Computer Science 302

Exam1

Thursday, Feb 25th, 2016
100 points (10% of final grade)
Instructors: Gary Dahl and Jim Williams

(Family) Last Name: ________________________ (Given) First Name: ________________________

CS Login Name: ________________________ NetID (email): ________________________@wisc.edu

Circle your Lecture:
Lec001 TR 8:00 (Jim)       Lec002 TR 1:00 (Jim)       Lec003 MWF 1:20 (Gary)       Lec004 MWF 9:55 (Gary)

SCANTRON: Fill in these fields (left to right) on the scantron form using a #2 pencil.

1. LAST NAME (family name) and FIRST NAME (given name), as much as there is space to enter.
2. IDENTIFICATION NUMBER is your Campus ID number.
3. SPECIAL CODES ABC: enter your three-digit lecture number: 001, 002, 003, or 004.
4. SPECIAL CODES F: write the letter P for Primary and fill in the bubble for #1.

FILL IN ALL BUBBLES!

This exam contains two parts and is worth a total of 100 points.
Part I contains 20 Simple Choice Questions worth 2 points each, for a total of 40 points possible.
Part II contains 20 Multiple Choice Questions worth 3 points each, for a total of 60 points possible.
You will have 120 Minutes to complete the exam.
Be sure to read through every question completely.

I certify that I will keep my answers covered so that they may not be viewed by another student during the exam or prior to completion of their exam. I also certify that I will not view or in any way use another's work or any unauthorized devices. I understand that I may not make any type of copy of any portion of this exam without express permission from my instructor. I understand that being caught allowing another to view my work or being caught viewing another's work are both violations of this agreement and that either will result in an automatic failure of the course. Any penalty will also be reported to the Deans Office for all involved.

Signature: ________________________

1. Be sure to review the reference pages as needed during the exam.
2. Turn off and put away your cell phone, calculator, Inspector Gadget (watches, glasses, pencils, etc.) now and wait for the proctor to signal the start of the exam.
Disclaimer: the following are provided for your reference only, and the inclusion of information here does not guarantee it will be used on the exam.

Operator Precedence Table:

<table>
<thead>
<tr>
<th>Level</th>
<th>Operator</th>
<th>Description</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>higher</td>
<td>( &lt;expression&gt; )</td>
<td>grouping with parentheses</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>[ ] ( ) .</td>
<td>array index, method call, member access (dot operator)</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>++ --</td>
<td>pre-increment, post-decrement</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>++ -- + - !</td>
<td>post-increment, unary plus/minus, logical negation</td>
<td>right to left</td>
</tr>
<tr>
<td></td>
<td>(type) new * / %</td>
<td>casting and creating object multiplication, division, modulus addition, subtraction, concatenation</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>+ - + &lt;= &gt; &gt;= instanceof == != &amp;</td>
<td>relational and Java’s instanceof operator equality</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>&amp;&amp; %=</td>
<td>conditional AND (short-circuits) conditional OR (short-circuits) ternary conditional assignment</td>
<td>left to right</td>
</tr>
<tr>
<td>lower</td>
<td>= += -= *= /= %=</td>
<td>assignment</td>
<td>right to left</td>
</tr>
</tbody>
</table>

Constants and Methods from the java.lang.Math class:

- Math.PI
- int min(int x, int n)
- int max(int x, int n)
- double min(double x, double n)
- double max(double x, double n)
- double random()
- double pow(double x, double n)
- double sqrt(double n)
- double abs(double n)
- double ceil(double n)
- double floor(double n)
- double sin(double theta)

Other trig methods also available.

Methods from the java.lang.Object class: (all members are public methods)

- String toString()
- boolean equals(Object o)

Returns a String representation of the object. This is the hash code of the instance unless toString() has been overridden.

Returns true if the object referenced as o is the same as this. It is often overridden (redefined) by instantiable classes.
Methods from the java.lang.String class: (* REMEMBER 0-based indexing is used)

- `int length()`  
  Returns number of characters in the String

- `char charAt(int index)`  
  Returns character at the specified index of the String

- `boolean contains(String s)`  
  Returns true iff string s is in this string, otherwise false

- `String toLowerCase()`  
  Returns a new string that is the lowercase version of this string.

- `String toUpperCase()`  
  Returns a new string that is the UPPERCASE version of this string.

- `int indexOf(String s)`  
  Returns the index within this string of the first character of the first occurrence of the specified string s or -1 if not found.

- `boolean equals(String s)`  
  Returns true if the contents of this String is the same as the contents of String s.

- `boolean equalsIgnoreCase(String s)`  
  Returns true iff the contents of this string are the same as that of the string s, ignoring differences in case.

- `String substring(int begin)`  
  Returns a new string that is a substring of this string starting at begin to the end of this string.

- `String substring(int begin, int end)`  
  Returns a new string that is a substring of this string starting at index begin up to but not including end.

- `boolean startsWith(String prefix)`  
  Returns true iff this string starts with the specified prefix, false otherwise.

- `boolean startsWith(String pre, int off)`  
  Returns true iff this string starts at the specified offset off with the specified prefix pre, false otherwise.

- `String [] split(String delimiter)`  
  Splits this String instance into tokens. The delimiter character is used to determine where to split the string. The delimiter token is not returned in any of the token instances.

Methods from the java.util.Random class:

- `new Random()`  
  Creates a new random number generator instance.

- `new Random(int s)`  
  Creates a new random number generator seeded with s.

- `int nextInt()`  
  Returns the next pseudo-random integer value.

- `int nextInt(int n)`  
  Returns the next pseudo-random integer value between 0 (inclusive) and n (exclusive).

- `double nextDouble()`  
  Returns the next pseudo-random double value between 0.0 (inclusive) and 1.0 (exclusive).
Methods from the java.util.Scanner class:

```java
new Scanner(System.in) // Creates a Scanner that reads from the keyboard.
new Scanner(String s) // Creates a Scanner to read the String s.
void close() throws IOException // Closes the stream and any associated file.
boolean hasNext() // Returns true if there’s another token of input.
boolean hasNextInt() // Returns true if the next input is an int value.
boolean hasNextDouble() // Returns true if the next input is a double value.
boolean hasNextLine() // Returns true if there’s another line of input.
String next() // Returns the next word only, as a String.
int nextInt() // Returns the next word only, as an int.
double nextDouble() // Returns the next word only, as a double.
String nextLine() // Returns the next line as a String.
```

Methods from the java.util.Arrays class:

```java
static String toString(E[] array) // Returns a String representation of any type (E[]) array.
static void sort(E[] array) // sorts the specified array in memory type E must be Comparable or Comparable<E>
static int[] copyOf(int[] orig, int newLength) // Copies the specified array, truncating or padding with zeros (if necessary) so the copy has the specified length.
static <E> E[] copyOf(E[] orig, int newLength) // Copies the specified array, truncating or padding with nulls (if necessary) so the copy has the specified length.
```

Methods from java.io.PrintStream class:

(called using System.out.print or System.out.println)

```java
void print( boolean b ) // Prints a boolean value.
void print( char c ) // Prints a character.
void print( char[] s) // Prints an array of characters.
void print( double d ) // Prints a double-precision floating-point number.
void print( int i ) // Prints an integer.
void print( String s ) // Prints a String.
void println( boolean b ) // Prints a boolean and then terminate the line.
void println( char c ) // Prints a character and then terminate the line.
void println( char[] s) // Prints an array of characters and then terminate the line.
void println( double d ) // Prints a double-precision floating-point number and then terminate the line.
void println( int i ) // Prints an integer and then terminate the line.
void println( String s ) // Prints a String and then terminate the line.
```
Part I: Simple Choice (Questions 1-20, 2 points each)

Ex. The return type of the expression `Math.pow(5, 6) * 2` is ____________.

A. int  
B. double

Part II: Multiple Choice (Questions 21-40, 3 points each)

Ex. What is displayed to the user when the following code fragment is executed?

```java
String[] list = { "a", "b", "c" };  
String result = "";  
for (int i = list.length; i > 0; i--)  
    for (int j = 0; j < list[i-1].length(); j++)  
        if ( j < list[j].length() )  
            result += i + ")" + list[3 - j - i] + " ";
System.out.println( result );
```

A. 1)a 2)b 3)c  
B. 3)a 2)b 1)c  
C. 1)c 2)b 3)a  
D. 3)a  
E. none of the above, an ArrayIndexOutOfBoundsException is thrown at runtime.

---

Double-check that you have answered all 40 questions on your SCANTRON bubble sheet. Then when you are done, turn in your SCANTRON sheet along your exam questions. Please have your UW ID ready.