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

Ans)

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

**int** id,salary;

String name,department;

**public** Employee(**int** id, **int** salary, String name, String department) {

**this**.id = id;

**this**.salary = salary;

**this**.name = name;

**this**.department = department;

}

**int** getSalary() {

**return** salary;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String deparment) {

**this**.department = deparment;

}

**public** **void** displayDetails() {

System.***out***.printf("id :"+getId()+" salary: "+getSalary()+" name : "+getName()+" Department: "+getDepartment());

}

**public** **class** hashset {

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

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

Employee e1=**new** Employee(123,25000,"Manu","IT");

set.add(e1);

e1.displayDetails();

}

}

}

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

Ans

**import** java.util.HashMap;

**import** java.util.Map.Entry;

**public** **class** Main {

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

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

map.put(123,1.23);

map.put(234,33.3);

map.put(67,23.3);

map.put(56,78.2);

map.put(65,7.2);

map.put(14,6.2);

map.put(6,8.2);

map.put(45,7.3);

map.put(24,4.5);

map.put(12,4.2);

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

// for(Entry<Integer,Double>entry:map.entrySet()) {

// Integer i=entry.getKey();

// Double d=entry.getValue();

// System.out.println("key "+i+" value "+d);

// }

}

}

3)Write a generic method to exchange the position of two different elements in an array

Ans)

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** Test2 {

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

T temp=a[i];

a[i]=a[j];

a[j]=temp;

}

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

String[] s= {"1","2","3","4"};

System.***out***.println("Before swapping " +Arrays.*toString*(s));

*swap*(s,0,3);

System.***out***.println("After swapping: "+Arrays.*toString*(s));

}

}

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

1. 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 accepted as Key or value input.

**import** java.util.Date;

**public** **class** Pair<T,V> {

**private** T Key;

**private** V Value;

**public** T getKey() {

**return** Key;

}

**public** **void** setKey(T key) {

Key = key;

}

**public** V getValue() {

**return** Value;

}

**public** **void** setValue(V value) {

Value = value;

}

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

Pair<String,Date> myObj=**new** Pair<>();

myObj.setKey("1");

myObj.setValue("Hello");

String key=myObj.getKey();

String value=myObj.getValue();

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

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

}

}

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

myObj.setKey(“Todays”)

myObj.setValue(“new java.util.Date”)

Ans)

myObj.setKey("Today is");

myObj.setValue(**new** java.util.Date());

String key=myObj.getKey();

Date value=myObj.getValue();

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

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

}