Queens College, CUNY
Department of Computer Science

CS 212 – Object-Oriented Programming in Java – Fall 2014 – Exam 2

SOLUTIONS

Last Name _____First Name_____ Seat_____

Directions: There are four questions. **Read the whole question before answering**. Proper use of Java concepts is expected; minor syntax errors will be overlooked.

Question 1. 25 points.

Provide appropriate *catch* blocks that will catch each possible exception that can occur in the following *try* block: (do not just catch class Exception!) For each exception the catch block should print the exception's message.

```
try {
   int j = Integer.ParseInt(s);   // throws NumberFormatException
   char c = myString.charAt(i);  // throws IndexOutOfBoundsException
   StringTokenizer st = new StringTokenizer(myString); // throws
                                                       //NullPointerException
}
catch (NumberFormatException nfe) {
  System.out.println(nfe.getMessage());
}
catch (IndexOutOfBoundsException ioobe) {
  System.out.println(ioobe.getMessage());
}
catch (NullPointerException npe) {
  System.out.println(nfe.getMessage());
}
```

Question 2. 25 points.

Look at the code on page 5 (you may detach the page).

a. What methods, if any, are missing from class *Question2*? Why?

isValid and CompareTo because they are in the interface that is implemented.

b. Of the methods you listed in part (a) above, do they need to be written in class *Exam2*? Why?

No, but the class will be abstract if they are not.

c. Is it allowed to put the statement
```
d++;
```
on the line labeled **statement 1**? Why

Yes, although d is private it is an instance variable of the same class.

d. Is it allowed to put the statement
```
b++;
```
on the line labeled **statement 1**? Why

Yes, it is inherited and may be changed because it is protected.

e. Is it allowed to put the statement
```
a++;
```
on the line labeled **statement 1**? Why

No. It is not inherited because it is private.

f. What will be the output when class *Question2Main* is run?

*** 1 1 1
*** 6 7 8
*** 1 1 1
*** 6 7 8
*** 6 7 8 9
*** 2 3 4
*** 2 3 4 0

Question 3. 35 points.

```
public class Question3Node {
   Question3 data;
   Question3Node next;
   public Question3Node() {
      data=null;
      next=null;
   }
   public Question3Node(Question3 value) {
      data=value;
      next=null;
   }
}

public class Question3List {
   Question3Node first;
   Question3Node last;
   int length;

   public Question3List() {
      QuestioneNode n = new Question3Node();
      first=n;
      last = n;
      length =0;

   }


   }
   public void prepend (Question3 value) {
      Question3Node n = new Question3Node(value);
      n.next = first.next;
      first.next = n;
      length++;
      if (first==last) last = n;


  }

    Public int count(int i) {
          Question3Node p = first.next;
          int count=0;
          while (p != null) {
              if(p.data.length()==i)
                    count++;
              p=p.next;
          }
          return count;
    }
}
```

a. Fill in the missing code for the constructor of this singly-linked list which has a head node.

b. Fill in the missing code for the method *prepend* which should add a new node to the **beginning** of the list (as the new first data node). The new node should contain the data given as the parameter.

b. Fill in the missing code for the method *count* which should return a count of the number of nodes whose data is of length *i*. Assume the object Question3 has a length method.

Question 4. 15 points

What will be the output of the following:

```
public class MainPPPQQQ {

   public static void main(String[] args) {
      for (int i=1; i<3;i++){
         PPP ppp = new PPP(i);
         QQQ qqq = new QQQ(i);
      }
   }
}
```

```
public class PPP {
   public static int ppp,x=0;
   public PPP(int p){
      ppp=p;
      x+=1;
      System.out.println("p is "+p+", ppp is "+ppp+", x is "+x);
   }
}
```

```
public class QQQ extends PPP {
   public QQQ (int q){
      super(q++);
      System.out.println("q is "+q+", ppp is "+ppp+", x is "+x);
      q++;
      x+=ppp;
   }
}
```

p is 1, ppp is 1, x is 1
p is 1, ppp is 1, x is 2
q is 2, ppp is 1, x is 2
p is 2, ppp is 2, x is 4
p is 2, ppp is 2, x is 5
q is 3, ppp is 2, x is 5

```
public class Question2 implements Exam2Interface {
    private int a;
    protected int  b;
    public int c;

    public Question2() {
        a=1;b=1;c=1;
        System.out.println(toString());
    }
    public Question2(int p,int q,int r) {
        a=p; b=q; c=r;
        System.out.println(toString());
    }
    public String toString() {
        return ("*** "+a+" "+b+" "+c);
    }
}

public interface Exam2Interface {
   public Boolean isValid (Question2 q2);
   public int compareTo(Question2 q2);
}

public class Exam2 extends Question2  {
    private int d;
    public Exam2() {
        super (2,3,4);
        System.out.println(toString()+" "+d);
    }
    public Exam2(int m, int n, int o, int p) {
        super(m,n,o);
        d = p;
        System.out.println(toString()+" "+d);
        //  statement 1
    }
}

public class Question2Main {

    public static void main(String[] args) {
       Question2 q2;
       Exam2 e2;
       q2 = new Question2();
       q2 = new Question2(6,7,8);
       e2 = new Exam2();
       e2 = new Exam2(6,7,8,9);


    }

}
```