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 method to employee class

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

Ans🡪

**package** org.generic.sets;

**import** java.util.HashSet;

**public** **class** example {

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

HashSet<emp> hs = **new** HashSet<emp>();

emp e = **new** emp(1, "ganu", "admin", 3000.25);

hs.add(e);

**for**(emp e1:hs)

{

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

}

}

}

**package** org.generic.sets;

**public** **class** emp {

**int** id;

String name, dep;

**double** sal;

**public** emp(**int** id, String name, String dep, **double** sal) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.dep = dep;

**this**.sal = sal;

}

}

Output:

1 ganu admin 3000.25

1. Write an application hold 10 random int values as keys and 10 random double values as values for a HashMap. Print the data store in the HashMap.

Ans🡪

**package** org.generic.sets;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** Mapp {

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

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

map.put(1, 1.23);

map.put(5, 5.26);

map.put(12, 12.75);

map.put(15, 15.23);

map.put(22, 22.45);

map.put(28, 28.28);

map.put(36, 36.15);

map.put(45, 45.25);

map.put(55, 55.85);

map.put(60, 60.23);

System.***out***.println("HashMap: ");

**for**(Map.Entry m:map.entrySet())

{

System.***out***.println(m.getKey()+" "+m.getValue());

}

}

}

Output:

HashMap:

1 1.23

36 36.15

5 5.26

22 22.45

55 55.85

12 12.75

28 28.28

60 60.23

45 45.25

15 15.23

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

Ans🡪

**package** org.generic.sets;

**import** java.util.Arrays;

**public** **class** Swap1 {

**public** **static** <T> **void** swap(**int**[] arr, **int** i1, **int** i2) {

**int** temp = arr[i2];

arr[i2]=arr[i1];

arr[i1]=temp;

}

**private** **static** **void** swap3() {

**int**[] arr = {1,2,5,8,9};

*swap*(arr,0,3);

System.***out***.println("Swapped"+Arrays.*toString*(arr));

}

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

*swap3*();

}

}

Output:

Swapped[8, 2, 5, 1, 9]

1. 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 tale case 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.

Ans🡪

**package** org.generic.sets;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** ExampleKeyValue {

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

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

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

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

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

**for**(Map.Entry m:map.entrySet())

{

System.***out***.println(m.getKey()+" "+m.getValue());

}

}

}

Output:

{1=Mango, 2=Orange}

1 Mango

2 Orange

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 value

Ans🡪

**package** org.generic.sets;

**import** java.util.Date;

**import** java.util.HashMap;

**public** **class** ExampleDate {

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

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

hs.put("Today is:",**new** Date());

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

}

}

Output:

{Today is:=Tue Jan 18 13:49:30 IST 2022}