**Assignments on Generics**

1. Use a HashSet to hold Employee Objects. Upon running the application, the details of the employees added to the HashSet should be displayed.

Employee<<class>>

|--id

|--name

|--salary

|--department

|--displayDetails()

Feel Free to add properties and methods to Employee Class

**Note:** if we try to store any object other than Employee Object in Hashset, we should not be allowed it.

**import** java.util.HashSet;

**public** **class** Employee1 {

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

// **TODO** Auto-generated method stub

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

Employee e=**new** Employee(1,"Rob","Analyst",25000.0);

set.add(e);

**for**(Employee e1:set)

{

e.Displaydetails();

}

}

}

**public** **class** Employee {

**int** id;

String name,dept;

**double** sal;

**public** Employee(**int** id, String name, String dept, **double** sal)

{

**super**();

**this**.id = id;

**this**.name = name;

**this**.dept = dept;

**this**.sal = sal;

}

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

System.***out***.println(id+" "+name+" "+dept+" "+sal);

}

}

**Output:** 1 Rob Analyst 25000.0

1. Write an application to hold 10 random int value as keys and 10 random doubles values as values for a HashMap. Print the data store in the HashMap. **Note**: Keys can only be int and values double.

**import** java.util.HashMap;

**public** **class** Numbers {

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

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

map.put(1, 1.0);

map.put(2, 2.0);

map.put(3, 3.0);

map.put(4, 4.0);

map.put(5, 5.0);

map.put(6, 6.0);

map.put(7, 7.0);

map.put(8, 8.0);

map.put(9, 9.0);

map.put(10, 10.0);

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

**for**(Integer key: map.keySet()) {

System.***out***.println(key + " = " + map.get(key));

}

}

}

**Output:** {1=1.0, 2=2.0, 3=3.0, 4=4.0, 5=5.0, 6=6.0, 7=7.0, 8=8.0, 9=9.0, 10=10.0}

1 = 1.0

2 = 2.0

3 = 3.0

4 = 4.0

5 = 5.0

6 = 6.0

7 = 7.0

8 = 8.0

9 = 9.0

10 = 10.0

1. Write a generic method to exchange the position 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** Test {

**public** **static** **final** <T> **void** swap (T[] a, **int** i, **int** j) {

T t = a[i];

a[i] = a[j];

a[j] = t;

}

**public** **static** **final** <T> **void** swap (List<T> b, **int** i, **int** j) {

Collections.<T>*swap*(b, i, j);

}

**private** **static** **void** swap1() {

Integer [] a = {25, 20};

*swap*(a, 0, 1);

System.***out***.println("a:"+Arrays.*toString*(a));

List<Integer> b = **new** ArrayList<Integer>(Arrays.*asList*(a));

*swap*(b, 0, 1);

System.***out***.println("b:"+b);

}

**public** **static** **void** main(String...args)

{

*swap1*();

}

}

**Output:** a:[20, 25]

b:[25, 20]

1. Design a class **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 class of the following scenarios:
2. 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”);

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

**public** **class** Pair1 {

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

// **TODO** Auto-generated method stub

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

map.put("Hello", "1");

map.put("World", "2");

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

}

}

**Output:** {Hello=1, World=2}

1. Create an object of the class Pair to store String value for the property key and java.util.Date as value for the property

myObj,setKey(“Today is”);

myObj.setValue(new java.uti.Data());

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

**public** **class** Pair2 {

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

// **TODO** Auto-generated method stub

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

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

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

}

}

**Output:** {Today is =Sun Jan 16 20:11:40 IST 2022}

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.Data should be allowed.