CSC 1051 Algorithms and Data Structures I

Final Examination
December 19, 2018

Name:______________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</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>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL 100

Please answer questions in the spaces provided. Please be legible. If you make a mistake or need more space, use backs of pages - clearly indicate where the answer can be found.

Good luck and best wishes for the holidays!
1. (_____/ 10) 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”.

a) ```java
int a = 4;
do {
    a--;
    System.out.println(a);
} while (a < 4);
```

Output:

b) ```java
String[] n = {"one", "two");
for (String word : n)
    for (String otherWord : n)
        System.out.println("Tra la la " + word + " and " + otherWord);
```

Output:

c) ```java
int[] stuff = {2, 3, 4, 5};
for (int i = 2; i <= 3; i++)
    System.out.println(stuff[i]);
```

Output:
(Question 1, continued)

d) for (int size = 0; size < 6; size ++)
   {
      System.out.print(size + " => ");
      // CAREFUL! missing break statements!
      switch(size)
      {
         case 0:
            System.out.println ("A");
            break;
         case 1:
            System.out.print ("B");
         case 2: case 3:
            System.out.println ("C");
            break;
         default:
            System.out.println ("E");
      }
   }

e) for (int x = 1; x <= 3; x++)
   for (int y = 0; y < x; y++)
   {
      System.out.print(x + "\t" + y);
      System.out.println();
   }

Output:
2. [ /10] Short answer questions

a) Suppose you look up a method in the Java API and find this method heading: 
   `public String mystery(double what)`

Fill in the following information about this method:

| Method name: ____________________________ |
| Method return type: ________________ |
| Required parameters for the method: **(how many? of what type(s)?)** |
| ______________________________________________________________________ |

b) Given a Random object named `rand`, write a Java expression that produces a value in the range `[a ... b]`, where `a` and `b` are variables of type `int` that have positive values, such that `a<b`. For example, if `a` and `b` are 23 and 25, respectively, then the expression should produce one of the numbers: [23, 24, 25].

c) What are the values of the following expressions?

`(double)(5 / 10) ________          (int) (0.36 * 10) _______`

d) What are the values of the following expressions after running the code shown in the box to the right?

```
String word = "New Year";
String message = "";
int n = 0;
while (n < 3)
{
    message = word.charAt(n) + message;
    n++;
}

boolean answer = n < 0;
```

```
word.length() ___

word.charAt(1)_____

answer _________

word.substring(2,6)_______

message ____________________
```

a) Write a Java method `maxOfThree()` with three parameters of type `double` that returns a value of type `double` that is the largest of the three given values. For example, `maxOfThree(35.2, 45.7, 22.8)` should return 45.7. Note that the method should not print anything.

b) Write a method `maxArray` with one parameter, an array of `double` that calculates and returns the maximum value stored in the array. For example, if the array contains the values {-3.5, -4.0, 5.4, 1.6}, the method should return the value 5.4. Note that the method should not print anything.
c) Write a method \texttt{starry} with one parameter, an \texttt{int} that returns a \texttt{String} made up of the number of asterisks given by the parameter. For example, \texttt{starry(5)} should return the String "*****". Note that the method should not print anything.
4. (_____/ 10) Consider the code for the start method of a JavaFX application, below, which produces the graphic shown.

a) Draw a group hierarchy diagram that includes all the graphical elements (line, circle, rect, ellipse, quote, message, root).

b) Using the grid on the next page sketch the graphic in the coordinate system.
   • Be sure to draw and position all the shapes precisely in the grid.
   • Mark the center of the circle and note its coordinates on the sketch.
   • Mark the endpoints of the line and note their coordinates on the sketch.
   • Mark the bottom right corner of the rectangle and note its coordinates on the sketch.

```java
public void start(Stage primaryStage) {
    Line line = new Line(300, 80, 150, 200);
    Circle circle = new Circle(250, 70, 20);
    circle.setFill(Color.BLUE);
    Rectangle rect = new Rectangle(50, 70, 250, 60);
    rect.setStroke(Color.RED);
    rect.setStrokeWidth(2);
    rect.setFill(null);
    Ellipse ellipse = new Ellipse(200, 150, 100, 20);
    ellipse.setFill(Color.PINK);
    Text quote = new Text(120, 100, "Happy Holidays!!!!");
    Group message = new Group(ellipse, quote);
    Group root = new Group( message, line, rect, circle);
    Scene scene = new Scene(root, 400, 300);
    primaryStage.setTitle("Greeting");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```
Draw diagrams showing the contents of array table after execution of the following code fragments. (Be sure to include indices in your diagram).

```java
int[][] table = new int[2][3];
for (int i = 0; i < 2; i++)
    table[i][2] = i + 1;
```
5. (____/10) Consider the following program and suppose the file data-in.txt is used as the input file. Show the contents of the file data-out.txt after execution of the program.

```
import java.util.Scanner;
import java.io.*;

public class FileOutputFinalF18 {
    public static void main(String[] args) throws IOException {
        Scanner inFile;
        PrintWriter outFile;

        inFile = new Scanner (new File("data-in.txt"));
        outFile = new PrintWriter("data-out.txt");

        int count = 0;

        while (inFile.hasNext())
        {
            count++;
            String chunk = inFile.nextLine();
            if (chunk.contains("in"))
                outFile.println(count + " " + chunk );
        }
        outFile.close();
        System.out.println("After " + count + " we say goodnight");
    }
}
```

```
data-in.txt
Rudolph, a youthful reindeer buck possesses an unusual luminous crimson nose.
```

```
data-out.txt
```

Rudolph, a youthful reindeer buck possesses an unusual luminous crimson nose.
6. (_____/ 10) Draw diagrams showing the contents of the arrays after execution of the following code fragments. **Be sure to include the indices for all the arrays.**

a) 
```java
int[] a = {3, 5, 2};
int[] b = new int[8];

for (int i=0; i<a.length; i++)
    b[i] = a[i];

for (int i=b.length - 1; i >= a.length; i--)
    b[i] = a[0];
```

b) 
```java
int[][] table = new int[4][5];
for (int i=0; i < 3; i++)
    table[i][i+1] = i + 10;
```

c) 
```java
char[][] table = new char[3][3];
String sample = "Relax. Exams can be fun. ";
int count = 0;
for (int i=0; i < 3; i++)
    for (int j=0; j < 3; j++)
    {
        table[i][j] = sample.charAt(count);
        count++;
    }
```
Consider the following program:

```java
import java.util.Scanner;
public class Lab2C {
    public static void main (String[] args) {
        int age = 0;
        String name = "";
        String petName = "";
        Scanner scan = new Scanner(System.in);

        System.out.println("Please enter name: ");
        name = scan.next();

        System.out.println("Please enter age: ");
        age = scan.nextInt();

        System.out.println("Please enter pet name: ");
        petName = scan.next();

        System.out.println("Hello, my name is " + name + " and I am " + age + " years old.");
        System.out.print("I love Villanova, but I miss");
        System.out.println("my pet " + petName);
    }
}
```

a) This code can throw InputMismatchException. Explain the following:

• Give an example of a situation that would cause this exception

• In which line would the exception be thrown? __________

• If you were to handle the exception, which lines need to go in a try block?

from __________ to __________

• What would be a reasonable way to handle the exception?
b) This code does not throw BadWordException, in the way we defined it in our lab, but it could have this feature added. Discuss the following:

- Give an example of a situation where it might be reasonable to throw the BadWordException

- Where in the code should that exception be thrown?

- Assuming that the BadWordException class is compiled and available in the same folder as your program, what else would you need to do in order to achieve this added feature? (check all that apply and explain your answer.)

  - Add “throws BadWordException” to the header of the main method
  - Add a try block with a catch for BadWordException
  - Add some more code to the program to throw BadWordException
8. (_____/ 10) Refer to the UML class diagram for the Movies/DVDCollection/DVD program on the next page.

a) We now wish to add a method to shuffle the DVD's in a DVDCollection. Do this by repeatedly exchanging the positions of random items in the collection. How many exchanges to make? You can assume that if you perform twice as many exchanges as there are items in the collection, that would be sufficient to render the collection shuffled.

Complete the code for the shuffle() method:

```java
public void shuffle()
{
    Random rand = new Random();
}
```

b) Complete the code of the Movies client, as directed by the comments.

```java
public class Movies
{
    public static void main(String[] args) throws IOException
    {
        DVDCollection movies = new DVDCollection();

        movies.addDVD("The Godfather", "Coppola", 1972, 24.95, true);
        movies.addDVD("District 9", "Blomkamp", 2009, 19.95, false);
        movies.addDVD("Iron Man", "Favreau", 2008, 15.95, false);
        movies.addDVD("Casablanca", "Curtiz", 1942, 19.95, false);
        System.out.println(movies);

        // shuffle the movies so far

        // add another movie (your choice of values for the parameters)

        // print out the movies in the collection
    }
}
Movies

+ main (args : String[]) : void

DVDCollection

- collection : DVD[]
- count : int
- totalCost : double

+ addDVD(title : String, director : String, year : int, cost : double, blueray : boolean) : void
+ toString() : String
- increaseSize() : void

DVD

- title : String
- director : String
- year : int
- cost : double
- blueray : boolean

+ toString() : String
9. (____/ 20)
a) Fill in some code for a Circle class, following guidelines given through comments.
   public class Circle
   {
      // instance variables
      private double xPosition;
      private double yPosition;
      private double radius;
      private boolean visible; // true=visible; false=hidden

      // constructor
      public Circle(double x, double y, double z)
      {
         xPosition = x;
         yPosition = y;
         radius = z;
         visible = false; // starts out hidden, always
      }

      // Another constructor – no parameters,
      // instantiates a circle of radius 1 positioned at (0,0)

      // move(): Move circle to position (x, y) given by parameters

      // toString(): Returns appropriate string describing Circle
      //   eg: “Visible circle of radius 7.302 at (0.34, -4.2222)”
// getRadius(): Returns the radius of the circle

// sizeUp(): increases size to twice original (no parameters)

// reveal(): make the Circle visible

// hide(): make the Circle be hidden

b) Write client code that uses the Circle class:
   • Instantiate a Circle object with radius 7.302 at {0.3, -4.2} and assign to variable circ1.

   • Instantiate a Circle object with radius 1 at {0, 0} and assign to a variable circ2. (Use 2\textsuperscript{nd} constructor.)

   • Move circ2 to position {0.3, -4.2}, and make both circles visible.

   • Declare and instantiate an array of 10 Circle objects (no need to initialize them).
Boolean Expressions

The only valid words for a boolean type are true and false.

Boolean variables:

```java
boolean aboveAgeLimit = false;
boolean usePlural = hours > 1;
```

Java Conditional statements alter the linear flow of control. They use boolean expressions to determine what to do next.

Example:

```java
if (credits == 0) System.out.println("CPA: None");
else if (gpa < 3) System.out.println("\n\tGPA: " + gpa);
```

The Conditional Operator Syntax

```java
condition ? expression1 : expression2
```

- If the condition is true, expression1 is evaluated; if it is false, expression2 is evaluated.
- The value of the entire conditional operator is the value of the selected expression.
- Example: Rewrite this:

```java
if (n > 0)
  System.out.print("positive");
else
  System.out.print("negative");
```

using conditional operator:

```java
System.out.print(n > 0 ? "positive": "negative");
```

Switch Statement in general

```java
switch (expression)
{
  case value1:
    statement-list1;
  case value2:
    statement-list2;
  case value3:
    statement-list3;
  default:
    statement-list-n;
}
```

If expression matches value2, control jumps to here.

If expression matches value3, control jumps to here.

If none of the values match the expression, control jumps to here.

While loop

```java
while (condition)
{
  statement
}
```

The body of a do loop executes at least once.

Do loop

```java
do
{
  statement
}
while (condition)
```

For: a loop with built in "counter"

```java
for (int count = 0; count < 5; count++)
{
  System.out.println(count);
  System.out.println(count);
  count++;
}
```

while loop equivalent

```java
int count = 0;
while (count < 5)
{
  System.out.println(count);
  count++;
}
```
## Random class

<table>
<thead>
<tr>
<th>double</th>
<th><code>nextDouble()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th><code>nextInt(int n)</code></th>
</tr>
</thead>
<tbody>
<tr>
<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>
<td></td>
</tr>
</tbody>
</table>

## Math class

<table>
<thead>
<tr>
<th>static double</th>
<th><code>abs(double a)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the absolute value of a double value.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static double</th>
<th><code>cos(double a)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the trigonometric cosine of an angle.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static double</th>
<th><code>pow(double a, double b)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the value of the first argument raised to the power of the second argument.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static double</th>
<th><code>random()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a double value greater than or equal to 0.0 and less than 1.0.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static long</th>
<th><code>round(double a)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the closest long to the argument.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static double</th>
<th><code>sin(double a)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the trigonometric sine of an angle.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static double</th>
<th><code>sqrt(double a)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the correctly rounded positive square root of a double value.</td>
<td></td>
</tr>
</tbody>
</table>

## String class

<table>
<thead>
<tr>
<th>char</th>
<th><code>charAt(int index)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the char value at the specified index.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th><code>compareTo(String anotherString)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compares two strings lexicographically.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th><code>indexOf(int ch)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the index within this string of the first occurrence of the specified character.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>isEmpty()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns true if, and only if, length() is 0.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th><code>length()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the length of this string.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>replace(char oldChar, char newChar)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>startsWith(String prefix)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tests if this string starts with the specified prefix.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>substring(int beginIndex)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a new string that is a substring of this string.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>substring(int beginIndex, int endIndex)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns a new string that is a substring of this string (incl. chars from beginIndex through endIndex-1)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>toLowerCase()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Converts all of the characters in this String to lower case using the rules of the default locale.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>contains(String anotherString)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns true if, and only if the string contains anotherString as a substring.</td>
<td></td>
</tr>
</tbody>
</table>
Zero.java -- updated

```java
public class Zero {
    public static void main(String[] args) {
        int numerator = 10;
        int denominator = 0;
        try {
            System.out.println(numerator/denominator);
        } catch (ArithmeticException problem) {
            System.out.println("Bad division");
        }
        System.out.println("this will not print");
    }
}
```

---

catch clauses – handle particular types of exceptions. Examples:

- `StringIndexOutOfBoundsException`
  - Thrown by a `charAt` method
  - Thrown by a `substring` method

- `NumberFormatException`
  - Thrown by a `parseInt` method
  - If the string does not contain an integer

- `ArithmeticException`
  - Thrown by division by zero

- `IndexOutOfBoundsException`
  - Bad index in array

---

```java
// ProductCodes.java Author: Lewis/Loftus
// Demonstrates the use of a try-catch block.
import java.util.Scanner;
public class ProductCodes {
    public static void main(String[] args) {
        String code;
        char zone;
        int district, valid = 0, banned = 0;
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter product code (XXX to quit): ");
        code = scan.nextLine();
        continue
```