CSC 1051 Algorithms and Data Structures I

Midterm Examination
February 28, 2019

Name:_____________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Please answer questions in the spaces provided. If you make a mistake or need more space, please use the back of pages and clearly indicate where the answer can be found. Good luck!
1. [20] What output is produced by the following program?

```java
public class SpringBreak {
    public static void main (String[] args) {
        int a = 20, b = 21, c = 22;

        System.out.println("Howdy, here are some numbers: ");
        System.out.println(" a = " + a + " b = " + b +" c = " + c);
        System.out.println ("Wishing you an\awesome \"Spring Break\") ;

        System.out.print ("Escape sequences...\n\t sound relevant ");
        System.out.println ("(keep \this\ in mind)");
        System.out.println();
        a = c;
        c = b;
        b = 30;
        System.out.println("Here are the final values: ");
        System.out.println(" a = " + a + " b = " + b +" c = " + c);
    }
}
```

Output:
2. What gets printed? Please show output as it will appear, or indicate “NO OUTPUT”. If there is an infinite loop, be sure to show some lines of the output followed by “... INFINITE LOOP”.

String word = "SCHOOLED";
for (int i = 0; i < word.length(); i++)
{
    if(i%2 == 0)
        System.out.print(word.charAt(i));
}

int a = 0;
while (a < 10)
{
    a += 5;
    System.out.println(a * 3);
}

int b = 10;
do {
    System.out.println(b);
    b--;
} while (b < 10);

for (int size = 0; size < 7; size++)
{
    System.out.print(size + " => ");
    // CAREFUL! missing break statements!
    switch(size)
    {
    case 1:
        System.out.println ("A");
        break;
    case 2: case 3:
        System.out.println ("B");
        break;
    case 4:
        System.out.print ("C");
    case 5:
        System.out.println ("D");
        break;
    default:
        System.out.println ("E");
    }
}

String season = "Spring";
for (int n = 0; n < season.length(); n++)
{
    System.out.print(n);
    System.out.print(season.charAt(n));
}
3. [ /20]  

a. The following code is supposed to compute and print the maximum of three integers, a, b, c. Without making any changes to the existing code, fill in the blanks to complete the code so that it works correctly.

```java
if (a > b)
    if (c < a)
        max = ___;
    else
        max = ___;
else
    if (b < c)
        max = ___;
    else
        max = ___;
System.out.println("max = " + max);
```

b. Given the following declarations, state the value and type of each of the expressions below.

```java
int num1 = 7, num2 = 3;
double val1 = 9.8;
String word = "gastronomy";
boolean part1 = false;
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>num1 % num2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>word.subString(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(int) val1 * 0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(double) (num1 / num2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>part1</td>
<td></td>
<td>(num1 &gt; num2)</td>
</tr>
<tr>
<td>part1? num1: num2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. [ /20]

a. Write a code fragment that prints the list of numbers 5, 10, 15, 20, 25, ..., 100 (multiples of 5). Be sure to include any necessary variable declarations.

b. The code below prints a pattern of asterisks.
   ➢ Add some code (annotate the code) so that the number of asterisks is printed at the beginning of each row. (e.g., a row with 3 asterisks appears as “3 ***”).
   ➢ Show the output produced from the finished code

```java
for (int row = 1; row < 5; row++)
{
    for (int col = 1; col <= 5; col++)
        System.out.print("*");

    System.out.println();
}
```

Output:

```
1  *
2 ***
3 *****
4  ***
```

c. Assume the following declarations:
   Random rand = new Random();
   NumberFormat money = NumberFormat.getCurrencyInstance();

Write a code fragment that prints a message of the form:
   Congratulations! You have the winning ticket: 3802
   You have won $22,790.00 !!!

where the winning ticket number is a randomly generated number in the range 1000-9999 and the amount is also randomly generated, in the range $20,000-$29,999 and displayed appropriately as a monetary amount.
5. [_____/ 20]
Construct an algorithm that inputs some non-zero integers from the user, terminated with a zero (sentinel value). The algorithm should print the numbers entered and compute and print their sum. After the sum is printed, print a goodbye message.

Example: If the numbers 5 -3 8 0 are entered as input, the algorithm should print:
5
-3
8
Sum = 10
Goodbye
[Note that the terminating zero should NOT be printed.]

Directions:
Write your algorithm by rearranging and structuring elements chosen from the list below, using indentation to show structure. Do not use anything else and note that not all of these are needed, but you may use one of them more than once, if necessary.

```plaintext
input num
input sum
num = 0
sum = 0
count = 0
num += 1
if (sum == num)
  if (sum < num)
    if (num < sum)
      else
        while (num < sum)
          while (sum < num)
            num = num + 1
            sum = sum + 1
            count = count + 1
            sum = sum + num
            sum = num + 1
            while (count < sum)
            while (num != sum)
            print "Goodbye"
```
## REFERENCE MATERIAL

### Random class

<table>
<thead>
<tr>
<th>double</th>
<th>nextDouble()</th>
<th>Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>nextInt(int n)</td>
<td>Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator’s sequence.</td>
</tr>
</tbody>
</table>

### Math class

<table>
<thead>
<tr>
<th>static double</th>
<th>abs(double a)</th>
<th>Returns the absolute value of a double value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static double</td>
<td>cos(double a)</td>
<td>Returns the trigonometric cosine of an angle.</td>
</tr>
<tr>
<td>static double</td>
<td>pow(double a, double b)</td>
<td>Returns the value of the first argument raised to the power of the second argument.</td>
</tr>
<tr>
<td>static double</td>
<td>random()</td>
<td>Returns a double value greater than or equal to 0.0 and less than 1.0.</td>
</tr>
<tr>
<td>static long</td>
<td>round(double a)</td>
<td>Returns the closest long to the argument.</td>
</tr>
<tr>
<td>static double</td>
<td>sin(double a)</td>
<td>Returns the trigonometric sine of an angle.</td>
</tr>
<tr>
<td>static double</td>
<td>sqrt(double a)</td>
<td>Returns the correctly rounded positive square root of a double value.</td>
</tr>
</tbody>
</table>

### String class

<table>
<thead>
<tr>
<th>char</th>
<th>charAt(int index)</th>
<th>Returns the char value at the specified index.</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>compareTo(String anotherString)</td>
<td>Compares two strings lexicographically.</td>
</tr>
<tr>
<td>int</td>
<td>indexOf(int ch)</td>
<td>Returns the index within this string of the first occurrence of the specified character.</td>
</tr>
<tr>
<td>boolean</td>
<td>isEmpty()</td>
<td>Returns true if, and only if, length() is 0.</td>
</tr>
<tr>
<td>int</td>
<td>length()</td>
<td>Returns the length of this string.</td>
</tr>
<tr>
<td>String</td>
<td>replace(char oldChar, char newChar)</td>
<td>Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.</td>
</tr>
<tr>
<td>boolean</td>
<td>startsWith(String prefix)</td>
<td>Tests if this string starts with the specified prefix.</td>
</tr>
<tr>
<td>String</td>
<td>substring(int beginIndex)</td>
<td>Returns a new string that is a substring of this string.</td>
</tr>
<tr>
<td>String</td>
<td>substring(int beginIndex, int endIndex)</td>
<td>Returns a new string that is a substring of this string.</td>
</tr>
<tr>
<td>String</td>
<td>toLowerCase()</td>
<td>Converts all of the characters in this String to lower case using the rules of the default locale.</td>
</tr>
<tr>
<td>String</td>
<td>trim()</td>
<td>Returns a copy of the string, with leading and trailing whitespace omitted.</td>
</tr>
</tbody>
</table>