CSC 211: Introductory Programming and Design

Midterm Exam 2 – November 17, 2003

Duration of the examination: 1 h 15 min.
Documentation: Textbook and notes allowed.
Comments: Use your time wisely (no problem deserves that you spend 1/2h on it).

Problem 1 15 Pts

Identify as many syntax error as you can in the following code.
When deciding if a statement is correct, you should assume that the corrections you made on the previous lines have been applied (i.e. that the preceding lines do not contain errors anymore).

Block 1

String[] S,
int[] myArray,
    B[5];
Integer[] A,
    [][] C;
double[] z[],
    [] D[10];

Block 2

float[] A = new Array[float];
int[] B = {1, 2, 3, 4, 5};
int[] C[] = {1, 2, 3, 4, 5};
int[] D[3][2] = {{1, 2}, {3, 4}, {5, 6}};
int[][] E = {{1, 2, 3}, {4}, {5, 6}};
int[][] F[][] = {{1, 2, 3}, 4, 5, {6, 7}};
float[] G[] = {1., 2., 3., 4.};
double[] H[][] = {{1., 4.}, {}, {7., 8., 9.}};
Block 3

```java
int A[][] = {1, 2, 3, 4, 5};
int[][] B = {{1, 2, 4, 5}, {6, 7}, {3}, {9, 8, 7}};
String D[] = {"one", "two", "three"};
int[] E = {3, 5, 7};
int[][] F, G, H;
float[] x = {2., 3.5, 7., 0., 5.}, y;

F = new int[5];
G = new int[B.length][];
for (short i = 0; i < B.length; i++)
    G[i] = B[i];

D = new String[3];
D[1] = "replacement string";

E = {10, 11, 12};

H = B;
H[1] = E;
H[4][1] = 0;

for (short i = 0; i < y.length; i++)
    y[i] = i * i;

B[1] = x;
y = B[2];
for (short i = 0; (i < E.length) && (i < x.length); i++)
    x[i] = E[i];
```
Problem 2

Write the code of a method named `bestString` that receives as parameters a one-dimensional array of `String` and a target `char`, and returns the element of the array that contains the most occurrences of the target character. In case of a tie, the element with the lowest index should be returned. If no element in the array contains the target character, return `null`. 
Problem 3  

Write the code of a method named `findPeaks` that receives as parameters a two-dimensional array of `int` representing an elevation map (altitude at points on a grid), and returns another two-dimensional array of `int` storing the row and column numbers in the first array of local peaks, that is, points that are higher than all their immediate neighbors.  

[10 pts extra credit for handling properly points that are on the border of the grid]
Consider the following (admittedly ugly and messy) class implementation:

```java
public class SillyClass {
    private int a, b;
    static int m = 0;  // counts objects created

    public SillyClass(int theA, int theB) {
        a = theA;
        b = theB;
        m++;  // increments counter
    }

    public SillyClass firstMethod(int a, SillyClass theObj) {
        int m;
        this.a += a++ + b++;
        this.m += 1;
        m = a + theObj.a;
        theObj.b -= 2;
        return new SillyClass(m, b);
    }

    public int secondMethod(SillyClass theObj) {
        SillyClass temp = theObj.firstMethod(a, this);
        return m + a + temp.a;
    }
}
```
Trace the value of $a$, $b$, $m$, $n$, and $p$ during the execution of the following block of code. Feel free to use the blank sheet on the next page. In general, the more information you provide on your thought process, the more you increase your chances of getting partial credit if your final result is incorrect.

```java
int a=2, b=5, m=0;
SillyClass x, y, z;

x = new SillyClass(++a, b);
y = new SillyClass(b, a++);
z = x.firstMethod(b, y);
y = y.firstMethod(m, new SillyClass(2, 4) );
x = new SillyClass(1, 2);
y = new SillyClass(3, 0);
m = x.secondMethod(y);
```
Problem 4

Properly indent on the grid provided in the next page the code listed below. You can select any of the three standard indentation styles we saw in class, but you have to be consistent in your use of that style.

```java
while (x > N)
    y = x + 1;
    if (y > 10) {
        x = x + 2;
        n = y + 1;
    } else
        y = x - 1;
    n = 2 * y;
if (x*x < 50)
    { n = 15;
        y = x + 2;
        if (y > n)
            if (y < 100)
                if (x + y > 0)
                    while (x < 100)
                        x = x + 3;
            else
                if (n > Z)
                    x = 20;
                    y = n + 1;
    }
    x = x + 2;
```