**ASSIGNMENTS ON GENRICS**

1. Use a Hashset to hold employee objects. Upon running application, the details of the employees added to the HashSet should be displayed.

//class first

**import** java.util.HashSet;

**public** **class** first

{

**public** **static** **void** main(String[] args)

{

HashSet<Employee>set=**new** HashSet<Employee>();

Employee e=**new** Employee(632,"AJAY","SOFTWARE DEVLOPER",25000);

set.add(e);

**for**(Employee e1:set)

{

e.Displaydetails();

}

}

}

//class Employee

**public** **class** Employee

{

**int** id;

String name,dep;

**int** sal;

**public** Employee(**int** id, String name, String dep, **int** sal)

{

**super**();

**this**.id = id;

**this**.name = name;

**this**.dep = dep;

}

**public** **void** Displaydetails()

{

System.***out***.println("ID: "+id);

System.***out***.println("Name: "+name);

System.***out***.println("Department: "+dep);

System.***out***.println("Salary: "+sal);

}

}

**OUTPUT:**

ID: 632

Name: AJAY

Department: SOFTWARE DEVLOPER

Salary: 25000

2. Write an application to hold 10 random int values as key and random double values as values for HashMap. Print the data store in HashMap.

**import** java.util.HashMap;

**public** **class** second

{

**public** **static** **void** main(String[] args)

{

HashMap<Integer,Double> hm = **new** HashMap<>();

hm.put(45,45.2);

hm.put(56,21.52);

hm.put(96, 56.7);

hm.put(45,26.32);

hm.put(89,35.32);

hm.put(96,32.6);

hm.put(72,32.8);

hm.put(89,52.5);

hm.put(78,23.8);

hm.put(54, 87.6);

System.***out***.println(hm);

}

}

**OUTPUT:**

{96=32.6, 54=87.6, 56=21.52, 72=32.8, 89=52.5, 45=26.32, 78=23.8}

3. Write a generic method to exchange the positions of two different elements in an array.

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** third

{

**public** **static** **final** <V> **void** swap (V[] x, **int** i, **int** j)

{

V t = x[i];

x[i] = x[j];

x[j] = t;

}

**private** **static** **void** print()

{

String [] array = {"Hii", "Good", "Morning", "All"};

System.***out***.println("Original Array: "+Arrays.*toString*(array));

*swap*(array, 0, 3);

System.***out***.println("Array after swapping: "+Arrays.*toString*(array));

}

**public** **static** **void** main(String []a)

{

*print*();

}

}

**OUTPUT:**

Original Array: [Hii, Good, Morning, All]

Array after swapping: [All, Good, Morning, Hii]

4. Design a class named Pair which has two properties. The name of the first property is key and that of the second property is value. When designing the class take case of the following scenarios

a. Create an Object of Pair class to store String value for the property key and String value for the property value. Restriction Apart from String type no other types should be acceptable as Key or value input

e.g.

myObj.setKey("1"); myObj.setValue("Hello");

b. Create an object of the class Pair to storering value for the property key and java.util.Date as value for the property value

myObj.se y("To is"); myObj.setValue(new java.util.Date());

Note: In scenario a. no data apart from String should be used for key and value, in scenario b. no data apart from String for key and java.util.Date should be allowed

**import** java.util.\*;

**public** **class** pair

{

**public** **static** **void** main(String[] args)

{

HashMap<String,Integer> map=**new** HashMap<>();

map.put(1,"Hello");

System.***out***.println(map);

HashMap<String,Date> map1 = **new** HashMap<>();

map1.put("Today is", **new** java.util.Date());

System.***out***.println( map1);

}

}

**OUTPUT:**

{1=Hello}

{Today is=Mon Jan 17 12:20:59 PST 2022}