S1007 - Introduction to Computer Science
Practice Midterm
Summer 2003

Name:

CUNIX ID:

You have ninety minutes to answer all of the questions below. Write your answers in the space provided. You should read over the entire exam before you answer any questions and budget your time accordingly.

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
<th>Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>
1. (8 points) Describe an algorithm for finding the second-largest number in a list of integers. The $n$-item list is contained in a sequence of variables $X_0, X_1, \ldots, X_n$. Do not worry about array references; assume you can access any item $X_i$ directly by name. The result should be placed in a variable $Y$. The minimum possible value of an integer is contained in a variable $\text{MIN\_INT}$. Each step of your algorithm should be precisely defined, but it need not be valid Java.

1. $max := X_0$, $Y := X_0$, $i := 1$.
2. If $X_i > max$, $Y := max$, $max := X_i$, go to Step 4.
3. If $X_i > Y$, $Y := X_i$.
4. $i := i + 1$.
5. If $i \leq n$, go to Step 2.
6. Return result $Y$. 

2. (10 points) Find 10 errors (there are more than 10) in the following Java code. The errors are both syntactic (compiler errors) and logical (bugs). For each error, identify the line number and briefly explain how to fix it.

```java
private class Circle {
    public static final float PI = 3.14;

    private double x, y; /* The center of the circle. */
    private double r; /* The radius of the circle. */

    public Circle(double r) {
        this(0.0, 0.0, this.r);
    }

    public double Circle(double x, double y, double r) {
        this.x = x;
        this.y = y;
        this.r = r;

        /* Returns the area of the circle. */
        public double area() {
            return PI * r^2;
        }

        /* Indicates whether a point is on the boundary of the circle. */
        public boolean onCircle(double x1, y1) {
            int dx = x1 - x;
            int dy = y1 - y;
            return dx*dx + dy*dy == r*r;
        }

        /* Creates a series of concentric circles */
        public static void main(String[] args) {
            int i;

            for(int i=0; i<10; i++)
                Circle c = Circle(10.0*i);
                System.out.println("r="," c.r+" A="," +c.area);
        }
}
```
2. (cont’d)

Line 1: `private` is not a valid class modifier. Removing it would give the class package scope.

Line 2: `3.14` is not a `float` literal. Change it to `3.14F`.

Line 8: `this.r` refers to the field `r`. The parameter in the call to `this` should just be plain `r`.

Line 11: Constructors may not have a return type. Omit `double`.

Line 15: There is no closing brace on the constructor.

Line 18: `r^2` is not “r squared”. “^” is the XOR operator. Since `r` isn’t even an integer, the operator is invalid. The expression should be `PI * r * r`.

Line 21: There is no “*/” closing the comment.

Line 22: There is no type specifier on `y1`. It should presumably be a `double`.

Lines 23-24: `int` variables `dx` and `dy` are assigned the value of a floating-point operation without a type cast. The variables should be of type `double`.

Line 25: As written this is an equality comparison between an integer expression and a floating-point expression. Even changing `dx` and `dy` to `double`, as above, testing floating-point results for equality is unlikely to produce the desired results, because of rounding errors. The value should be tested against some threshold of error, e.g.:

```java
    public boolean onCircle(double x1, double y1) {
        static final double EPSILON = 0.01;
        double dx = x1 - x;
        double dy = y1 - y;
        return (dx*dx + dy*dy - r*r) < EPSILON;
    }
```

Line 29: `string` should be `String`.

Line 30-32: `int` `i` is defined twice inside of `main`.

Line 32-34: There are no braces surrounding the `for` loop. As is, the `println` statement will only print the value of the last `Circle`.

Line 34: The call to `area` does not have parentheses.
3. (10 points) Write a class named `IntBits` with a single private field named `num` of type `int`. The methods of your class should be as follows:

- A public constructor that takes a single integer parameter and uses it to initialize `num`.
- A public method named `printBits` that prints out each of the significant bits in `num` (zero bits to the left of the most significant non-zero bit should be ignored). `printBits` takes no parameters and has return type `void`. The output should look something like: “0=0 1=0 2=1 3=0 4=1, etc.”.
- A `main` method that creates ten instances of `IntBits` with the parameters 10, 20, 30, ..., 100 and invokes `printBits` on each one. (Warning: points will be taken off if `main` contains more than five lines of code.) (Hint: Use a loop.)

```java
public class IntBits {
    private int num;

    public IntBits(int num) {
        this.num = num;
    }

    public void printBits() {
        int i = 0;
        if (num == 0)
            System.out.println("0=0");
        else {
            int n = num;
            while (n != 0) {
                System.out.print((i++ "=" + (n & 1) + " ") + "");
                n >>>= 1;
            }
        }
    }

    public static void main(String[] args) {
        for (int i = 10; i <= 100; i += 10) {
            IntBits ibits = new IntBits(i);
            System.out.println("The bits of "+i+: ");
            i.printBits();
        }
    }
}
```
4. (4 points) What is the difference between a public and private member of a Java class?

A public member of a class can be read and modified by any class. A private member of a class can only be read and modified within the class itself.

5. (4 points) What are the values of x and y after the following code executes?

```java
int x = 0;
int y = x++ + ++x;
```

x++ has higher precedence than ++x, so it is evaluated first. The value of x++ is 0 and x becomes 1. ++x increments x and returns the new value 2.

x = 2
y = 0+2 = 2
6. Compute the value of each of the following Java expressions (2 points each):

   a) \( \frac{36}{23 \gg 2} \times (1 \ll 2) + 1 \) + \( 9 \ll 2 \) \% 5

   \[
   23 \gg 2 = \quad \begin{array}{c}
   0001 \ 0111_2 \gg 2 \\
   = \quad 0000 \ 0101_2 \\
   = \quad 5
   \end{array}
   \]

   \[
   (1 \ll 2) + 1 = \quad \begin{array}{c}
   (0001_2 \ll 2) + 1 \\
   = 0100_2 + 1 \\
   = 0101_2 \\
   = \quad 5
   \end{array}
   \]

   \[
   9 \ll 2 = \quad \begin{array}{c}
   0000 \ 1001_2 \ll 2 \\
   = 0010 \ 0100_2 \\
   = \quad 36
   \end{array}
   \]

   \[
   36/5*5 + 36/5 = \quad 36
   \]

   b) \((\text{byte})(127+1) > 0) \&\& (true == false) || (21.0/4.0 > 5)\)

   \[
   (\text{byte})(127+1) > 0 = -128 > 0 \\
   \quad \quad = \quad false
   \]

   \[
   true == false = \quad false
   \]

   \[
   21.0/4.0 > 5 = 5.25 > 5 \\
   \quad \quad = \quad true
   \]

   \[
   false \&\& false || true = (false \&\& false) || true \\
   \quad \quad = \quad true
   \]