PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

AUGUST EXAMINATIONS 2000

CSC 108H1 Y
St. George/Erindale Campuses

Duration — 3 hours

Examination Aids: Textbook (Arnow & Weiss, Introduction to Programming using Java) and API reference (Appendix O from Lewis & Loftus, Java Software Solutions, or the entire book by Lewis & Loftus).

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

Last Name: ____________________

First Name: ____________________

Lecture Section: (circle one)	St. George L0101 (Ortigas)	St. George L5101 (Gries)	Erindale (Baumgartner)

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 final examination consists of 8 questions on 12 pages (including this one). *When you receive the signal to start, please make sure that your copy of the examination is complete.* Answer each question directly on the examination 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.

Unless otherwise stated, comments are not required.

# 1:	______/10
# 2:	______/ 8
# 3:	______/10
# 4:	______/ 6
# 5:	______/10
# 6:	______/10
# 7:	______/10
# 8:	______/10
TOTAL:	______/74

Good Luck!

PLEASE HAND IN

Question 1. [10 MARKS]

A to-do list is a collection of tasks. The tasks are prioritized on a five-point scale, with 0 being the highest priority and 4 being the lowest. Furthermore, the tasks are sorted by priority.

Conceptually, a to-do list might look like this:

```
Priority    Tasks
--------    -----
0           Pay bills
2           Do groceries
2           Do laundry
4           Fix VCR
```

The above example illustrates two important points:

1. More than one task may share the same priority (`Do groceries` and `Do laundry` both have priority 2), and
2. not every priority level has tasks.

The following operations can be performed on a to-do list.

- **addTask(t, p):** Adds task `t` to the to-do list with priority `p`.
- **getPriority(t):** Returns the priority of task `t`, or -1 if `t` doesn't exist. (Hint: use `Vector` method `contains(obj)`, which returns `true` if `obj` is in the `Vector`.)
- **getHighestPriorityTask()** Returns the task with the highest priority, or `null` if no tasks exist. If more than one task share the highest priority, then just return one task.
- **listTasks():** Print out this to-do list, following the format in the example above.

Implement the Java class `ToDoList` on the following page. That is, write a method for each operation. Use `String`s to represent tasks and `int`s to represent priorities; note that this means that all return values and parameters must be `int`s or `String`s. You must use an array of `Vector`s as your data structure to store the priorities and the tasks at that priority. You may assume that there are exactly five priority levels. Be sure to initialize your instance varibles, either as default values or in a constructor.

The description of this question is on the previous page.

```
public class ToDoList {









}
```

Question 2. [8 MARKS]

Part (a) [2 MARKS]

Write your student number **legibly** at the top of every page of this exam. (On page 1, write it where we ask for it.)

Part (b) [3 MARKS]

Are there any situations in which the loop inside method `f()` can terminate?

```
public static void f(int[] a) {
    int i = 0;
    while (i != a.length) {
        System.out.println(a[i]);
    }
}
```

☐ Terminates ☐ Can never terminate

If the loop can terminate, describe how it might happen. If it does not terminate, explain why.

Part (c) [3 MARKS]

Are there any situations in which the loop inside method `g()` can terminate?

```
public static void g(int[] a) {
    int i = -1;
    while (i != a.length/2) {
        i = i + 2;
        System.out.println(a[i]);
    }
}
```

☐ Terminates ☐ Can never terminate

If the loop can terminate, describe how it might happen. If it does not terminate, explain why.

Question 3. [10 MARKS]

A student, protesting the formatting requirements for submitted Java programs, wants a method which reads lines from a `BufferedReader` and prints them out with statements which are split over more than one line rejoined.
Specifically, every time it encounters a line whose last character isn't a ';' it adds on a space and the next line.

Example:

```
public static void g(int[] a) {
    int i = 0;
    while (i != a.length/2) {
        System.out.println(a[i]);
        i = i + 2;
    }
}
```

would be printed as:

```
public static void g(int[] a) {     int i = 0;
    while (i != a.length/2) {        System.out.println(a[i]);
        i = i + 2;
    } }
```

Complete the following method:

```
// Read lines from br, terminating when br.readLine() returns null. Split
// lines according to the scheme described above.
public static void printMangled(BufferedReader br) {

}
```

Question 4. [6 MARKS]

Consider the following program, which runs without error:

```
import java.io.*;
public class Test {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        fa(Integer.parseInt(in.readLine()));
    }

    public static int fa(int i) {
        int x = 0;
        for (int k = 1; k <= i; k += 2) {
            x += mi(re(i*2)); // A
        }
        System.out.println(x);
        return x;
    }

    public static int re(int j) {
        return mi(j*4); // B
    }

    public static int mi(int k) {
        return k/2; // C
    }
}
```

Part (a) [2 MARKS]

If the input is the string "4", what output does the program produce?
Answer:

Part (b) [2 MARKS]

If a represents the number of times that the line labeled `A` is executed, and c represents the number of times that the line labeled `C` is executed, precisely describe the relationship between a and c.
Answer:

Part (c) [2 MARKS]

If the input contains a string representing the number N, how many times is the line labeled `B` executed? Circle the function that most closely resembles the behaviour of the program with respect to the line labeled `B`, *ignoring constant factors.*

`1      log N      sqrt(N)      N      N^2      N^3      N log N      N^2 log N`

Question 5. [10 MARKS]

```
public abstract class Pet {
    private String name;
    protected boolean alive = true;
    public Pet(String name) {
        this.name = name;
    }
    public abstract void feed();
    public String getStatus () {
        String result = name + " is ";
        if (alive) {
            result += "alive.";
        } else {
            result += "deceased.";
        }
        return result;
    }
}
```

```
public class Cat extends Pet {
    public Cat(String name) { super(name); }

    public void feed() {
        System.out.print("purr ");
    }
    public void rub() {
        System.out.print("purr ");
    }
}
```

```
public class Menagerie {
    public static void main(String[] args) {
        Cat c = new Cat("Spot");  c.rub();  c.feed();
        System.out.println(c.getStatus());
        p = c;
        Pet p = new Cat("Scratchy");
        c = p;
        p.rub();
        p.feed();
        System.out.println(p.getStatus());
        Goldfish g = new Goldfish("Bubbles");
        p = g;
        g.feed();
        p.feed();  p.feed();
        System.out.print(p.getStatus() + " ");
        System.out.println(g.getStatus());
    }
}
```

```
public class Goldfish extends Pet {
    private int timesFed = 0;
    public Goldfish(String name) { super(name); }
    // Don't overfeed!
    public void feed() {
        timesFed++;
        if (timesFed > 2) {
            alive = false;
        }
    }
}
```

Part (a) [3 MARKS]

Cross out all the statements in `Menagerie` which cause a compile time error.

Part (b) [3 MARKS]

If the lines that you crossed out did not exist, what is the output?

Part (c) [4 MARKS]

Define a static method `rubAll` which rubs all the `Cats` in a given `Vector` of `Cats`. (There are no `Fish` in the `Vector`, only `Cats`.)

Question 6. [10 MARKS]

Consider the following code:

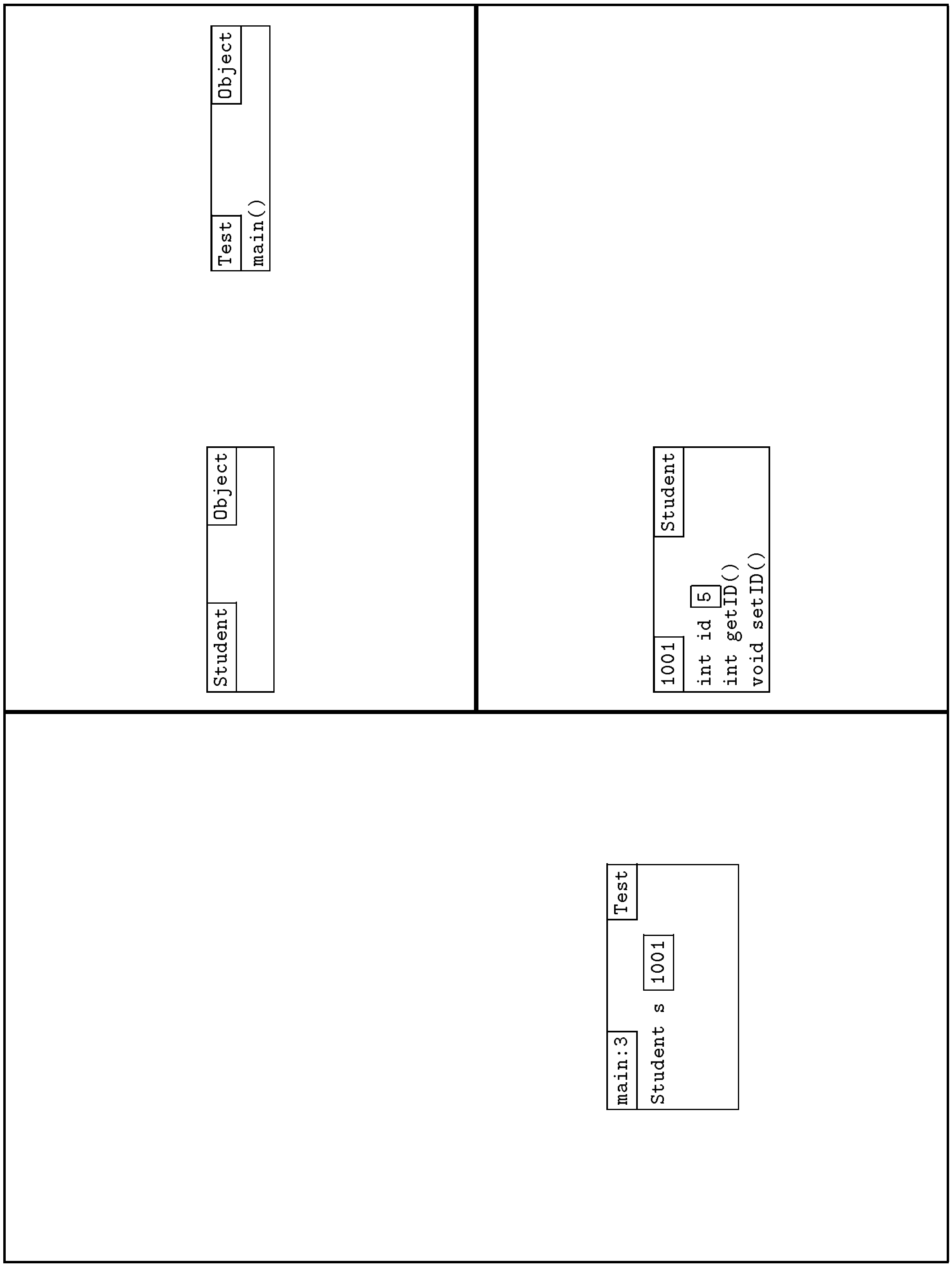
```
class Student {
    private int id;
    public void setID(int i) { id = i; }
    public int getID() { return id; }
}

public class Test {
    public static void main(String[] args) {
        Student s = new Student();   // Line 1
        s.setID(5);                  // Line 2
        Student t = s;               // Line 3
        m(s);                        // Line 4
        System.out.println(s == t);  // Line 5
    }

    public static int m(Student s) {
        Student k; // Line 1
        k = s;     // Line 2; stop tracing before
                   //   you execute this line.

        // YOUR CODE FOR PART B, IF ANY, GOES HERE.





    }
}
```

Part (a) [6 MARKS]

The memory model on the following page shows the state just before line 3 of method `main` is executed. (Line 3 is the line that declares variable `t`.) Continue the trace until just before line 2 of method `m` has been executed.

Part (b) [4 MARKS]

If there is something you can write inside method `m` that will cause the last line of method `main` to print `false`, write it. If there isn't, explain why not.


Student
Object
Test
Object
main()
1001
Student
int id 5
int getID()
void setID()
main:3
Test
Student s 1001

Question 7. [10 MARKS]

Part (a) [6 MARKS]

On the following page, rewrite the two classes shown at the bottom of this page so that they are children of a common superclass. (Write that superclass, too.) Only move common information to the superclass. This includes all method headers, but only some method bodies and some variables.
Leave your variable visibility as protected so that subclasses can see them; don't make them private. Be sure to make methods abstract when appropriate.

Part (b) [4 MARKS]

Write a class `BlueJeans` (blue jeans are a type of pants). They must *always* be blue.

```
import java.util.*;
// The root class of clothes worn on the top half of the body.
public abstract class Top {
    // My colour.
    protected String colour;
    // My size; usually one of small, medium and large.
    protected String size;

    // Set my colour to c.
    public void setColour(String c) { colour = c; }
    // Return my colour.
    public String getColour() { return colour; }

    // Set my size to s
    public void setSize(String s) { size = s; }
    // Return my size.
    public String getSize() { return size; }
}

// The root class of clothes worn on the bottom half of the body.
public abstract class Bottom {
    // My colour.
    protected String colour;
    // My size; usually in the range 3-18.
    protected int size;

    // Set my colour to c.
    public void setColour(String c) { colour = c; }
    // Return my colour.
    public String getColour() { return colour; }

    // setSize: Set my size to s, which is a String representation of
    // my integer size.
    public void setSize(String s) { size = Integer.parseInt(s); }
    // Return my size.
    public String getSize() { return "" + size; }
}
```

Code for the question on the previous page should be written here.

Question 8. [10 MARKS]

For data compression and other purposes, we often need to look for runs of repeated characters in data. For example, in the string

```
abbbbccdefffffg
```

there is a run of 4 `b`'s, a run of 2 `c`'s, and a run of 5 `f`'s. To be exact, we also consider that the `a`, the `d`, the `e` and the `g` are all runs of length 1.

In this question, you must complete the method `encodeString`, which is begun below. Here are some examples of values returned by `encodeString`:

- `encodeString("abbbbccdefffffg")` returns `"1a4b2c1d1e5f1g"`
- `encodeString("a")` returns `"1a"`
- `encodeString("xxbbbxx")` returns `"2x3b2x"` (The two runs of x's are separate runs.)
- `encodeString("aaAa")` returns `"2a1A1a"` (Upper-case and lower-case are different.)
- `encodeString("")` returns the empty string.

Here is the beginning of the method you are to complete:

```
// encodeString: returns the contents of data, encoded.
public static String encodeString (String data) {
    // You write the rest.











}
```

Total Marks = 74