Name: _____

Student No: _____

Department of Computer Science
University of Western Ontario

CS 025a Computer Science Fundamentals I

October 16, 2007

Midterm Test

TIME: 1 hour and 45 minutes

One single-sided hand-written sheet of notes allowed.

This test has four parts, with a total value of 75 points.

Part I: True or False (10 points)

Decide whether the following statements are true or false.

Write **T** in the blank following each of the true statements and **F** in the blank following each of the false statements. Make a clear distinction between your **T**s and **F**s — ambiguous letters will be marked as incorrect.

1. Classes in Java may be used to give user-defined types. ____
2. Quicksort can be expressed most naturally using recursion. ____
3. When growing dynamically sized arrays, it is always more efficient to grow them one slot at a time. ____
4. It is best for all software components to have sophisticated, complex interfaces to avoid any extra computation and to anticipate every possible situation. ____
5. The only way to calculate factorials is with a for loop. ____
6. Lisp syntax is harder to analyze than C syntax. ____
7. With garbage collection, the programmer must explicitly deallocate all objects. ____
8. Java supports garbage collection. ____
9. In Java, null can only be assigned to variables of type String. ____
10. Generally a program with $100n$ lines of code takes about 100 times longer to develop than a program of n lines of code. ____

Name: _____

Student No: _____

Part II: Short Answer (10 Points)

For the multiple choice questions, circle exactly one of A, B, C or D corresponding to the best answer.

For the other questions, fill in the blank with a one or two word answer.

1. Java's syntax most closely resembles that of:

   (A) C (B) English (C) Lisp (D) Visual Basic

2. Which of the following is not a primitive type in Java:

   (A) boolean (B) complex (C) double (D) int

3. Which of the following is not a Java control flow statement:

   (A) for (B) if (C) try (D) jump

4. In Java, types are associated with variables by their:

   (A) declaration (B) scope (C) spelling (D) use

5. In Java, error situations are best handled using?

   (A) arrays (B) exceptions (C) loops (D) recursion

6. In Java, a method can specify that it cannot be over-ridden by using the _____ keyword in its declaration.

7. In Java, a field that should be visible within a class and its subclasses, but not otherwise, should be declared _____.

8. In Java, output can be displayed on the console using _____.

9. In Java, exceptions are handled using a _____ statement.

10. In Java, new objects are allocated using _____.

Name: _____

Student No: _____

Part III: Tic Tac Toe (35 points)

For the following questions, we will address aspects of the game Tic-Tac-Toe.

The game of Tic-Tac-Toe is played on a 3×3 grid, with players "X" and "O" taking turns marking their symbols in empty squares. The winner is the first player to place 3 of their own symbols in a line, either as one of the rows, or either of the diagonals. The game is very simple and normally tied by all but the very weakest players.

If you cannot answer one of the questions, continue and try to solve the following questions.

Question III.1 (1 point) Writing programs when you have only a high-level idea of the blocks is an example of _____-down design (fill in the blank).

Question III.2 (10 points) Write a class `TTT_Position` that represents the board with a private array of array of integers.

It should export

- three intgeger constants (whose value is up to you) named `EmptySquare`, `PlayerX` and `PlayerO`,
- a public constructor `TTT_Position()` that makes a position in which all the squares are empty,
- a private constructor `TTT_Position(TTT_Position another)` that takes another position as its argument and makes a new one with the same entries,
- public methods
  `int getValue(int row, int col)` and
  `void setValue(int row, int col, int value)`
  to get the contents and set the contents of a square.
- `size()` that returns the width of the square (in this case 3), declared so that it can be called as `TTT_Position.size()`.

Question III.3 (4 points) Write a method `boolean hasEmptySquare()` that could be added to the class `TTT_Position` to tell whether a position has any empty squares.

Name: _____

Student No: _____

Question III.4 (10 points) Write a set of five related classes

```
TTT_Line
TTT_Row                 // To represent each of the 3 rows
TTT_Column              // To represent each of the 3 columns
TTT_Diagonal            // To represent the diagonal (0,0), (1,1), (2,2)
TTT_Antidiagonal        // To represent the diagonal (2,0), (1,1), (0,2)
```

Use subclassing so that all of the classes `TTT_Row`, `TTT_Column`, `TTT_Diagonal` and `TTT_Antidiagonal` are subclasses of `TTT_Line`.

The class `TTT_Line` should provide the method `int getValue(int i)` that gives the contents of its i-th entry, for $i = 0, 1, 2$.

Question III.5 (5 points) Write a method `int countLines(int player)` that could be added to the class `TTT_Position` to tell how many lines (rows, columns, diagonals) the given player has.

You will get full marks for using the classes `TTT_Line`, `TTT_Row`, `TTT_Column`, `TTT_Diagonal` and `TTT_Antidiagonal`. If you work on the array directly, you will get only part marks.

Question III.6 (5 points) Write a method `int winner()` that could be added to `TTT_Position` to check whether the position is won by a player. It should use the methods `countLines` defined above. It should return `PlayerX` or `PlayerO` if the game has been won by "X" or "O" respectively. It should return `EmptySquare` when (i) there are no empty squares and the game is tied or (ii) there are empty squares left and the game is tied. The tied situation includes the cases when neither player has a line of symbols or (for completeness) both players have a line of symbols. a method `size()` that returns 3.

Name: _____

Student No: _____

Part IV: Iteration and Recursion (20 points)

Question IV.1 (5 points) Write an iterative Java program to calculate the factorial $n! = n \times (n-1) \times \cdots \times 2 \times 1$.

Question IV.2 (5 points) Write a recursive Java program to calculate the factorial $n! = n \times (n-1) \times \cdots \times 2 \times 1$.

Question IV.3 (5 points) Write an iterative Java program to compute x to the power n for x a double precision floating point number and n a positive integer.

For large n, your program takes time roughly proportional to some function of n. Which of the choices below best describes this function for the running time?
(a) n (b) n^2 (c) $\log(n)$ (d) 2^n (e) none of the above.

Question IV.4 (5 points) Write a recursive Java program to compute x to the power n for x a double precision floating point number and n a positive integer. You should use the method of "repeated squaring."

For large n, your program takes time roughly proportional to some function of n. Which of the choices below best describes this function for the running time?
(a) n (b) n^2 (c) $\log(n)$ (d) 2^n (e) none of the above.