Computer Science 300

Sample Exam

Today's Date
100 points (XX% of final grade)
Instructor Name(s)

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

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

Circle your Lecture:

Lec001 TR 11:00AM (Gary)       Lec002 TR 1:00PM (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 or 002.
4. SPECIAL CODES F: write the letter P for Primary and fill in the bubble for #1.

FILL IN THE BUBBLES CORRESPONDING TO ALL OF ANSWERS!

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 carefully 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 may result in an automatic failure of the exam, the course, and additional consequences in accordance with the Department of Computer Sciences, College of Letters and Sciences, and/or the University of Wisconsin – Madison.

Signature: __________________________________________

1. Be sure to review the reference pages as needed throughout 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>post-increment, post-decrement</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>++ -- + - !</td>
<td>pre-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</td>
<td>right to left</td>
</tr>
<tr>
<td></td>
<td>* / %</td>
<td>multiplication, division, modulus</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>+ - +</td>
<td>addition, subtraction, concatenation</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>&lt; &lt;= &gt; &gt;=</td>
<td>relational and Java’s instanceof operator</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>== !=</td>
<td>equality</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td>&amp; &amp;</td>
<td>conditional AND (short-circuits)</td>
<td>left to right</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>? :</td>
<td>ternary conditional</td>
<td>right to left</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`: Field that represents the constant \( \pi \)
- `Math.abs(double n)`: Returns the absolute value of \( n \)
- `Math.min(double x, double n)`: Returns the smaller of the two values as a double
- `Math.min(int x, int n)`: Returns the smaller of the two values as an int
- `Math.pow(double x, double n)`: Returns \( x^n \)
- `Math.random()`: Returns a random value between 0 (inc.) and 1 (exclusive)
- `Math.sqrt(double n)`: Returns \( \sqrt{n} \)

### Methods from the `java.lang.Object` class:

- `String toString()`: Returns a String representation of the object. This is the hash code of the instance unless `toString()` has been overridden.
- `boolean equals(Object o)`: 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.Integer` class: (which implements Comparable)

- `static int parseInt(String s)`: Converts \( s \) into the corresponding int value.
- `int intValue()`: Return the int value of this `Integer` instance.
- `int compareTo(String n)`: Returns a negative value if this `Integer` is smaller than \( n \), zero if they are equal, and a positive value if this `Integer` is larger than \( n \).
Methods from the java.lang.String class: (which implements Comparable)

int length() Returns number of characters in the String
char charAt(int index) Returns character at the specified index of the 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.
String concat(String s) Returns a new string that is the result of concatenating the String s to the end of this string.
boolean equals(String s) Returns true if the contents of this String is the same as the contents of String s.
String substring(int begin) Returns a new string that is a substring of this string starting at begin to the end of this string.
int compareTo(String s) Returns a negative value if this String is alphabetically earlier than s, zero if they are equal, and a positive value if this String is alphabetically later than s.

The java.lang.Comparable interface:

int compareTo(Object obj) Returns a negative value if this is less than obj, zero if they are equal, and a positive value if this is greater than obj.

The java.lang.Comparable<T> interface:

int compareTo(T obj) Returns a negative value if this is less than obj, zero if they are equal, and a positive value if this is greater than obj.

Methods from java.io.PrintStream class:
(called using System.out.print or System.out.println)

void print ( String s ) Prints a String.
Void print ( Object o ) Prints String representation of Object using String.valueOf(o)
void println( boolean b ) Prints a Boolean value and then terminate the line.
void println( String s ) Prints a String and then terminate the line.
void println( Object o ) Prints String representation of Object using String.valueOf(o) and then terminates the line.

Methods from the java.util.Arrays class:

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>
String toString() Calls toString on each element in the list & returns a single comma separated String of these results.
Methods from the java.util.ArrayList<E> class (*REMEMBER 0-based indexing): Note the E’s below are replaced with the particular ArrayList’s element type.

```
ArrayList<E>()       Constructs an empty list.
ArrayList<E>(int cap) Constructs an empty list with cap initial capacity.
int size()          Returns the number of used elements in this list.
E get(int index)    Returns the item at the specified index in this list.
void add(E item)    Adds the specified item to the end of this list.
void add(int index, E item) Adds the specified item by inserting it into this list at the specified index.
E remove(int index) Removes and returns the item from the specified index.
```

Methods from java.io.PrintWriter class:

```
PrintWriter(String filename) throws FileNotFoundException
    Creates a PrintWriter for the given filename.
PrintWriter(File out) throws FileNotFoundException
    Creates a PrintWriter from out.
void close()
    Closes the stream and associated file.
void print(String s)
    Prints given string.
void println(String s)
    Prints given string followed by a newline.
```

Methods from the java.util.Scanner class:

```
Scanner(System.in)
    Creates a Scanner that reads from the keyboard.
Scanner(String s)
    Creates a Scanner to read the String s.
Scanner(File fn) throws FileNotFoundException
    Create a Scanner to read from file.
void close() throws IOException
    Closes the stream and any associated file.
boolean hasNextInt()
    Returns true if the next input is an int 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.
```

Exception Class Inheritance Hierarchy

```
public class Throwable extends Object
public class Exception extends Throwable
    public class RuntimeException extends Exception
        public class ArithmeticException extends RuntimeException
        public class IndexOutOfBoundsException extends RuntimeException
        public class ClassCastException extends RuntimeException
        public class NullPointerException extends RuntimeException
        public class IOException extends Exception
            public class FileNotFoundException extends IOException
            public class EOFException extends IOException
```
Part I: Simple Choice (Questions 1-20, 2 points each)

#. What type of data does the following expression evaluate to: Math.pow( 5, 6 ) * 2

   A. int
   B. double

...  

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

#. What is printed to the console when the following code 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)c 2)b 1)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.