Part-A

1.Develop a java application for data management about sales of different products in a company 'x' has different stores and each store maintains different products and their prices data, for efficient data management the company maintains product and stores mapping in the form of records, it maintains data for efficient retrieval in the form of store id as key and collections of product id and its price as key values pairs

Eg:{ ["store A": {["apple":100],["banana":50]} , "store B": {["grapes":200]} ….]};

Once the data stored in database, run the following queries

1. **If a store id is given then retrieve and print all the products and their prices.**
2. **If a store id is given then find the product which has highest price.**
3. **If the product name is given then find and display all the stores it present.**

```
import java.util.*;

public class A1 {
   HashMap<String,HashMap<String,Integer>> hm=new HashMap<>();

   static void option1(HashMap<String,HashMap<String,Integer>> hm)
   {
      System.out.println("option1 of query");
      Scanner sc=new Scanner(System.in);
      System.out.println("enter store");
      String key1=sc.nextLine();
     // sc.nextLine();
```

```
    HashMap<String,Integer> temp= hm.get(key1);
    if(temp==null)
    {
        System.out.println("store does not exist");
        return;
    }

    System.out.println(temp);
}
static void option2(HashMap<String,HashMap<String,Integer>> hm)
{
    System.out.println("option2 of query");
    Scanner sc=new Scanner(System.in);
    System.out.println("enter store");
    String key1=sc.nextLine();
    // sc.nextLine();

    HashMap<String,Integer> temp= hm.get(key1);
    if(temp==null)
    {
        System.out.println("store does not exist");
        return;
    }

    String res="";
    int max=0;

    for(String s:temp.keySet())
    {
        if(max<temp.get(s))
        {
            res=s;
            max=temp.get(s);
        }
    }
```

```
        System.out.println("product having highest price is "+res+" of price
"+max);

    }
    static void option3(HashMap<String,HashMap<String,Integer>> hm)
    {
        System.out.println("option3 of query");
        Scanner sc=new Scanner(System.in);
        System.out.println("enter product");
        String key2=sc.nextLine();

        System.out.println("product "+key2+" present in stores:  ");
        for(String s1:hm.keySet())
        {
            HashMap<String,Integer> temp= hm.get(s1);
            if(temp.containsKey(key2))
            {
                System.out.print(s1+", ");
            }
        }
        System.out.println();


    }
    void query()
    {
        Scanner sc=new Scanner(System.in);
        int n=0;

        System.out.println("enter option");
        System.out.println("1: to retrieve a store items ");
        System.out.println("2: highest product price of given store");
        System.out.println("3: to display stores that contains given product");
        n=sc.nextInt();
        if(n<0||n>3) System.out.println("try again");

        if(n==1)
        {
            option1(hm);
        }
        else if(n==2)
        {
            option2(hm);
        }
        else {
            option3(hm);
        }

    }


    void dataStore()
    {
        Scanner sc=new Scanner(System.in);
        String key1="";
        String key2="";
        int value=0;

        System.out.println("enter store");
        key1=sc.nextLine();
        //sc.nextLine();

        System.out.println("enter product");
        key2=sc.nextLine();
        //sc.nextLine();

        System.out.println("enter price");
        value=sc.nextInt();
        if(hm.containsKey(key1))
        {
            HashMap<String,Integer> temp= hm.get(key1);
            temp.put(key2,value);
        }
        else
        {
            HashMap<String,Integer> hm1=new HashMap<>();
            hm1.put(key2,value);
            hm.put(key1,hm1);
        }

        System.out.println(hm);
    }

    public static void main(String[] args) {

           A1 a=new A1();



        //HashMap<String,Integer> hm1=new HashMap<>();
        int n=0;

        Scanner sc=new Scanner(System.in);

        do {
            System.out.println("Options: ");
            System.out.println("1: data store ");
            System.out.println("2: query");
            System.out.println("3: exit");
            n=sc.nextInt();
            if(n>3||n<0) continue;
```

```
            if(n==1)
            {
               System.out.println("data store");
               a.dataStore();

            }
            else if(n==2)
            {
               System.out.println("query");
               a.query();
            }
         }while(n!=3);
      }
   }
```

2. sorting of employee records based on different parameters according to requirements create and store employee records as collections of objects and the record format is employee name, salary and DOJ

```
import java.util.*;

public class A2 {
    ArrayList<String[]> al=new ArrayList<>();

    static void sort1(ArrayList<String[]> al)
    {
        System.out.println("sal asc");
        Collections.sort(al,(a,b)->
        {
            if(Integer.parseInt(a[1])<Integer.parseInt(b[1])) return -1;
            else return 1;
        }
        );

        for(String s[]:al)
        {
            System.out.println("emp id: "+ s[0]+", emp sal: "+s[1]+", emp doj:
"+s[2]);
        }



    }
    static void sort2(ArrayList<String[]> al)
    {
        System.out.println("sal desc");
        Collections.sort(al,(a,b)->
                {
                    if(Integer.parseInt(a[1])<Integer.parseInt(b[1])) return 1;
                    else return -1;
                }
        );
        for(String s[]:al)
        {
            System.out.println("emp id: "+ s[0]+", emp sal: "+s[1]+", emp doj:
"+s[2]);
        }

    }
    static void sort3(ArrayList<String[]> al)
    {
        System.out.println("doj asc");
        Collections.sort(al,(a,b)->
        {
            String s1=a[2];
            String s2=b[2];
            String[] s11=s1.split("-");
            String[] s22=s2.split("-");

            int y1=Integer.parseInt(s11[2]);
            int y2=Integer.parseInt(s22[2]);

            if(y1<y2)
            {
                return -1;
            }
            else if(y1>y2) return 1;

            int m1=Integer.parseInt(s11[1]);
            int m2=Integer.parseInt(s11[1]);
            if(m1<m2)
            {
                return -1;
            }
            else if(m1>m2) return 1;
            int d1=Integer.parseInt(s11[0]);
            int d2=Integer.parseInt(s22[0]);
            if(d1<d2)
            {
                return -1;
            }
            else if(d1>d2) return 1;

            else return 0;
        });
        for(String s[]:al)
        {
            System.out.println("emp id: "+ s[0]+", emp sal: "+s[1]+", emp doj:
"+s[2]);
        }
    }
    static void sort4(ArrayList<String[]> al)
    {
        System.out.println("doj desc");
        Collections.sort(al,(a,b)->
        {
            String s1=a[2];
            String s2=b[2];
            String[] s11=s1.split("-");
            String[] s22=s2.split("-");

            int y1=Integer.parseInt(s11[2]);
            int y2=Integer.parseInt(s22[2]);

            if(y1<y2)
            {
                return 1;
            }
            else if(y1>y2) return -1;

            int m1=Integer.parseInt(s11[1]);
            int m2=Integer.parseInt(s11[1]);
            if(m1<m2)
            {
                return 1;
            }
            else if(m1>m2) return -1;


            int d1=Integer.parseInt(s11[0]);
            int d2=Integer.parseInt(s22[0]);
            if(d1<d2)
            {
                return 1;
            }
            else if(d1>d2) return -1;

            else return 0;
        });
        for(String s[]:al)
        {
            System.out.println("emp id: "+ s[0]+", emp sal: "+s[1]+", emp doj:
"+s[2]);
        }
    }

    public static void main(String[] args) {
        A2 a=new A2();
        Scanner sc=new Scanner(System.in);
        int n=1;
        while(true){
            System.out.println("enter 1 to add record");
            n=sc.nextInt();
            if(n==1)
            {
                sc.nextLine();
                System.out.println("enter employe id");
                String s1=sc.nextLine();

                System.out.println("enter salary");
                String s2=sc.nextLine();

                System.out.println("enter DOJ");
                String s3=sc.nextLine();

                a.al.add(new String[]{s1,s2,s3});

            }
            else
            {
                break;
            }
        }

        do {
            System.out.println("enter option");
            System.out.println("1: salary ASC");
            System.out.println("2: salary DESC");
            System.out.println("3: DOJ ASC");
            System.out.println("4: DOJ DESC");
            System.out.println("5: exit");
            n=sc.nextInt();
            if(n<0||n>5) continue;

            if(n==1)
            {
                sort1(a.al);
```

```
            }
            else if(n==2)
            {
                sort2(a.al);
            }
            else if(n==3)
            {
                sort3(a.al);
            }
            else if(n==4)
            {
                sort4(a.al);
            }
        }while(n!=5);

    }
}
```

Part-B

1) **What will be the output of the following Java code?**

```
class exception_handling
{
    public static void main(String args[])
    {
        try
        {
            int a = args.length;
            int b = 10 / a;
            System.out.print(a);
            try
            {
                if (a == 1)
                    a = a / a - a;
                if (a == 2)
                {
                    int []c = {1};
                    c[8] = 9;
                }
            }
            catch (ArrayIndexOutOfBoundException e)
            {
                System.out.println("TypeA");
            }
            catch (ArithmeticException e)
            {
                System.out.println("TypeB");
            }
        }
    }
}
```

(a) TypeA
(b) TypeB
(c) Compile Time Error
(d) 0TypeB

Ans: b (if no runtime arguments given)

2) **What will be the output of the following Java program?**

```
interface calculate
{
   void cal(int item);
}
class display implements calculate
{
   int x;
   public void cal(int item)
   {
      x = item * item;
   }
}
class interfaces
{
   public static void main(String args[])
   {
      display arr = new display;
      arr.x = 0;
      arr.cal(2);
      System.out.print(arr.x);
   }
}
```

(a) 0
(b) 2
(c) 4
(d) None of the mentioned

Ans: d

3) **What will be the output of following code?**

```
class DThread implements Runnable
{
  public void run()
  {
    System.out.println("Run called");
  }
}
class DemoThread
{
  public static void main(String[] args)
  {
    Thread  t = new Thread();
    t.start();
    System.out.println("Main method");
  }
}
```

(a)
Run called

(b)
Main method

(c)
Run called Main method

(d)
Compilation error

Ans: d

4) Which of following is true about notifyAll()?

(notifyAll()

(a)

Wakes up all thread that are waiting on this object's monitor

(b)

Wakes up all thread that are not waiting on this object's monitor

(c)

Wakes up only one thread that are waiting on this object's monitor

(d)

Wakes up only two thread that are waiting on this object's monitor

Ans: a

5) **What is the output of the following program?**

```
import java.util.*;

public class priorityQueue {
  public static void main(String[] args)
  {
    PriorityQueue<Integer> queue
      = new PriorityQueue<>();
    queue.add(11);
    queue.add(10);
    queue.add(22);
    queue.add(5);
    queue.add(12);
    queue.add(2);

    while (queue.isEmpty() == false)
      System.out.printf("%d ", queue.remove());

    System.out.println("\n");
  }
}
```

a) 11 10 22 5 12 2

b) 2 12 5 22 10 11

c) 2 5 10 11 12 22

d) 22 12 11 10 5 2

Ans: d