

## Assignment-8

### Collections-2

**Q1.** In the following Java program, two PriorityQueuees are created—one for Integer and one for Character with a reverse order comparator.

```
public class PriorityQueueImplementation {

    public static void main(String[] args)
    {

        PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();
        pQueue.add(10);
        pQueue.add(20);
        pQueue.add(15);
        pQueue.add(5);
        pQueue.add(25);
        pQueue.add(30);
        pQueue.add(20);

        PriorityQueue<Character> pqc = new PriorityQueue<>(Comparator.reverseOrder());
        pqc.add('B');
        pqc.add('C');
        pqc.add('A');
        pqc.add('B');
        pqc.add('b');
        pqc.add('a');
        pqc.add('d');
        pqc.add('c');

        System.out.println("Character Priority Queue: " + pqc);

        System.out.println("Integer Priority Queue: " + pQueue);

        pqc.offer('F');
        pqc.remove('d');
        System.out.println("Character Priority Queue after applying offer & remove: " + pqc);

        System.out.println("Priority Queue After Poll: " + pqc);

        System.out.println("Priority Queue After Peek: " + pqc);

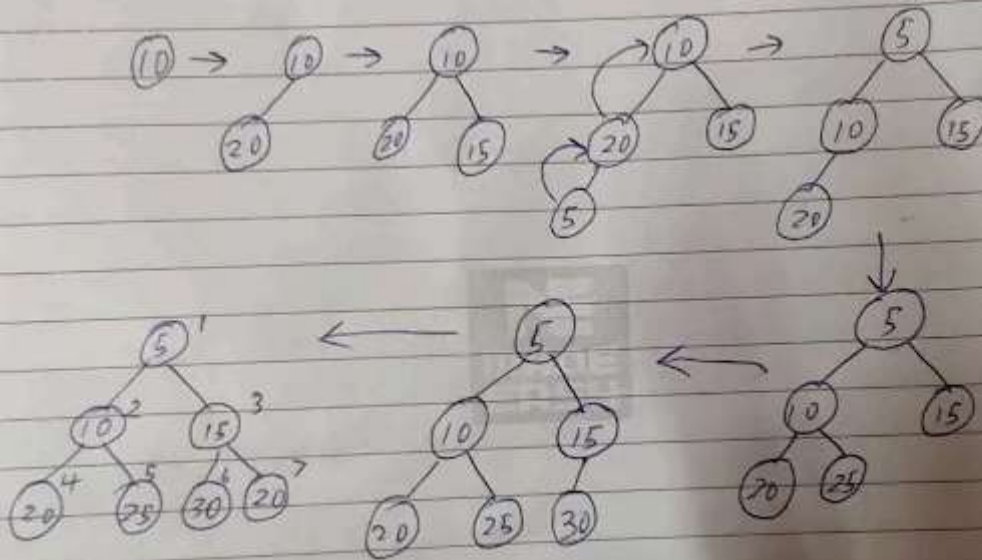
        System.out.println("Priority Queue After Element: " + pqc);
    }
}
```



8 Priority queue makes min heap and ~~return~~ return it when priority queue object is called.

Not. added →

10, 20, 15, 5, 25, 30, 20

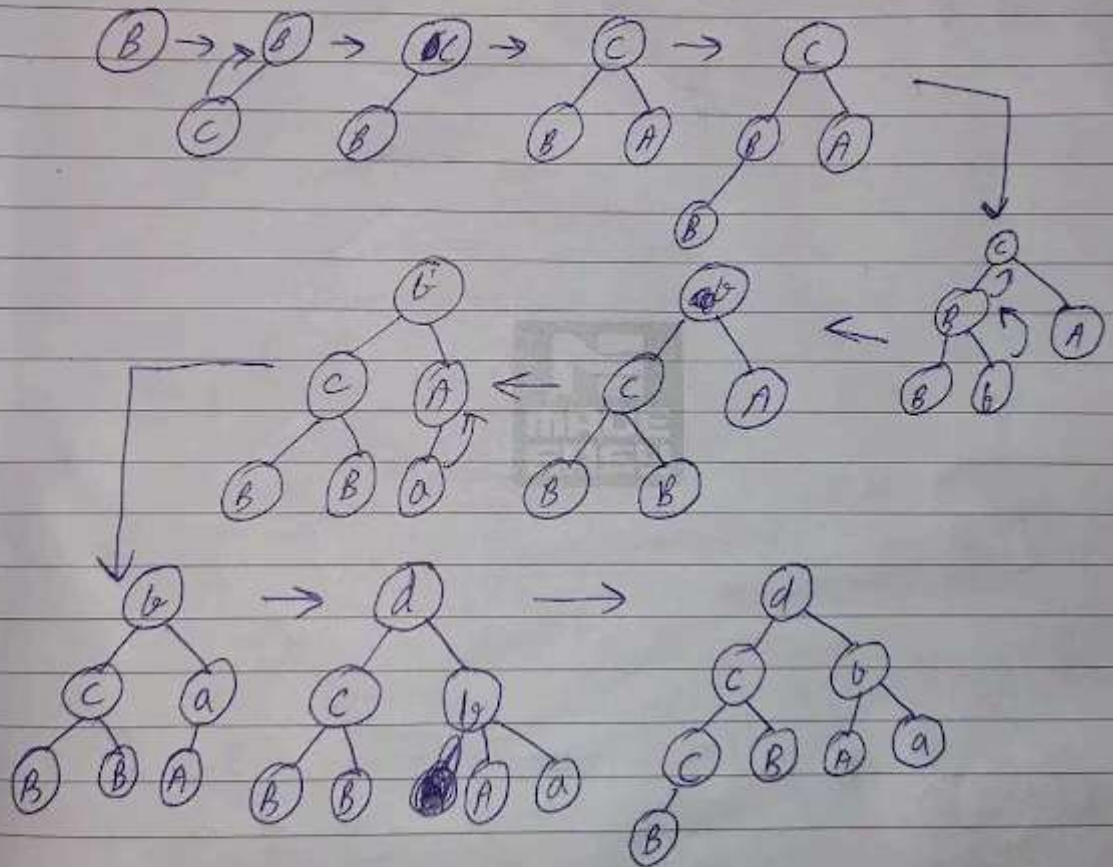


[ 5, 10, 15, 20, 25, 30, 20 ]

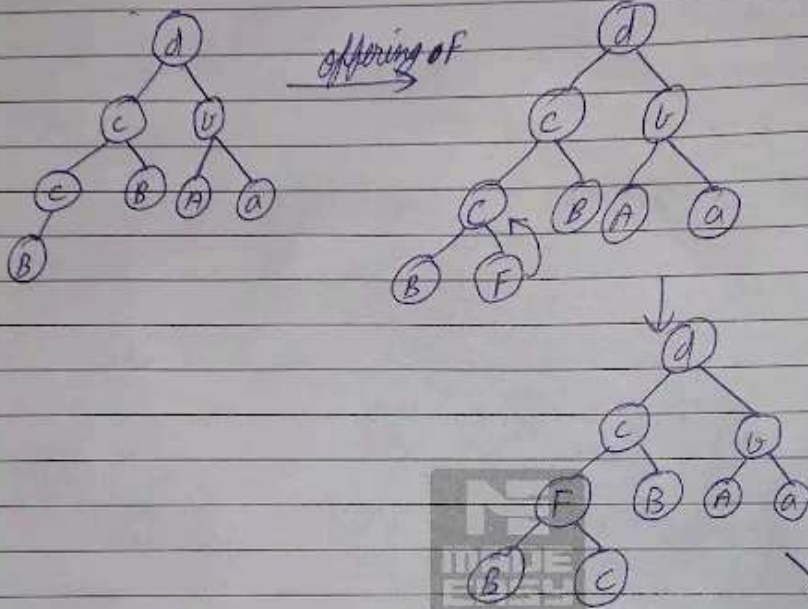


Characters added  $\rightarrow$

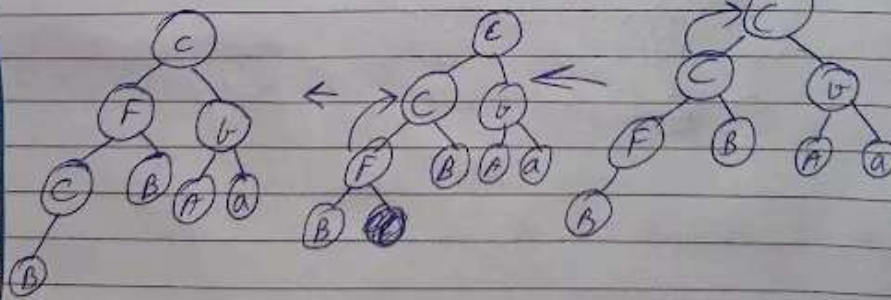
B, C, A, B, b, a, d, c  
as we have used comparator to reverse order the priority key will make max heap.



[d, C, b, C, B, A, a, B]

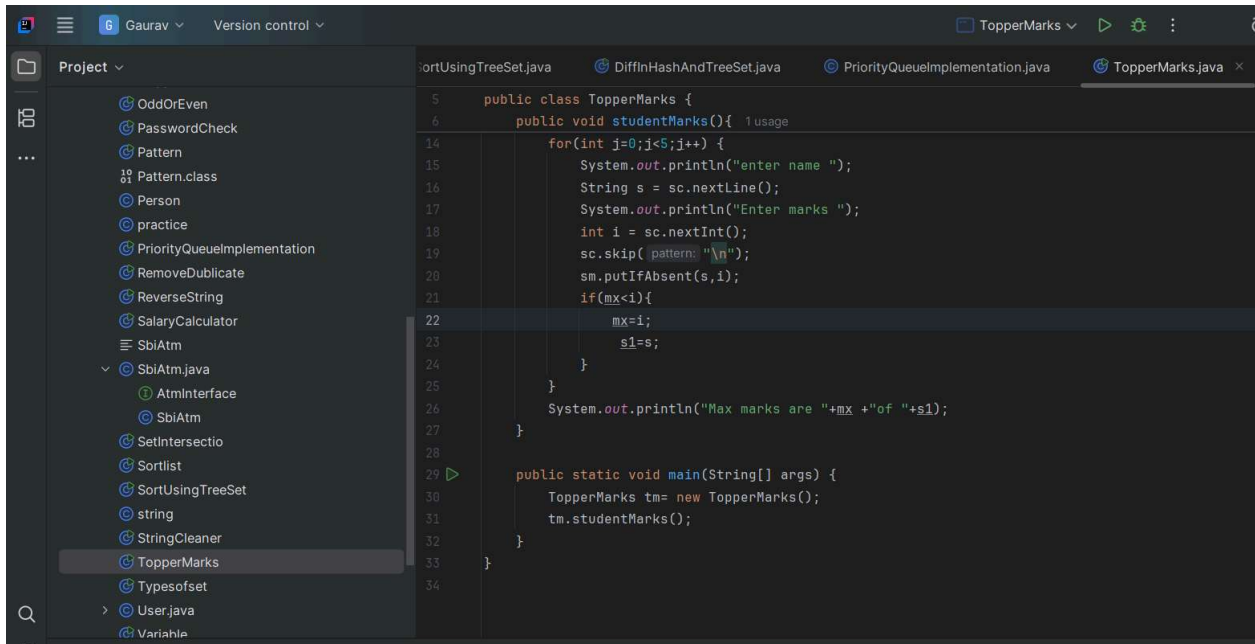


[d, c, b, F, B, A, a, B, C] Remove d

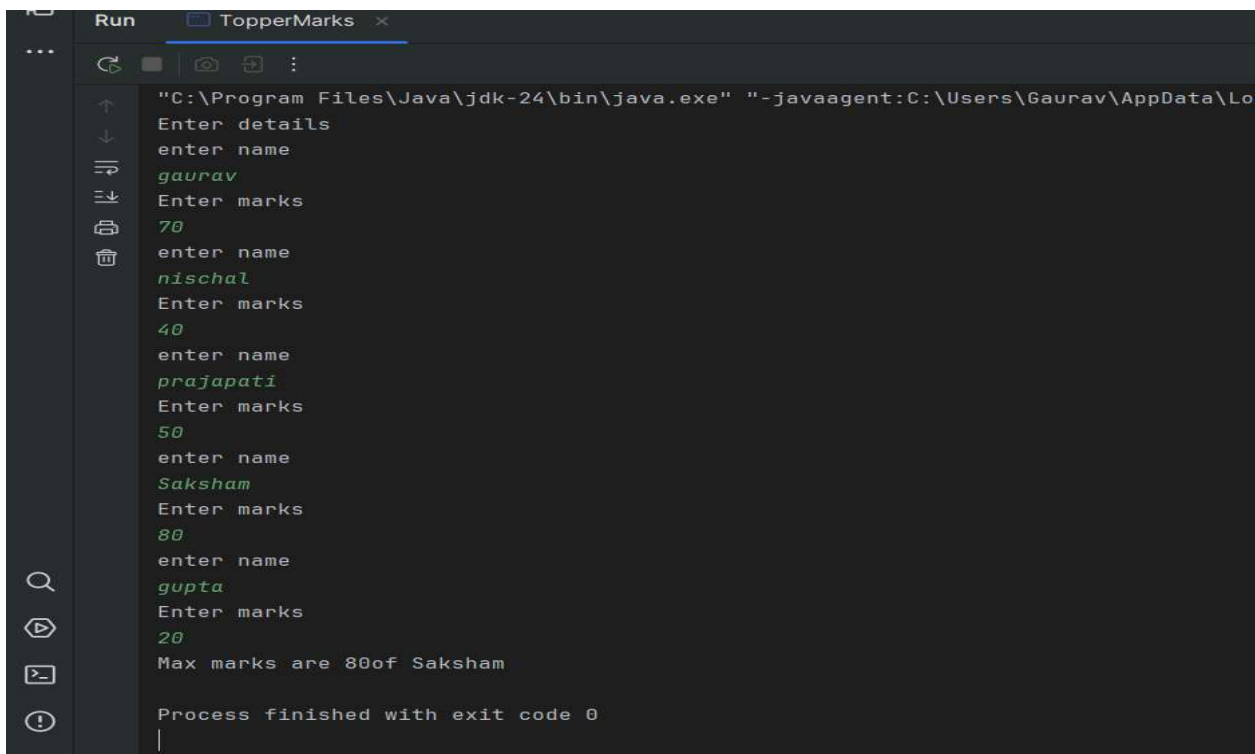


[c, F, b, C, B, A, a, B]

Q2.WAP that stores student names and their marks, adds 5 students, and print the topper student name(i.e., the one with the highest marks):



```
5 public class TopperMarks {
6     public void studentMarks(){ 1 usage
14         for(int j=0;j<5;j++) {
15             System.out.println("enter name ");
16             String s = sc.nextLine();
17             System.out.println("Enter marks ");
18             int i = sc.nextInt();
19             sc.skip( pattern: "\n");
20             sm.putIfAbsent(s,i);
21             if(mx<i){
22                 mx=i;
23                 s1=s;
24             }
25         }
26         System.out.println("Max marks are "+mx +"of "+s1);
27     }
28
29     public static void main(String[] args) {
30         TopperMarks tm= new TopperMarks();
31         tm.studentMarks();
32     }
33 }
34
```



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Lo
Enter details
enter name
gaurav
Enter marks
70
enter name
nischal
Enter marks
40
enter name
prajapati
Enter marks
50
enter name
Saksham
Enter marks
80
enter name
gupta
Enter marks
20
Max marks are 80of Saksham

Process finished with exit code 0
|
```



Q3. Write a method that returns a Map of character frequency in a string.

Take Input as: hello world

Output should be:

h : 1

e : 1

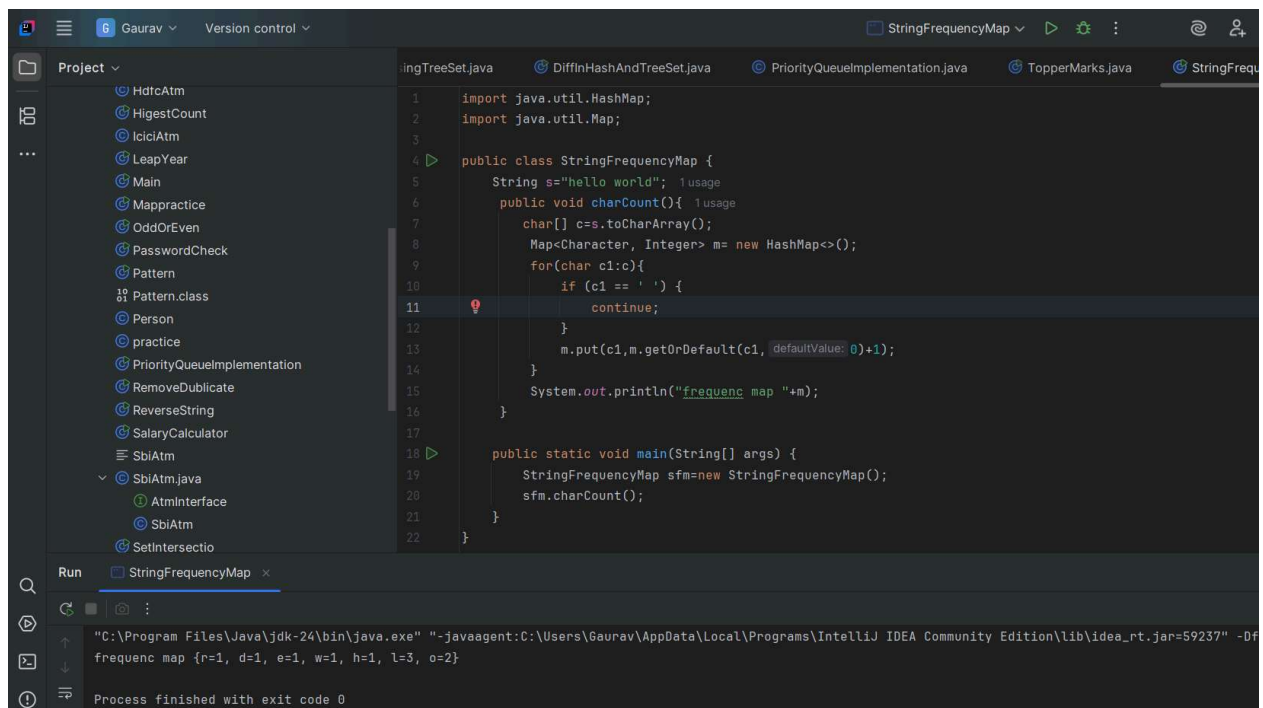
l : 3

o : 2

w : 1

r : 1

d : 1



The screenshot shows an IDE with a project named 'Gaurav'. The 'Project' view on the left lists various Java files, including 'SbiAtm.java'. The main editor displays the code for 'StringFrequencyMap.java'. The code imports 'HashMap' and 'Map' from 'java.util'. It defines a class 'StringFrequencyMap' with a static string 's = "hello world"', a 'charCount()' method, and a 'main()' method. The 'charCount()' method converts the string to a character array, iterates through it, and updates a 'HashMap' with character frequencies. The 'main()' method creates an instance of 'StringFrequencyMap' and calls 'charCount()'. The 'Run' view at the bottom shows the command executed and the output: 'frequenc map {r=1, d=1, e=1, w=1, h=1, l=3, o=2}'. The process finished with exit code 0.

```
1 import java.util.HashMap;
2 import java.util.Map;
3
4 public class StringFrequencyMap {
5     String s="hello world"; //usage
6     public void charCount(){ //usage
7         char[] c=s.toCharArray();
8         Map<Character, Integer> m= new HashMap<>();
9         for(char c1:c){
10             if (c1 == ' ') {
11                 continue;
12             }
13             m.put(c1,m.getOrDefault(c1, defaultValue: 0)+1);
14         }
15         System.out.println("frequenc map "+m);
16     }
17
18     public static void main(String[] args) {
19         StringFrequencyMap sfm=new StringFrequencyMap();
20         sfm.charCount();
21     }
22 }
```

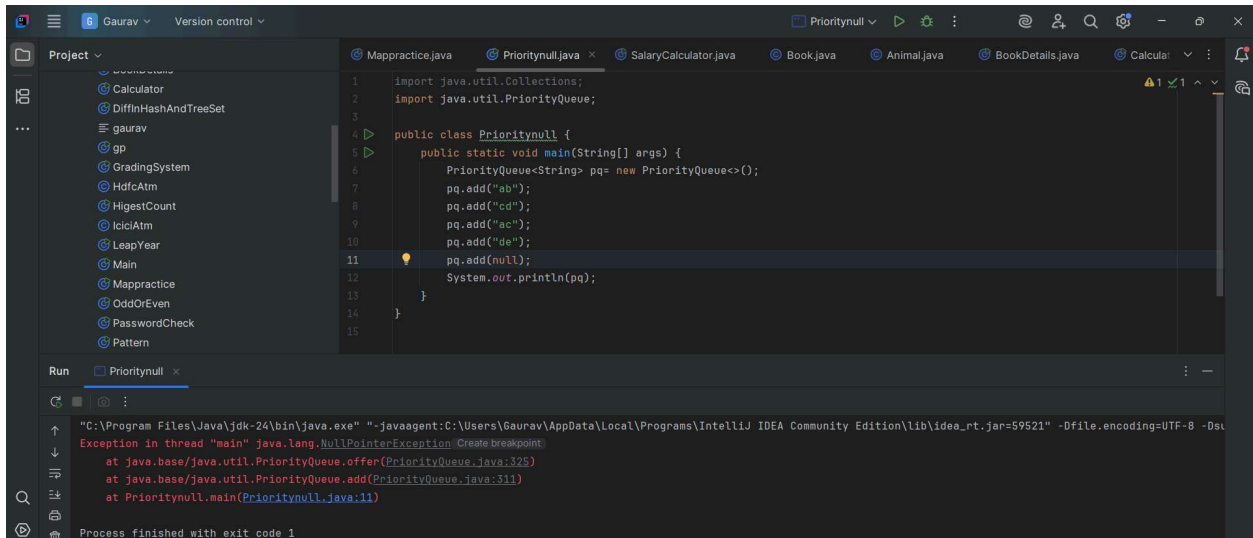
Run StringFrequencyMap

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea\_rt.jar=59237" -Dr

frequenc map {r=1, d=1, e=1, w=1, h=1, l=3, o=2}

Process finished with exit code 0

Q4.Can you store null elements in a PriorityQueue? Explain with a coding example.



```
1 import java.util.Collections;
2 import java.util.PriorityQueue;
3
4 public class Prioritynull {
5     public static void main(String[] args) {
6         PriorityQueue<String> pq= new PriorityQueue<>();
7         pq.add("ab");
8         pq.add("cd");
9         pq.add("ac");
10        pq.add("de");
11        pq.add(null);
12        System.out.println(pq);
13    }
14 }
15 }
```

Run Prioritynull x

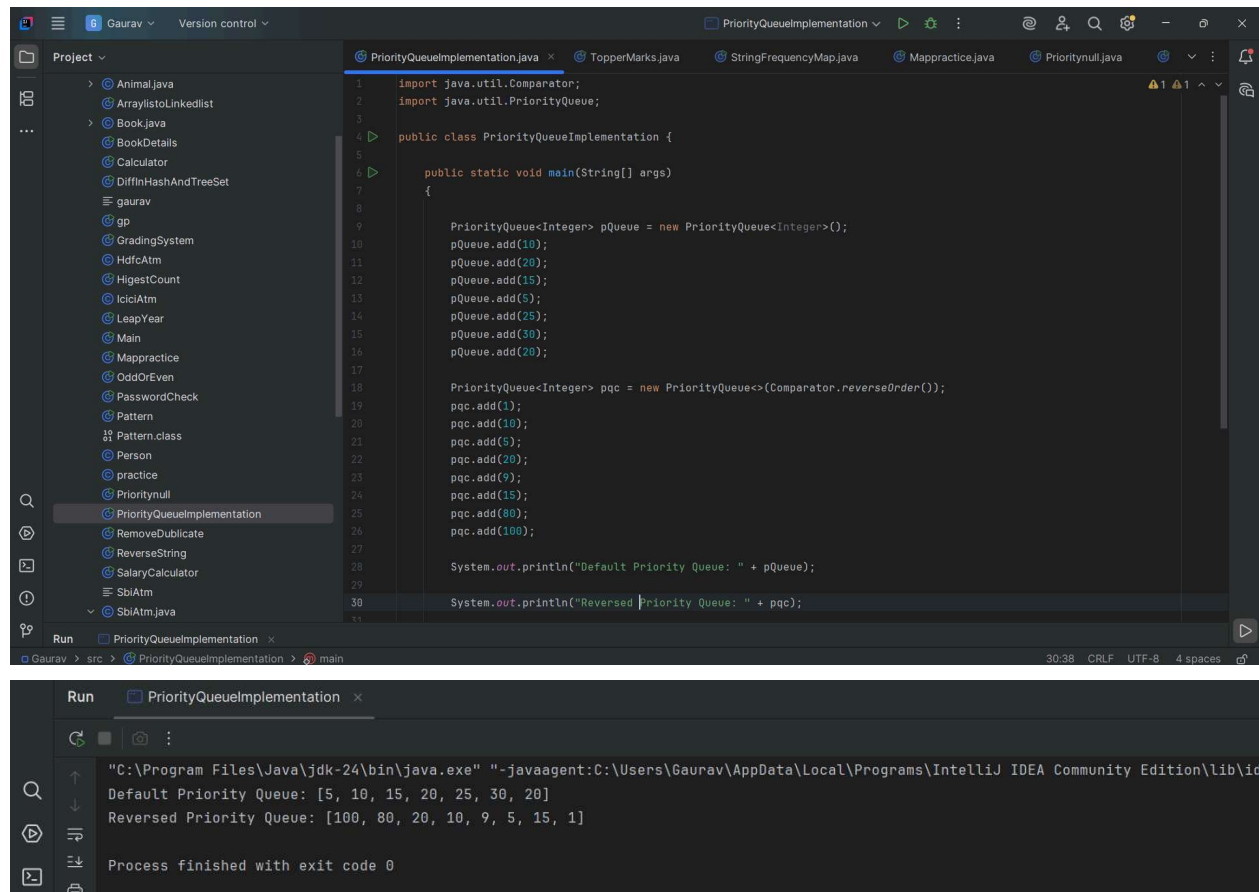
```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=59521" -Dfile.encoding=UTF-8 -Dsu
Exception in thread "main" java.lang.NullPointerException: Create breakpoint
    at java.base/java.util.PriorityQueue.offer(PriorityQueue.java:326)
    at java.base/java.util.PriorityQueue.add(PriorityQueue.java:311)
    at Prioritynull.main(Prioritynull.java:11)
Process finished with exit code 1
```

Priority Queue does not accept null, as priority queue makes a min heap which requires comparison but if we provide null which cannot be compared leads to problem in comparison due to which heap formation is not possible.

Q5.What is the default ordering in a PriorityQueue if no comparator is provided? Explain with the help of coding example

**Ans.**Priority queue uses the min heap by default which leads element to ordered in ascending order.

To reverse the order means to arrange in decreasing order we need to use comparator or collection.reverse().



```
1 import java.util.Comparator;
2 import java.util.PriorityQueue;
3
4 public class PriorityQueueImplementation {
5
6     public static void main(String[] args)
7     {
8
9         PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();
10        pQueue.add(10);
11        pQueue.add(20);
12        pQueue.add(15);
13        pQueue.add(5);
14        pQueue.add(25);
15        pQueue.add(30);
16        pQueue.add(20);
17
18        PriorityQueue<Integer> pqc = new PriorityQueue<>(Comparator.reverseOrder());
19        pqc.add(1);
20        pqc.add(10);
21        pqc.add(5);
22        pqc.add(20);
23        pqc.add(9);
24        pqc.add(15);
25        pqc.add(80);
26        pqc.add(100);
27
28        System.out.println("Default Priority Queue: " + pQueue);
29
30        System.out.println("Reversed Priority Queue: " + pqc);
31    }
32 }
```

Run

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\id
Default Priority Queue: [5, 10, 15, 20, 25, 30, 20]
Reversed Priority Queue: [100, 80, 20, 10, 9, 5, 15, 1]
Process finished with exit code 0
```



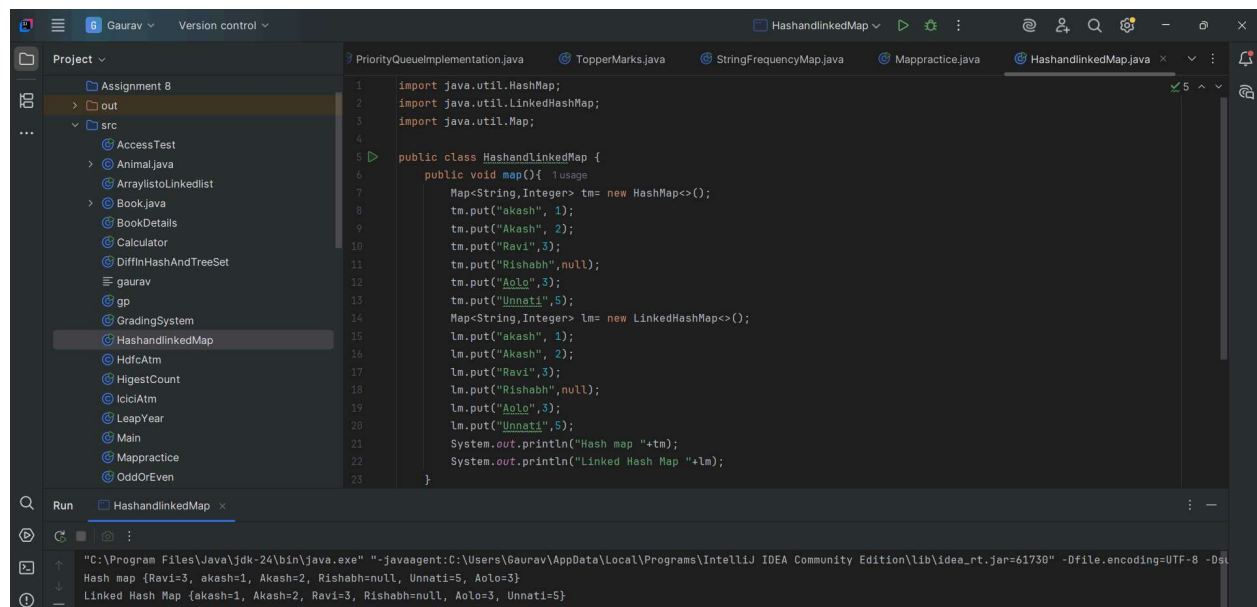
Q6.In which cases would you prefer LinkedHashMap over HashMap?

Explain with a coding example

**Ans.**The main difference between LinkedHashMap and HashMap is that Linked hash map maintain the insertion order whereas hash map do not maintain the insertion order.

We can use the linked hashmap when we need to maintain the insertion order.

For example we can use linked hash map to maintain the transaction history like type of transaction and amount it will only maintain last debit and [credit](#). We can also use it to implement LRU cash.



```
1 import java.util.HashMap;
2 import java.util.LinkedHashMap;
3 import java.util.Map;
4
5 public class HashandLinkedMap {
6     public void map(){
7         Map<String,Integer> tm= new HashMap<>();
8         tm.put("akash", 1);
9         tm.put("Akash", 2);
10        tm.put("Ravi",3);
11        tm.put("Rishabh",null);
12        tm.put("Aolo",3);
13        tm.put("Unnati",5);
14        Map<String,Integer> lm= new LinkedHashMap<>();
15        lm.put("akash", 1);
16        lm.put("Akash", 2);
17        lm.put("Ravi",3);
18        lm.put("Rishabh",null);
19        lm.put("Aolo",3);
20        lm.put("Unnati",5);
21        System.out.println("Hash map "+tm);
22        System.out.println("Linked Hash Map "+lm);
23    }
24 }
```

Run HashandLinkedMap

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea\_rt.jar=61730" -Dfile.encoding=UTF-8 -D...

Hash map {Ravi=3, akash=1, Akash=2, Rishabh=null, Unnati=5, Aolo=3}

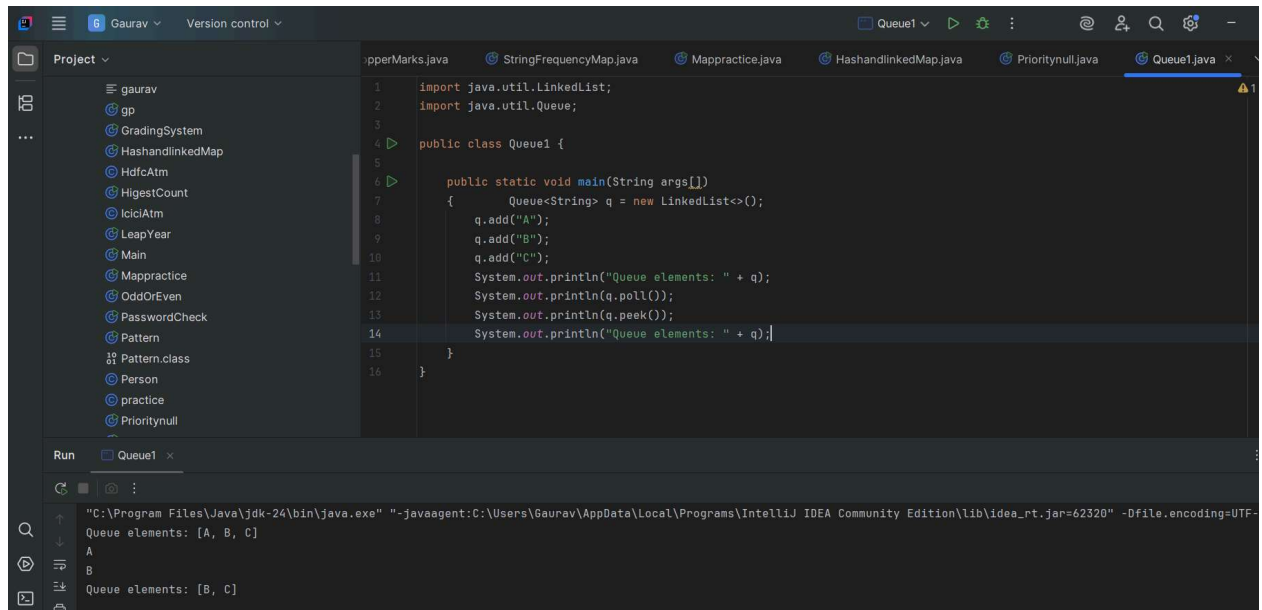
Linked Hash Map {akash=1, Akash=2, Ravi=3, Rishabh=null, Aolo=3, Unnati=5}

Q7.What is a Queue? How is it different from a Stack?

**Ans.**Queue is basically a liner data structure which is based on first in first out concept. The element which is entered first is taken out first.

The main difference between stack and queue is that stack is lifo(Last in First out) and queue is fifo(first in first out).

In Java queues implement Queue Interface of collections whereas stack implements list interface.



```
1 import java.util.LinkedList;
2 import java.util.Queue;
3
4 public class Queue1 {
5
6     public static void main(String args[])
7     {
8         Queue<String> q = new LinkedList<>();
9         q.add("A");
10        q.add("B");
11        q.add("C");
12        System.out.println("Queue elements: " + q);
13        System.out.println(q.poll());
14        System.out.println(q.peek());
15        System.out.println("Queue elements: " + q);
16    }
17 }
```

Run Queue1

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Users\Gaurav\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=62320" -Dfile.encoding=UTF-8
Queue elements: [A, B, C]
A
B
Queue elements: [B, C]
```