CS 11 Fall 2004 — Mid-term 1

You have **50 minutes** to complete this exam. There are a total of 100 points on this test, so be sure to budget your time on questions accordingly. **Write clearly, and answer all questions completely but concisely.**

1. (20 points) Consider the code fragment below. **Mark each location** where an *automatic cast* will occur. Also find each location where an *explicit cast* must be inserted for the code to compile successfully, and **correct the line** with that explicit cast included. [Note: Be aware that each line may require more than one cast!]

```
byte b = 13;
int i = b;
char c = i;
int i2 = c + i;
char c2 = c + 3;
boolean b = (i < c);
```

2. (15 points) The following method contains an error that will prevent it from compiling. **Find and correct it**.

```
public static int getPositiveInt () {

    do {

        System.out.print("Enter a positive integer: ");
        int x = Keyboard.readInt();
        if (x <= 0) {
            System.out.println("No good, try again: ");
        }

    } while (x <= 0);

    return x;

}
```

3. (30 points) **Write a method** named `printTable` that accepts a *height* and a *width* as parameters and then prints a table of those dimensions, where the table follows the following pattern:

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
```

The above table is the result of calling `printTable(4, 5)`.

4. (25 points) Consider the following program:

```
class Factorial {

    public static void main (String[] args) {

        int x = calcFactorial(4);
        System.out.println("Final answer = " + x);

    } // main

    public static int calcFactorial (int n) {

        System.out.println("Requested " + n + '!');
        int result;
        if (n == 0) {
            result = 1;
        } else {
            result = n * calcFactorial(n-1);
        }

        System.out.println("Returning " + n + "! = " + result);
        return result;

    } // calcFactorial

} // class Factorial
```

Provide the output that this program would produce.

5. (10 points) Consider the *Fibonacci sequence:*

   `0, 1, 1, 2, 3, 5, 8, 13`

   Specifically, we define it as:

   - $F(0) = 0$
   - $F(1) = 1$
   - $F(x) = F(x-1) + F(x-2)$ if $x \geq 2$

   Write a recursive method that calculates a requested Fibonacci number. That is, complete the method below (where the declaration is provided), and use no loop structures.

   ```
   public static int calcFibonacci (int x) {
   ```