Duration: **50 minutes**
Aids Allowed: **The Java API: An Introduction for Students**

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

Last Name: ______________________________

First Name: ______________________________

Lecture Section: __________ **Instructor:** ____________________

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 midterm test consists of 3 questions on 6 pages (including this one), printed on **both** sides of the paper. *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. If you need more space for one of your solutions, use the final page and *indicate clearly the part of your work that 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 think that you need much more room that what is available, take the time to stop and think about what you are doing instead: there is probably a shorter, easier solution. Also, remember that hints are just hints: you are not required to follow them if you can think of a different solution.

MARKING GUIDE

BONUS MARKS:	______/ 4
# 1:	______/ 8
# 2:	______/ 6
# 3:	______/22
TOTAL:	______/36

Good Luck!

Question 1. [8 MARKS]

A primitive data type can be considered to be similar to a class in some ways. For example, both have specifications to allocate memory for data storage, and both can perform operations on the data that they store.

Part (a) [6 MARKS]

(1 for each correct, -.5 for each incorrect, 0 for no answer)

Beside each of the following statements, write **I** or **O**:
I if it most closely describes memory allocation for an integer
O if it most closely describes memory allocation for an object

1. ____ procedure for memory allocation requires no specification from the programmer
2. ____ procedure for memory allocation requires some specification from the programmer
3. ____ execution of the declaration only allocates memory for reference
4. ____ execution of the declaration allocates memory for the variable
5. ____ size of storage for is fixed at 4 bytes
6. ____ size of storage depends on user specification

Part (b) [2 MARKS]

The data type `int` has built–in operators (such as +,-,*,/) from which expressions involving integer variables and values can be built. These operators take two integers and compute an integer result. Name 2 'operations' that the String class provides for 'computing results' involving its objects.

Question 2. [6 MARKS]

In the program below, part of the code necessary for a program to read from the keyboard and write to a file has been provided. In the blank space provided, write the appropriate statement(s) to:

1. Read input from the keyboard. Assume that the user will type something in response to the prompt "Type 1 for file, 2 for display", and then press return.
2. Write the string "This is a file" to the file if the user chose the first option
3. Write the string "This is the display" to the display if the user chose the second option
4. Write the string "You didn't follow instructions!" to the display if the user responds in any other manner.

```
import java.io.*;
class Test {

    private static BufferedReader input;
    private static PrintStream output;

    public static void main(String[] args) throws IOException {
           InputStreamReader k = new InputStreamReader(System.in);
           input = new BufferedReader(k);
           output = new PrintStream(new FileOutputStream("testout.txt"));
           System.out.println("Type 1 for file, 2 for display");




           output.flush();
           output.close();
        }
}
```

Question 3. [22 MARKS]

The program below compiles and runs without error. Parts a) through d) and the bonus question on the following pages refer to this program. (The line numbers in the left margin are *not* part of the program; they are present so the questions can refer to the program specifically)

```
1  public class Website {
2
3     public int numVisitors;
4     private String name;
5     private static String url;
6     private static String versionInfo = "Version 1.0";
7
8     public Website(String n, String u) {
9
10          name = n;
11          url  = u;
12          numVisitors = 0;
13    }
14
15    public static double getVersion() {
16
17         return Double.parseDouble(getVersionNumber());
18    }
19
20    private static String getVersionNumber(){
21         return versionInfo.substring(8,11);
22    }
23
24   public String getName() {
25
26         return name;
27    }
28
29    public String toString() {
30
31         String stringRepresentation;
32         stringRepresentation = "Website " + url + " entitled " + name;
33         return  stringRepresentation;
34    }
35 }
36
37 class Test {
38    public static void main(String args[]) {
39
40         String site = "http://www.cs.utoronto.ca/~csc108h";
41         Website w = new Website("Java 108", site);
42
43         System.out.println("You are looking at " + w);
44         System.out.println("Version " + (Website.getVersion() + 1));
45    }
46 }
```

Part (a) [6 MARKS]

(i) [2 MARKS]

Write down the name of each local variable in the program.

(ii) [2 MARKS]

Write down the name of each instance variable in the program.

(iii) [2 MARKS]

Write down the name of each declared variable that could be used in the method **getVersion()**

Part (b) [10 MARKS]

[One mark for each correct answer, -.5 marks for each incorrect answer, 0 if not answered]
Circle **error** or **okay** for each question below if the statement would cause an error when inserted on the given line. (The error could occur while compiling or running the program)

1. error okay **String s = site;** on line 9
2. error okay **double j = getVersion();** on line 25
3. error okay **String v = Website.getVersionNumber();** on line 42
4. error okay **String s = getName();** on line 16
5. error okay **System.out.println(Website.getName());** on line 42
6. error okay **System.out.println(w.stringRepresentation);** on line 42
7. error okay **name = stringRepresentation;** on line 25
8. error okay **w.numVisitors++;** on line 42
9. error okay **String baseUrl = Website.url.substring(0,25);** on line 42
10. error okay **System.out.println(w.getName());** on line 42

Part (c) [4 MARKS]

Print the output of the program.

Part (d) [2 MARKS]

Consider the program if we replaced (line 44)
System.out.println("Version " + (Website.getVersion() + 1));
with
System.out.println("Version " + Website.getVersion() + 1);
Print the output of this program.

Bonus. [4 MARKS]

Suppose that the variable **site** might store *any* URL that starts with "http://" and has at least one more '/' character in it.

What statement(s) would you insert between lines 41 and 43 to extract the portion of the string **site** which occurs after "http://" and before the next "/", and stores it in a String variable called **machineName** ?

(Note: this should work for *any* URL, not just the one in the code above. If the value of the variable **site** from the code above were used, your statements would give the value "www.cs.utoronto.ca" to the variable **machineName**)

You are welcome to use String methods `indexOf()` and `substring()` which can be found in your API.

Total Marks = 36