CSC 1051 Algorithms and Data Structures I

Happy Cinco de Mayo!!!!

Final Examination
May 5, 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>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100</strong></td>
<td>****</td>
</tr>
</tbody>
</table>

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 a great summer!*
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".

```java
int a = 3;
while (a < 8)
{
    if ((a%2)==0)
        System.out.println(a);
    a++;
}
```

```
int[] b = {4, 5, 6, 7};
for (int i= 2; i < b.length; i++)
    System.out.println(b[i] * 3);
```

```
int a = 7;
do
{
    a--;
    System.out.println(a);
} while (a < 5);
```

```
String[] word = {"up", "down"};
for (String w1 : word)
    for (String w2 : word)
        System.out.println("yes " + w1 + " " + w2 + " " + w1);
```

```
boolean flipFlop = false;
int[] c = {2, 5, 1, 3, 4};
for (int i = 0; i < c.length; i++)
{
    System.out.print(i + " ");
    System.out.println(c > 3 ? " yes" : " no");
}
2. (_____/10)
We studied the problem of repeatedly obtaining input and performing a calculation, for example, computing the circumference of a circle given its radius, using the following algorithm:

*Rewrite this algorithm, modifying it so that it uses a while structure to repeat the processing of each input in two different ways.*

a) Keep computing circumferences and ask each time whether to keep going.

*Variables:*

*Algorithm:*

```
Variables: radius, circ

Algorithm:
input radius
circ = 2 * radius * PI
print circ
```

b) Keep computing circumferences until user inputs -1 for the radius *(sentinel value)*

*Variables:*

*Algorithm:*

```
```

c) Compute the circumference of 5 circles *(exact count).*

*Variables:*

*Algorithm:*

```
```
3. (_____ / 10) Complete the following code fragment following the hints given as comments. (This is similar to the project 5 problem of expanding the queue.)

String queue = "::JJJJJJ:KKKKKK:AAAAAA"

int t1 = scan.nextInt();
int t2 = scan.nextInt();
Random rand = new Random();

//*** 1 numTimes is a number in the range [t1 .. t2]
int numTimes =

//*** 2 personSymbol = a random char in the range ['A' .. 'Z']
char personSymbol =

//*** 3 updating the String representing the queue:
// add a colon followed by numTimes the personSymbol.
// For example, if numTimes = 3 and personSymbol = 'B',
// the queue will become "::JJJJJJ:KKKKKK:AAAAAA:BBB"

//
4. (________/ 10)
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, `max(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**.
5. (_____ / 10) Consider the code for the start method of a JavaFX application, below, which produces the graphic shown to the right.

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

```java
public void start(Stage primaryStage) {
    Line line = new Line(35, 60, 150, 170);
    Circle circle = new Circle(100, 65, 20);
    circle.setFill(Color.BLUE);
    Rectangle rect = new Rectangle(60, 70, 250, 60);
    rect.setStroke(Color.RED);
    rect.setStrokeWidth(2);
    rect.setFill(null);
    Ellipse ellipse = new Ellipse(200, 100, 150, 50);
    ellipse.setFill(Color.PINK);
    Text quote = new Text(120, 100, "CINCO DE MAYO!!!!");
    Group message = new Group(ellipse, quote);
    Group root = new Group(message, line, rect, circle);
    Scene scene = new Scene(root, 400, 200);
    primaryStage.setTitle("Cinco de Mayo");
    primaryStage.setScene(scene);
    primaryStage.show();
}
```
6. (_____/ 10) Fill in code for an Employee class, following guidelines in comments.

```java
public class Employee {
    // instance variables
    String name;
    String position;
    double hourly; // hourly wages
    int yearHired

    // constructor: Construct object with w, x, y, and z as
    // name, position, hourly pay rate, and
    // year hired, respectively.

    // toString(): Returns a String corresponding to object.

    // getYearHired(): Accessor for yearHired

    // wages(): Given the number of hours worked (a value of type double,
    // returns the wages of this employee, calculated based on
    // hourly rate, for up to 40 hours and 1.5 overtime of
    // hourly rate for hours over 40.

}
```
7. (______/ 10) Using the Employee class from the previous question:
   a) Draw a UML diagram for the Employee class.

b) Write client code that uses the Employee class:
   Instantiate an Employee object with name “Lucia Rodriguez”, with position “software engineer”, hourly rate $42.50, hired in 2013. Assign it to a variable named coderBoss.

c) Write client code that uses the Employee class:
   Suppose you have three Employee objects e1, e2 and e3 and that e2 worked 44.5 hours last week, whereas e1 and e3 both worked 40 hours. Write some client code to calculate and print: 1) the wages for each employee and 2) the average of their wages. (Note: it is NOT necessary to format as currency).

d) Write client code that uses the Employee class:
   Suppose you have two Employee objects e1, e2. Write some client code that uses the getYearHired() and toString() methods of the Employee class to print the information for the Employee who has been with the company the longest (i.e., hired earlier).
8. (_____ / 10) Consider the following program:

```java
public class TwoDArray {
    // Creates a 2D array of integers, fills it with increasing
    // integer values, then prints them out.
    public static void main (String[] args)
    {
        int[][] table = new int[3][9];
        // Load the table with values
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                table[row][col] = row * 10 + col;
        // Print the table
        for (int row=0; row < table.length; row++)
            { for (int col=0; col < table[row].length; col++)
                System.out.print (table[row][col] + " ");
                System.out.println();
            }
    }
}
```

a) The output produced is shown below. Circle the entries for `table[1][2]` and for `table[2][3]`

```
  0  1  2  3  4  5  6  7  8
10 11 12 13 14 15 16 17 18
20 21 22 23 24 25 26 27 28
```

b) On the next page, rewrite the code of the main method, so that instead of a 2D array of `int`, it creates a 2D array of 3x3 values of type `boolean` as follows:

- use an initializer list to instantiate and initialize the array `table` to represent the following configuration, where the asterisk indicates a `true` value at that position in the array (no asterisk means `false`). For example, `table[1][3]` is `false`, whereas `table[2][1]` is `true`.
- Modify the loop that prints the values of the array so that it prints an asterisk or blank, for each element in the table. The output does NOT have to show row and column labels, it should look like this:

```
# | 0 1 2
---
0 | * * *
1 | * * *
2 | * *
```
//******************************************************************************
// TwoDArray.java    Author: Lewis/Loftus
// Demonstrates the use of a two-dimensional array.
//******************************************************************************
public class TwoDArray
{
    //----------------------------------------------------------
    // Creates a 2D array of integers, fills it with increasing
    // integer values, then prints them out.
    //----------------------------------------------------------
    public static void main (String[] args)
    {
        
    }
}
9. [____/ 20] Consider the program ProductCodesS18Final.java on the last page. The code contains several errors and omissions that need to be corrected.

a. Line 26 is causing an error:

```java
26          System.out.println("\"products.txt\"\ + " not found");
```

• What kind of error is it? (Syntax, Runtime, or Logical) _________________
• Show how to correct it

What is the problem and how would you correct it?

b. The following elements are missing from the code, but the code has comments // *1, // *2, ... in their places. Note that there is one more number than there are elements (not all the placements are used). Write the number corresponding to each one for its correct placement.

```java
// *_____ fileScan = new Scanner(System.in);
// *_____ throws IOException
// *_____ outfile.close();
// *_____ import java.io.*;
// *_____ PrintWriter outfile = new PrintWriter ("out.txt");
```

c. Suppose the code has been corrected according to a and b, above, so now you are to run it with a text file `products.txt`, (available in the same folder as the code) with the contents below. Some of the input from this file, however, causes the code to throw exceptions. For each line in this text file, mark it either as “ok” or specify which of the following exceptions will be thrown; choose from one of the following:

- NumberFormatException
- FileNotFoundException
- StringIndexOutOfBoundsException
- arrayIndexOutOfBoundsException

```text
products.txt

AAAAAAAAAAA
AAA3446AAR
AAA1111AAR
AAA111111
BBB2222AAR
CCC11111111
DDD1234
DDD123456R
```

```text

AAAAAAAAAAA
AAA3446AAR
AAA1111AAR
AAA111111
BBB2222AAR
CCC11111111
DDD1234
DDD123456R
```
d. Suppose you would like to catch and handle NumberFormatException by skipping this input using the following catch clause:

```java
    catch (NumberFormatException exception)
    {
        System.out.println("Bad code: " + code);
    }
```

In order to do this, you need to include some lines in the try block and add the catch immediately after it. The lines included should be:

from line ___________ to line ___________ (inclusive).

e. Suppose the code is now fully debugged and has been improved to handle all exceptions caused by ill-formed input in the file. What can you expect for the output from this program and what will be the contents of the file out.txt?

**OUTPUT:**

out.txt
ProductCodesF18Final.java

import java.util.Scanner;

public class ProductCodesS18Final {
    public static void main(String[] args) {
        String code;
        char zone;
        int district, valid = 0, banned = 0;

        File inFile;
        inFile = new File("products.txt");
        Scanner fileScan;

        try {
            fileScan = new Scanner(inFile);
        } catch (IOException e) {
            System.out.println("products.txt" + " not found");
            System.out.println("Input from keyboard instead");
        }

        while (fileScan.hasNext()) {
            code = fileScan.nextLine();
            zone = code.charAt(9);
            district = Integer.parseInt(code.substring(3, 7));
            valid++;
            outfile.print(code);
            if (zone == 'R' && district > 2000) {
                banned++;
                outfile.print("tB");
            }
            outfile.println();
        }

        System.out.println("# of valid codes entered: " + valid);
        System.out.println("# of banned codes entered: " + banned);
    }
}

Villanova University  CSC 1051  www.csc.villanova.edu/~map/1051  Dr. Papalaskari
The Conditional Operator Syntax

- The value of the entire conditional operator is the value of the boolean expression.

- If the condition is true, expression_1 is evaluated.

- If the condition is false, expression_2 is evaluated.

- The example: Read this:

```
if (a > 0) {
    System.out.println("positive");
} else {
    System.out.println("negative");
}
```

Java Conditional Statements

These statements are used to determine the linear flow of control in a program. They use boolean expressions to determine what to do next.

Example:

```
if (credits == 0) {
    System.out.println("credits
```

```java
else {
    System.out.println("gpa = " + gpa);
```

The Switch Statement in General

```
switch (expression) {
    case VALUE_1:
        statement_1;
        break;
    case VALUE_2:
        statement_2;
        break;
    default:
        statement_default;
        break;
}
```

```java
if (expression) {
    statement_if
}
else {
    statement_else
```

The body of a do loop executes at least once.
Example: Stars.java

```java
public class Stars {
    public static void main(String[] args) {
        final int MAX_ROWS = 10;
        for (int row = 1; row <= MAX_ROWS; row++) {
            for (int star = 1; star <= row; star++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

---

UML Class Diagrams

UML = Unified Modelling Language

- Example: A UML class diagram for the RollingDice program.

- Class diagram components:
  - Class: RollingDice
  - Main method signature: `main(String[] args)`
  - Methods: `roll()`, `getFaceValue()`, `toString()`

---

Arrays

- Declaration:
  ```java
double[] scores = new double[10];
```

- Initialization:
  ```java
  double[] scores = {7.9, 8.7, 9.4, 8.2, 9.8, 8.7, 8.1, 7.4, 9.1};
  ```

- Size of array:
  ```java
  System.out.println(scores.length);  // 10
  ```

Arrays as Parameters

- Example:
  ```java
  double[] courseGrade = new double[3][10];
  int[] ratings = {4, 3, 3, 1, 4, 3, 1, 0, 3, 4};
  System.out.println(average(ratings));
  ```

- Method definition for `average`:
  ```java
  public static double average(int[] a) {
      double sum = 0;
      for (int num : a) {
          sum += num;
      }
      return sum / a.length;
  }
  ```

---

2D Arrays - Overview

- Example 2D array:
  ```java
  double[][] courseGrade = new double[3][10];
  ```

---

can be omitted!
try / catch

- Create a try block surrounding code that we think may cause an exception

- catch clause has code to tell it what to do
  - the “exception handler”
  - Can have multiple catch clauses
    - One for each type of exception thrown by try block

- If no exception is thrown, processing continues following the try statement (skips catch clauses)

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");
    }
}
```

But is our trace gone???

- No
- Methods exist to get the trace and system error message
  - Method: `getMessage()`
    - Returns a string explaining the reason the exception was thrown

  - Method: `printStackTrace()`
    - Prints the call stack trace indicating where the error occurred

Unchecked Exceptions