WIA1002/WIB1002 Data Structure

Tutorial: Priority Queue

1. Describe the main difference between Queue and PriorityQueue.

2. Briefly provide THREE (3) real-life example in using PriorityQueue.

3. Show the output for every System.out.println ((a) – (f)) in the following code:

```
 import java.util.*;
public static void main(String args[])
{
   PriorityQueue<String> pQueue = new PriorityQueue<String>();

   pQueue.offer("C++");
   pQueue.offer("Python");
   pQueue.offer("Java");
   pQueue.offer("Fortran");

   System.out.println("peek() gives us: "+ pQueue.peek());     //(a)

   System.out.println("The queue elements:");                  //(b)
   Iterator itr = pQueue.iterator();
   while (itr.hasNext())
      System.out.println(itr.next());                         //(b)

   pQueue.poll();
   System.out.println("After poll():");                        //(c)
   Iterator<String> itr2 = pQueue.iterator();
   while (itr2.hasNext())
      System.out.println(itr2.next());                        //(c)

   pQueue.remove("Java");
   System.out.println("After remove():");                      //(d)
   Iterator<String> itr3 = pQueue.iterator();
   while (itr3.hasNext())
      System.out.println(itr3.next());                        //(d)

   boolean b = pQueue.contains("Ruby");
   System.out.println ( "Priority queue contains Ruby or not?: " + b);          //(e)

   Object[] arr = pQueue.toArray();
   System.out.println ( "Value in array: ");                   //(f)
   for (int i = 0; i<arr.length; i++)
     System.out.println ( "Value: " + arr[i].toString()) ;     //(f)
```

```
}
```

4. Answer the following sub-questions with referring to the following code:

```
public class PriorityQueue2 {
   public static void main(String... args ){
      PriorityQueueComparator pqc=new PriorityQueueComparator();
      PriorityQueue<String> pq=new PriorityQueue<String>(5,pqc);
      pq.add("Jason");
      pq.add("Ali");
      pq.add("Muhamad");
      for(String s:pq){
            System.out.println(s);
      }
   }
}

public class PriorityQueueComparator implements Comparator<String>{
         public int compare(String s1, String s2) {
      if (s1.length() < s2.length()) {
         return -1;
      }
      if (s1.length() > s2.length()) {
         return 1;
      }
      return 0;
   }
}
```

a) What is the purpose of the PriorityQueueComparator in the code?
b) What is the output for the code?