Name: ____________________

Student No: ______________

Department of Computer Science
University of Western Ontario

CS 1025a Computer Science Fundamentals I

October 28, 2008

Midterm Test

TIME: 1 hour and 45 minutes

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

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. Comments are for wimps. If a program was hard to write, it should be hard to read. ____
2. When writing programs, efficiency is always more important than clarity. ____
3. It is best for all software components to have sophisticated, complex interfaces to avoid any extra computation and to anticipate every possible situation. ____
4. A program with 100,000 lines of code takes about 100 times longer to develop than a program of 1000 lines of code. ____
5. In Java, all variables are initialized to `null`. ____
6. In Java, abstract classes must always have constructors. ____
7. Without garbage collection, the programmer should explicitly deallocate unused objects. ____
8. Java is the preferred language for all programming tasks. ____
9. A program with asymptotic complexity $O(n)$ is always faster than a program with asymptotic complexity $O(n^2)$. ____
10. When growing dynamically sized arrays, it is always more efficient to grow them one slot at a time. ____

Name: ____________________

Student No: ______________

Part II: Java Knowledge – 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 word answer.

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

   (A) Ada (B) C++ (C) Lisp (D) Visual Basic

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

   (A) `boolean` (B) `char` (C) `double` (D) `list`

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

   (A) `for` (B) `if` (C) `select` (D) `return`

4. In Java, subclasses are created using the keyword:

   (A) `extends` (B) `inherits` (C) `instanceof` (D) `uses`

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 does not depend on a specific object by using the ______________ keyword in its declaration.

7. In Java, a field that should be visible within a class, but not in its subclasses and not outside the class, should be declared ______________.

8. In Java, a useful library for building GUIs is the ______________ library.

9. In Java, exceptions are raised using a ______________ statement.

10. In Java, to access the constructor of a superclass, the ______________ keyword should be used.

Name: ____________________

Student No: ______________

Part III: Imperative Java Programming (25 points)

Question III.1 (5 points) Write a static Java method to calculate the sum of two integers.

Question III.2 (5 points) Write a static Java method that takes an integer parameter n and returns an array of integers containing the values i^2 for i going from 0 to $n - 1$.

Question III.3 (5 points) Write a static Java method that takes an array of arrays of integers (a value of type `int [][]` and finds the maximum integer contained in it. You can assume that none of the arrays are empty, i.e. null or of size zero.

Question III.4 (3 points) Redo Question III.4, but this time allow for the possibility that some of the arrays might be empty. Your method should have a second argument that is the value to be returned in case all the inner arrays are empty.

Question III.5 (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."

Question III.6 (2 points) For large n, computing x^n using "repeated squaring" 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.

Name: ____________________

Student No: ________________

Part IV: Class Hierarchies Example (30 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 a row, column or one of the two diagonals. The game is very simple and normally tied by adult players.

If you cannot answer one of the questions, continue and try to solve the remaining questions. In each question you may assume the existence of all the classes and methods described in the previous questions.

Question IV.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 IV.2 (5 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 IV.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 IV.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 an **abstract** class `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 IV.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 IV.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.