QUESTION #1 *<35 points >*

```
import type.lang.*;

public class Transit
{  public static void main(String[] ar)
   {  Log log = Log.getRandom();
      IO.println("Enter route number ...");
      int route = IO.readInt();
      Route r = new Route(route);
      String lower = r.getLowerStation();
      String upper = r.getUpperStation();
      int expected = r.getExpectedTime();
      IO.print("Route Number " + route + ", " + lower);
      IO.println("<--> " + upper + " (" + expected + "min)");
      IO.println();
      IO.println("From\tVehicle\tDeparted\tArrived\tDeviation");
      for (Trip t = log.getFirst(route), t != null; t = log.getNext())
      {  if (t.getDirection == Trip.UP)
         {  IO.print(lower + " \t" + upper + "\t");
         } else
         {  IO.print(upper + " \t" + lower + "\t");
         }
         IO.print(t.getVehicleNumber());
         int saved = 0;
         if (t instanceof XTrip)
         {  (saved = (XTrip) t).getSavedTime();
            IO.print("-X");
         }
         Time start = t.getStartTime();
         Time end = t.getEndTime();
         IO.print("\t" + start + "\t" + end + "\t");
         IO.println(start.getInterval(end) - (expected - saved));
      }
   }
}
```

Marking Guideline

Give 35, 26, 17, 9, or 0. A few minor errors lead to 26, one major error leads to 17, and two or three major errors lead to 9. The key points are:

- Browsing the collection (major)
- Accessing the needed elements from the aggregates (major)
- Special handling of subclass (major)
- Casting (minor)
- Computing Time (minor)
- Incorporating saved time for xtrips (minor)
- Distinguishing up from down trips (minor)

QUESTION #2 *<35 points>*

```
import type.lang.*;
import java.util.*;
import java.text.*;

public class Bill
{  public static void main(String[] ar)
   {  UniReader ur = new UniReader("calls.txt");
      DateFormat df = DateFormat.getDateTimeInstance();
      int recNumber = 0;
      for (String rec = ur.readLine(); !ur.eof(); rec = ur.readLine())
      {  StringTokenizer stk = new StringTokenizer(rec, "|");
         int field = 0;
         try
         {  double rate = Double.parseDouble(stk.nextToken());
            field++;
            long from = df.parse(stk.nextToken()).getTime();
            field++;
            long upto = df.parse(stk.nextToken()).getTime();
            SE.require(upto >= from);
            IO.print(recNumber + "\tOK\t$");
            IO.println(rate * (upto - from) / 1000 / 100, ".2");
         } catch (SEpreconditionException e)
         {  IO.println(recNumber +
                       "\t\"start\" is later than \"end\"!");
         } catch (NoSuchElementException e)
         {  IO.println(recNumber + "\tMissing field!");
         } catch (NumberFormatException e)
         {  IO.println(recNumber + "\tNon-numeric rate!");
         } catch (ParseException e)
         {  if (field == 1)
            {  IO.println(recNumber + "\t\"start\" is not a date!");
            } else
            {  IO.println(recNumber + "\t\"end\" is not a date!");
            }
         }
         recNumber++;
      }
      ur.close();
   }
}
```

Marking Guideline

Give 35, 26, 17, 9, or 0. A few minor errors lead to 26, one major error leads to 17, and two or three major errors lead to 9. The key points are:

- Reading Loop (major)
- Handling exceptions (major)
- Correct Usage of other classes (major)

QUESTION #3 *<30 points >*

```
import type.lang.*;
import java.util.*;

public class LengthMode
{  public static void main(String[] ar)
   {  UniReader ur = new UniReader("strands.txt");
      Map dna = new TreeMap();
      for (String rec = ur.readLine(); !ur.eof(); rec = ur.readLine())
      {  String len = "" + rec.length();
         if (dna.containsKey(len))
         {  ((Set) dna.get(len)).add(rec);
         } else
         {  Set s = new TreeSet();
            s.add(rec);
            dna.put(len, s);
         }
      }
      ur.close();
      Iterator it = dna.keySet().iterator();
      //int max = 0;
      //Set best = null;
      for (; it.hasNext();)
      {  String len = (String) it.next();
         Set set = (Set) dna.get(len);
         IO.print(len + "\t");
         IO.println(set);
      }

   }
}
```

Marking Guideline

Give 30, 23, 15, 8, or 0. A few minor errors lead to 23, one major error leads to 15, and two or three major errors lead to 8.