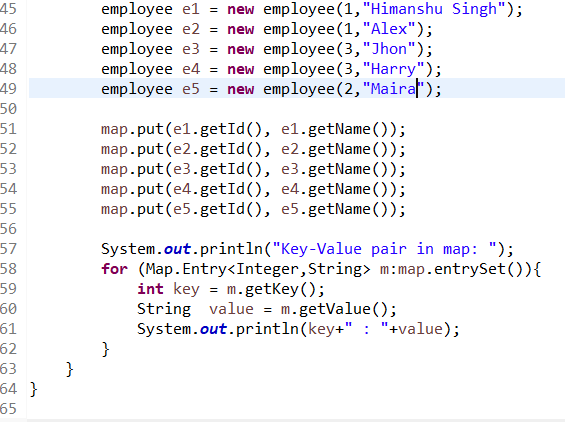
HashMap and HashSet use the hashcode value of an object to find out how the object would be stored in the collection, and subsequently hashcode is used to help locate the object in the collection. Hashing retrieval involves:

First, find out the right bucket using hashCode().

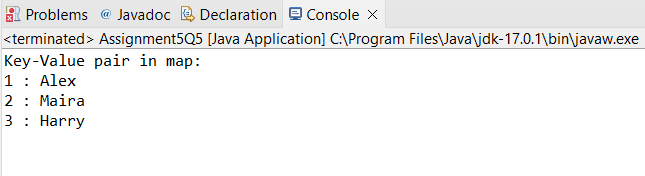
Secondly, search the bucket for the right element using equals().

**Code:**





**Output:**



**Q6. Implement the console based chatting using collections. Here are the options to be placed for to the user:**

**>java ChatApplication**

**Options:**

**A) Create a chatroom**

**B) Add the user**

**C) User login**

**D) Send a message**

**E) Display the messages from a specific chatroom**

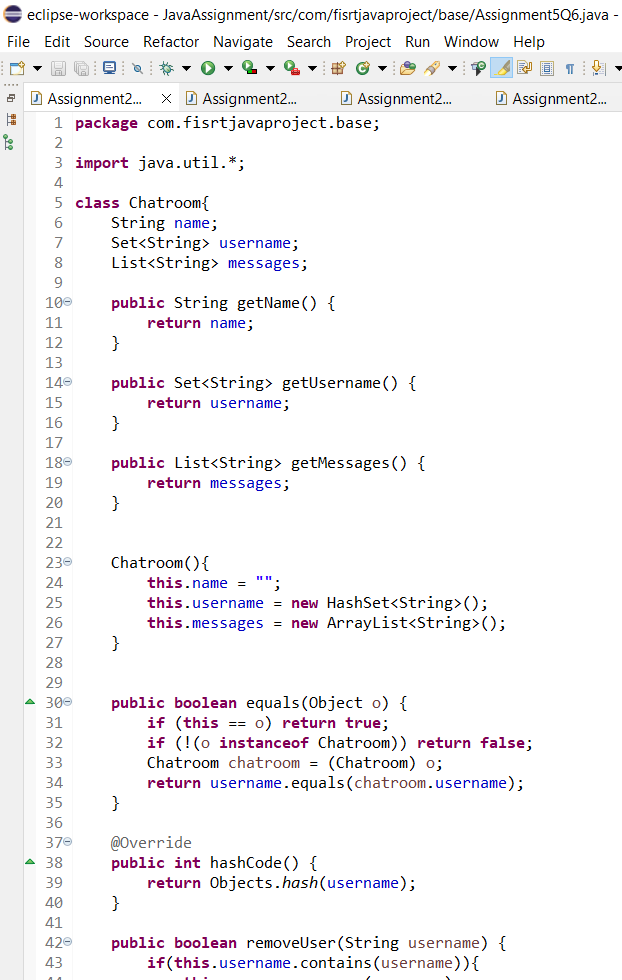
**F) List down all users belonging to the specified chat room.**

**G) Logout**

**H) Delete an user**

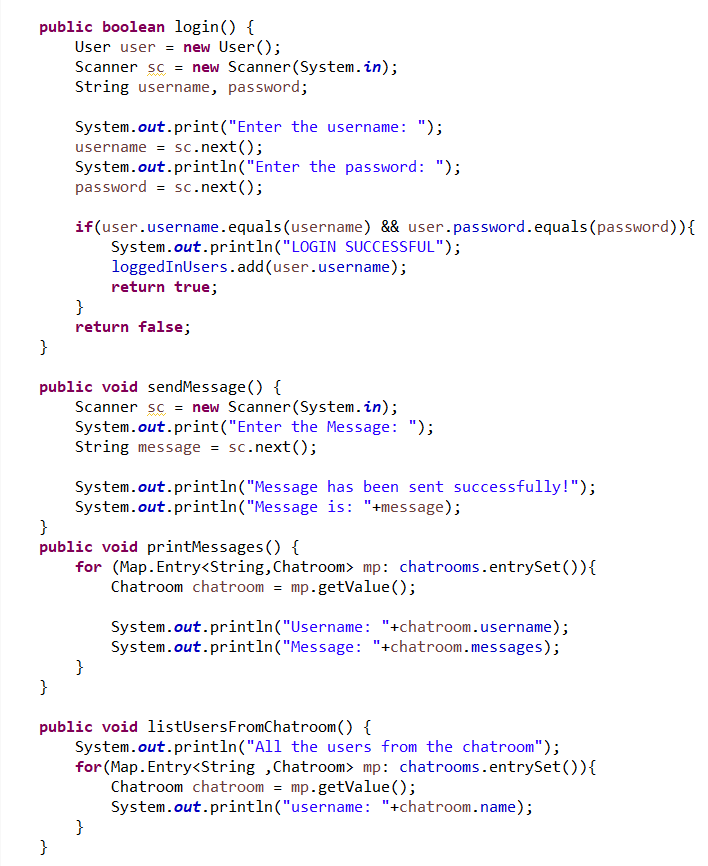
**I) Delete the chat room.**

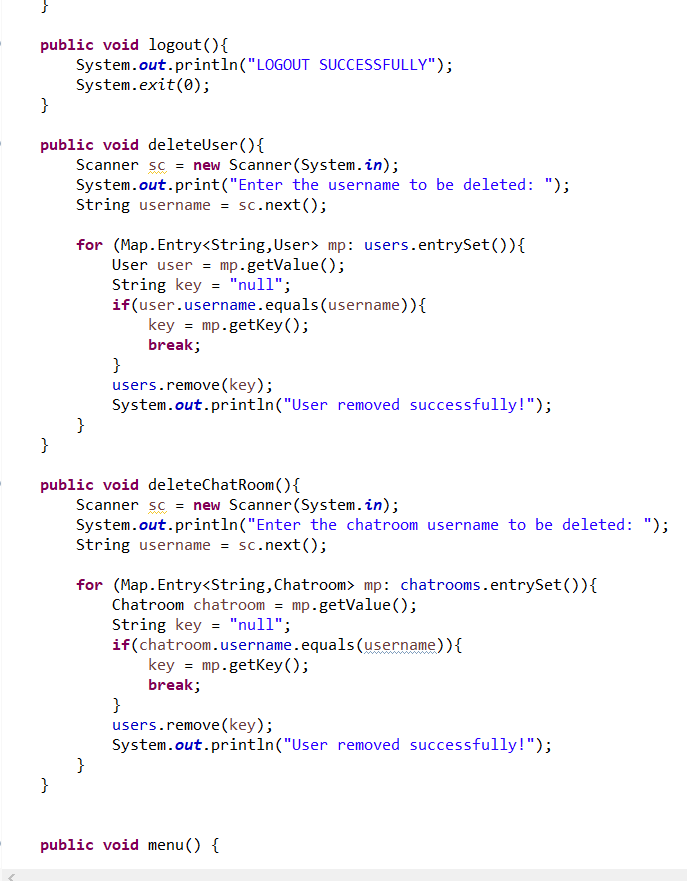
**Code:**





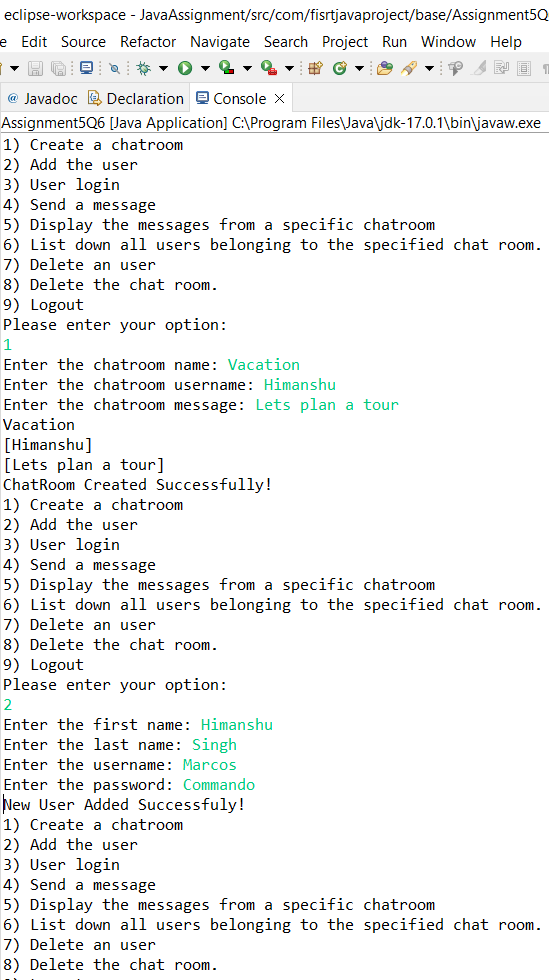


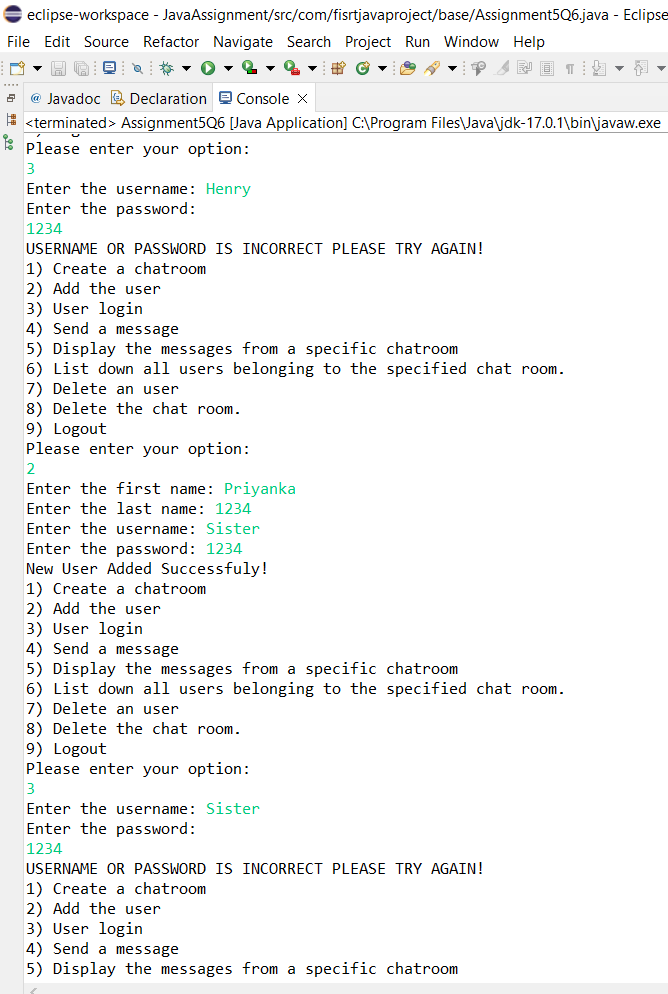


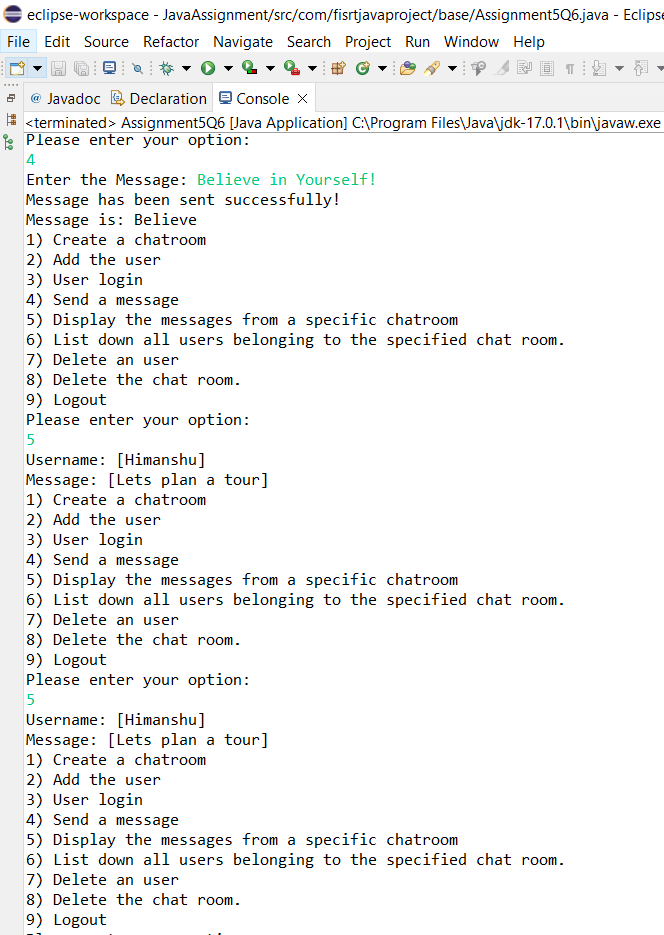


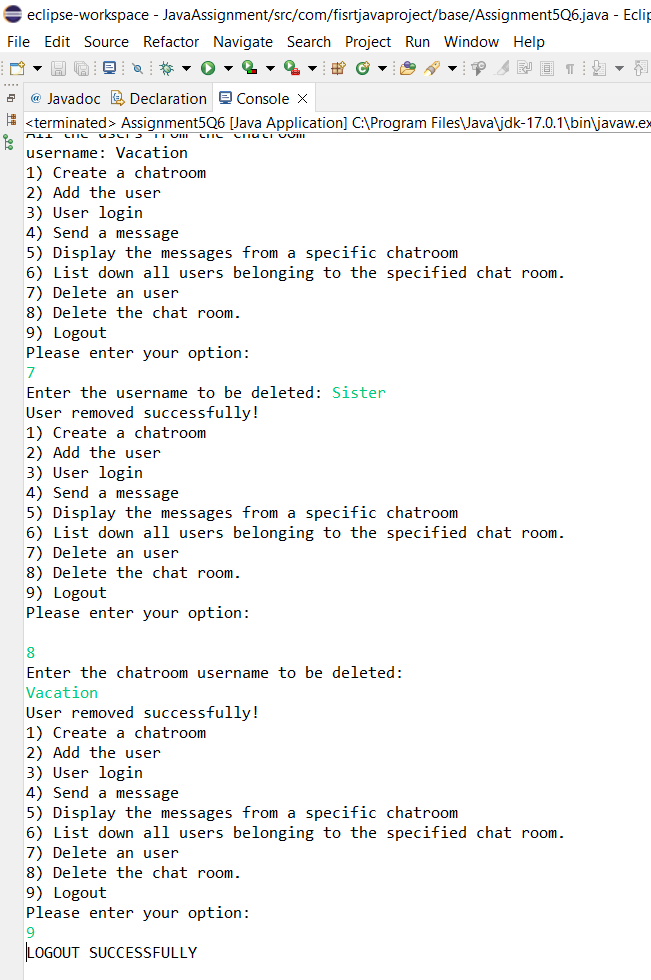


**Output :**









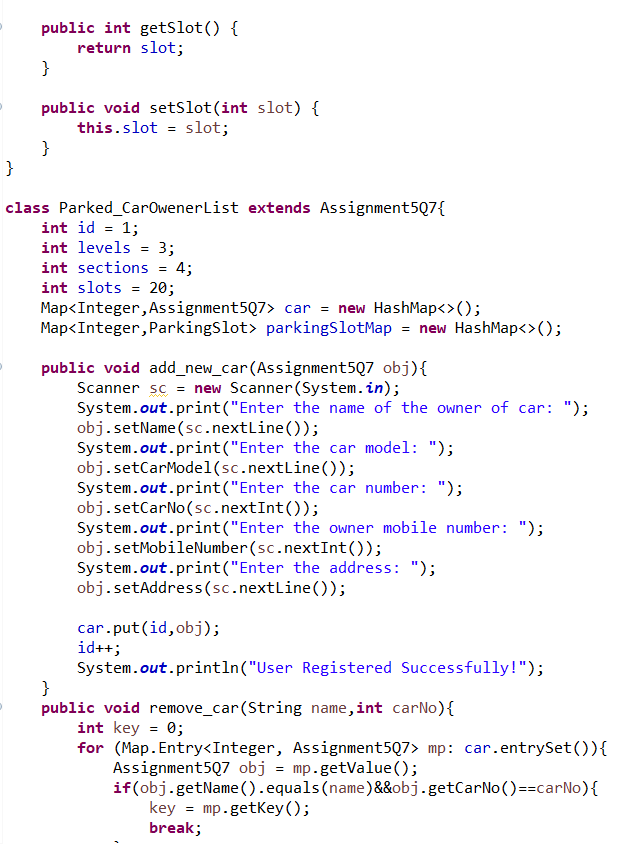
**Q7. There is parking slot available in R-Mall with 3 floors; each floor has 4 sections and each section can maximum park 20 cars. You need to design one application which will maintain all car details in such way when a car owner arrives to collect his care your application should provide details including where it is located.**

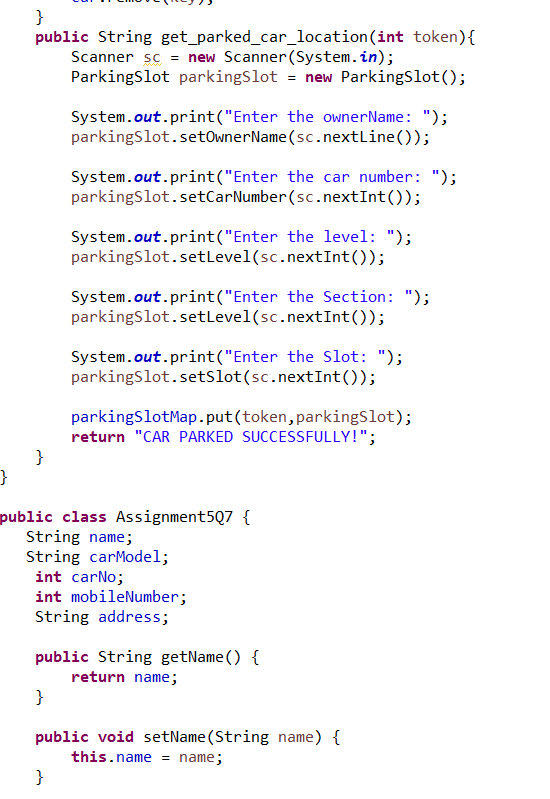
**a. Create class Parked\_CarOwner\_Details which will have field’s owerName, carModel, carNO, owerMobileNo, owerAddress with setter and getter methods.**

**b. Create class Parked\_CarOwenerList which will have method’s int add\_new\_car(Parked\_CarOwner\_Details obj), remove\_car(), get\_parked\_car\_location(token).**

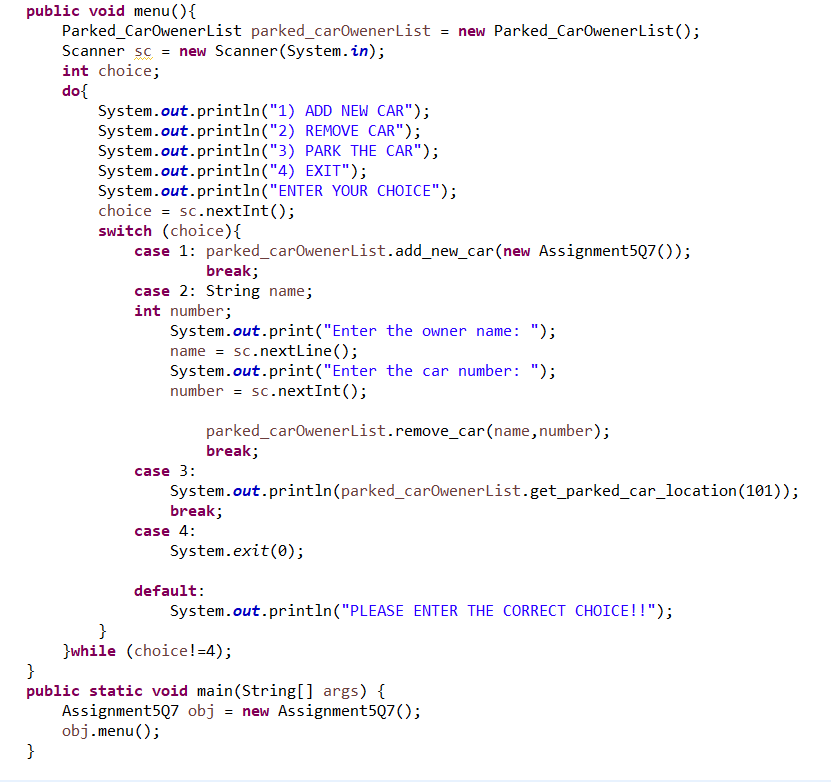
**Code :**



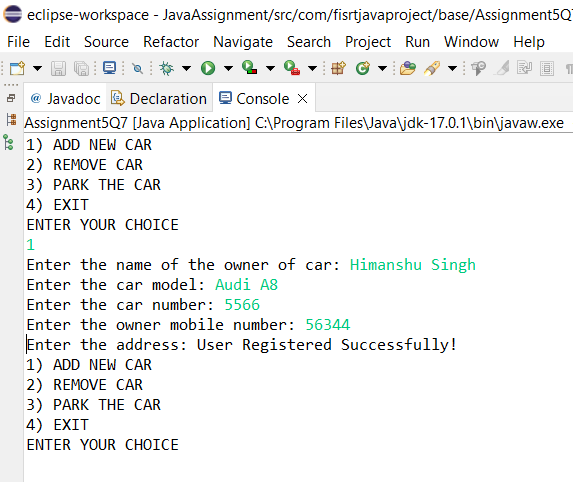








**Output :**

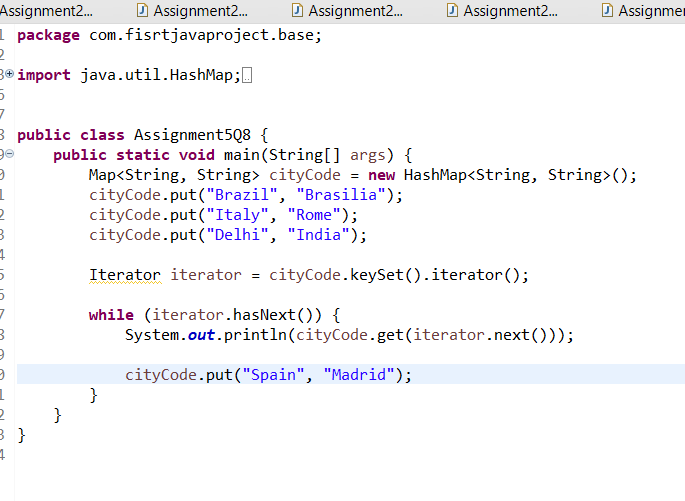


**Q8.1.  Test fail-fast iterators within multithreaded environment. Note example of fail fast iterator is Vector, ArrayList, HashSet etc. And fail-safe is ConcurrentHashMap, CopyOnWriteArrayList etc.**

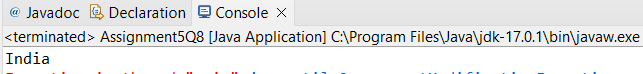
**Fail Fast:**

Iterators in java are used to iterate over the Collection objects.Fail-Fast iterators immediately throw ConcurrentModificationException if there is structural modification of the collection. Structural modification means adding, removing or updating any element from collection while a thread is iterating over that collection. Iterator on ArrayList, HashMap classes are some examples of fail-fast Iterator.

**Code:**



**Output:**

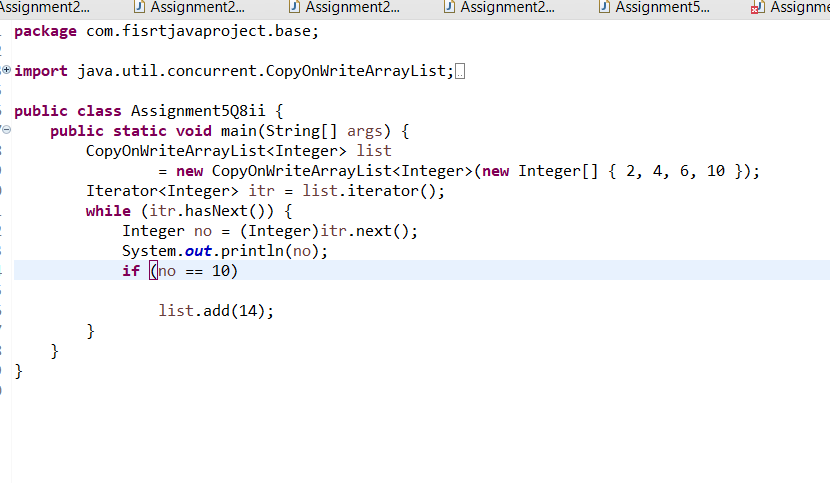


**Q8.2. Test fail-safe iterators within multithreaded environment. Note example of fail fast iterator is Vector, ArrayList, HashSet etc. And fail-safe is ConcurrentHashMap, CopyOnWriteArrayList etc.**

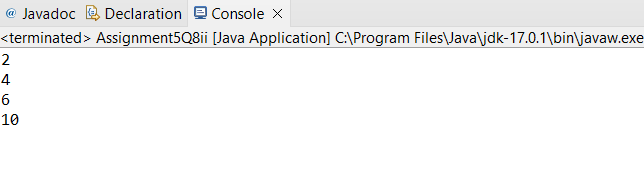
**Fail Safe:**

Fail-Safe iterators don’t throw any exceptions if a collection is structurally modified while iterating over it. This is because, they operate on the clone of the collection, not on the original collection and that’s why they are called fail-safe iterators. Iterator on CopyOnWriteArrayList, ConcurrentHashMap classes are examples of fail-safe Iterator.

**Code:**



**Output:**

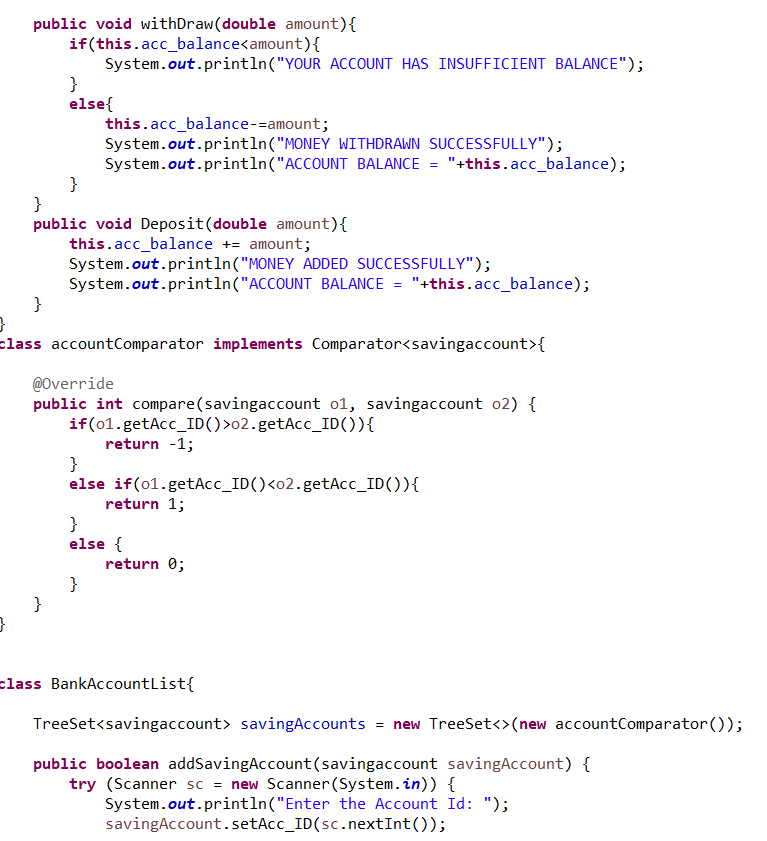


[**Q9**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2150)**. Create a Class SavingAccount with field’s acc\_balance, acc\_ID, accountHoldername, isSalaryAccount. Also add setter and getter methods with business method like withdraw and deposit.**

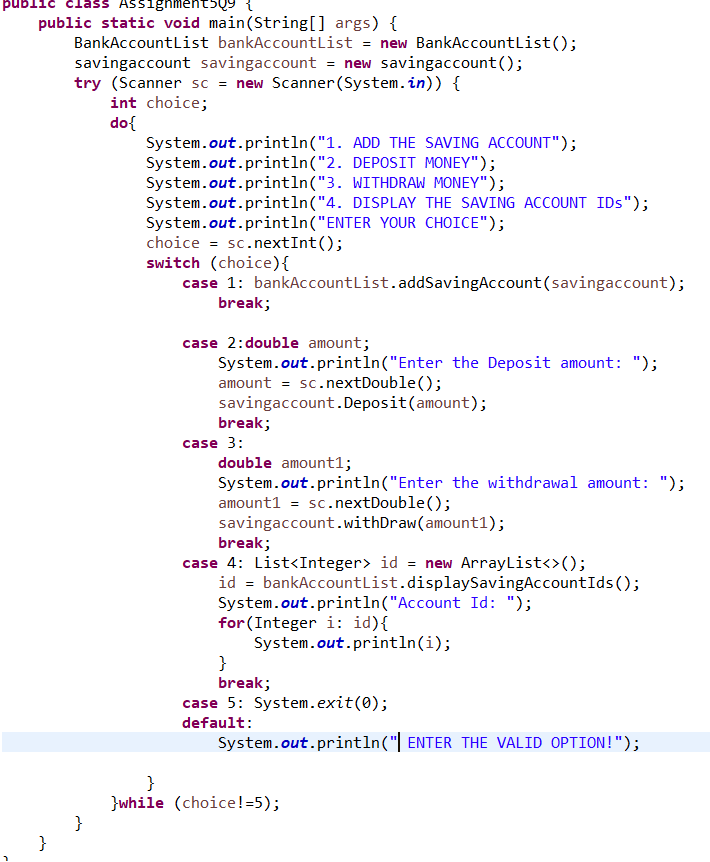
**a. Create class BankAccountList which will maintain SavingAccount objects. Ensure that this class should not allow duplicates as well as data should be displayed in sorted order. (as per acc\_ID)**

**Code:**









**Output:**

