Mid-Term Test — Section L0102

Duration: *50 minutes* (10:10am–11:00am)
Aids Allowed: NONE (not even calculators)

Student Number: |_|_|_|_|_|_|_|_|_|

Last Name: ______________________________

First Name: ______________________________

Tutorial Section: (circle one)	Kenneth (MP 203)	Daniel (SS 1086)	Ruth (BF 323)	Sandra (IN 204)	Izabella (UC 52)

Do ***not*** *turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above, and read the instructions below *carefully*.)

This term test consists of 4 questions on 5 pages (including this one). *When you receive the signal to start, please make sure that your copy of the test is complete.* Answer each question directly on the test paper, in the space provided, and *use the reverse side of the pages for rough work.* (If you need more space for one of your solutions, use the reverse side of the page and indicate **clearly** which part of your work should be marked.)

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Also, unreadable answers will be given zero (0) so write legibly. In your answers, you may use without justification any facts given during the course, *as long as you state them clearly*. You must justify any other facts needed for your solution.

General Hint: We were careful to leave ample space on the test paper to answer each question. If you find yourself using much more room than what is available, you're probably missing something, so you should stop and take the time to think about what you're doing.

Bonus:	______/ 2
# 1:	______/10
# 2:	______/ 5
# 3:	______/ 8
# 4:	______/ 7
TOTAL:	______/30

Good Luck!　　　　PLEASE HAND IN

Bonus. [2 MARKS]

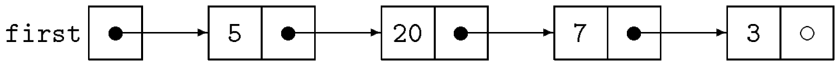
Write your name and student number **legibly** at the top of every page of this test, *except page 1.*

Question 1. [10 MARKS]

Consider the following `IntNode` class.

```
class IntNode {
   public int value;
   public IntNode link;
   // Details of constructor intentionally omitted...
}
```

Write a method `skipSum` that takes as input a reference to the first node in a linked list and returns the sum of every other node, starting with the first one. For example, the value returned should be 12 (the sum of 5 and 7) if the linked list looks like the picture below. (*Your method should work for **any** linked list, not just for the example given here.*)




```
// Returns the sum of every other value in a linked list, starting with
// the value stored in the node referenced by 'first'.
int skipSum( IntNode first ) {















}//end skipSum
```

Question 2. [5 MARKS]

Aliens have landed on Earth! They have been monitoring our communications through the Internet for quite a while, so they think that Java is a language used for communication, not programming. For this reason, their first communication to the leaders of Earth is actually a Java program, given below. This program has been received by the world's leaders, but unfortunately, the alien's technology has caused a world-wide power outage so there is no computer that can be used to run the program.

File "`Yak.java`":

```
import narf.Gmpn;
class Yak {
   public static void main(String[] a) {
      narf.Vlrk abra = new narf.Brzt();
      abra.kadabra( new Gmpn("Zous") );
   }
}//end Yak
```

File "`Gmpn.java`":

```
package narf;
public class Gmpn {
   public static String iuea;
   public Gmpn( String s ) {
      iiks( s );
   }
   public String iiks( String s ) {
      String shpr = iuea;
      if ( s.equals("Zous") )
         iuea = "Live Long and Prosper.";
      else if ( s.equals("Brgl") )
         iuea = "Resistance is Futile!";
      else
         iuea = "Dudes!";
      return shpr;
   }
}//end Gmpn
```

File "`Vlrk.java`":

```
package narf;
public class Vlrk {
   protected String gnnf;
   public Vlrk() {
      gnnf = "Brgl";
   }
   public String kadabra( Gmpn przt ) {
      System.out.println( Gmpn.iuea );
      return przt.iiks( gnnf );
   }
}//end Vlrk
```

File "`Brzt.java`":

```
package narf;
public class Brzt extends Vlrk {
   public Brzt() {
      super();
      gnnf = "Ytlw";
   }
   public String kadabra( Gmpn przt ) {
      gnnf = super.kadabra( przt );
      System.out.println( gnnf );
      return gnnf;
   }
}//end Brzt
```

Your task is to figure out the output of the alien's program. (This is not just for marks in CSC 148: the fate of our entire planet rests in *your* hands!) Below, write the output of the program when the `main` method from class `Yak` is run.

Question 3. [8 MARKS]

Professor Bouzille has composed an official reply to the alien's message, but it needs to be double-checked. In the code below, **circle** every line that will cause an error *at compile-time*, and for each such line, give a short explanation of the reason for the error (each explanation should be only a few words long). Write your explanation right next to the line causing the error. (**Be careful:** marks will be deducted for circling lines that compile without error!)

File "`First.java`":

```
import pack.Hello;

public class First {
   public void main( String[] args ) {
      Hello h = new Hello( "friends" );
      Goodbye g = new Goodbye( "aloha" );
      g.scramble( h );
      System.out.println( h.greeting );
   }
}//end First
```

File "`Hello.java`":

```
package pack;

public class Hello {
   static String greeting;
   private String message;

   public Hello( String message ) {
      Hello.greeting = "Welcome";
      message = message;
   }

   protected void scramble( Hello h ) {
      message = h.message;
   }
}//end Hello
```

File "`Goodbye.java`":

```
package pack;

public class Goodbye extends Hello {
   protected static String contents;

   public Goodbye( String message ) {
      super( message );
   }

   protected void scramble( Hello mess ) {
      super( mess );
      contents = super.message;
   }
}//end Goodbye
```

Question 4. [7 MARKS]

Once the errors have been fixed in the professor's program, and the reply has been sent to the aliens, a number of other communications are exchanged between the Earth's leaders and the aliens. In order to keep track of these communications, you are asked to define an ADT that supports the following operations:

- record the size of a message when it is received,
- return the *minimum* size of all messages received so far (throw `NoMessageException` if no message has been received),
- return the *maximum* size of all messages received so far (throw `NoMessageException` if no message has been received).

[2] **(a)** Write Java code to define the new exception `NoMessageException`.

[5] **(b)** Write a Java `interface` to represent the ADT described above. (To get full marks, you *must* include appropriate comments.)