CSC 1051 Algorithms and Data Structures I

Midterm Examination
March 2, 2017

Name: ______________KEY____________

<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>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>6</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 for some other reason need more space, please use the back of pages and clearly indicate where the answer can be found. Good luck!
1. (___/10) What gets printed? Please show output as it will appear, or indicate “NO OUTPUT”, or show some of the output followed by “INFINITE LOOP.”

```java
int a = 3;
while (a >= 0)
{
    System.out.println(a);
    a++;
}
```

Output:

3
4
5
... INFINITE LOOP

```java
int a = 3;
while (a > 0)
{
    System.out.println(a * 2);
    a--;
}
```

Output:

6
4
2

```java
int a = 0;
while (a > 0)
{
    System.out.println(a);
    a = a + 3;
}
```

Output:

NO OUTPUT

```java
String word = "visual";
int x = 0;
while (x < word.length())
{
    System.out.print(word.charAt(x));
    System.out.print(word.charAt(x));
    x++;
}
```

Output:

vviissuuaall
2. Using the coordinate system below representing the graphics area of an applet, draw the graphics displayed by the applet, identifying positions MID and TOP (in terms of their use in the applet), and the shapes RECTANGLE, OVAL, LINE.

```java
import javax.swing.JApplet;
import java.awt.*;

public class FunShapes extends JApplet
{
    //––––––––––––––––––––––––––––––––––––––––––––––––––––––––
    // Draws something...
    //––––––––––––––––––––––––––––––––––––––––––––––––––––––––
    public void paint (Graphics page)
    {
        final int MID = 40; // MID
        final int TOP = 20; // TOP

        page.drawOval (MID-10, TOP-20, 20,40); // OVAL
        page.drawRect (0, 0, 30, 40); // RECTANGLE
        page.drawLine (0, 60, 100, 60); // LINE
    }
}
```
3. [________/ 20]
Construct an algorithm that inputs three integers $n$, $m$, and $d$. The algorithm should count up by increments of $d$, starting from $n$ and stopping at $m$. After it is done printing the numbers, it should print a goodbye message. You can assume that the numbers entered will be positive integers with $n < m$, so you do NOT need to check for mistakes in the input.

**Example 1:** If the inputs are $7, 25, 4$, then the algorithm should print:

```
 7  11  15  19  23
Goodbye
```

**Example 2:** If the inputs are $12, 32, 5$, then the algorithm should print:

```
 12  17  22  27  32
Goodbye
```

**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:

```
input n
input m
input d
input n + d
n = 0
m= 0
d = 0
n = n+d
n = m
m = n
d = d+1
n = n+1
if (n > 0)

if (n < m)
else
while (n < m)
while (n <= m)
while (n > m)
while (d > 0)
while (d >= 0)
while (d < m)
print n
print m
print d
print "Goodbye"
```

```
input n
input m
input d

while (n <= m)
  print n
  n = n + d

print "Goodbye"
```

**NOTE:** other correct solutions are possible.
4. What output is produced by the following program?

```java
public class OneMoreTime {
    public static void main (String[] args) {
        int x = 12, a = 13, b = 14, c = 15;

        System.out.println("Howdy, here are some numbers: ");
        System.out.println("x = " + x + " a = " + a + " b = " + b + " c = " + c);
        System.out.println ("I promise\nthis is the \n"last time\"");
        System.out.println ("you have to do this ");
        System.out.println ("so\nplease count the "\"\"s carefully!");

        if (a>0)
            if (b<0)
                x = x + 5;
            else
                if (a>5)
                    x = x + 4;
                else
                    x = x + 3;
        else
            x = x + 2;

        System.out.println();
        System.out.println("Here are the numbers after the if/else: ");
        System.out.println("x = " + x + " a = " + a + " b = " + b + " c = " + c);
        System.out.println();
        c = a;
        a = b;
        b = 2000;
        System.out.println("Here are the final values: ");
        System.out.println("x = " + x + " a = " + a + " b = " + b + " c = " + c);
    }
}
```

Output:

```
Howdy, here are some numbers: x = 12 a = 13 b = 14 c = 15
I promise this is the "last time"
you have to do this so please count the "\"s carefully!

Here are the numbers after the if/else: x = 16 a = 13 b = 14 c = 15

Here are the final values: x = 16 a = 14 b = 2000 c = 13
```
5. [ /20] Short answer questions.

a) For each of the following expressions, indicate the order in which the operations are performed by writing a number beneath each operator.

\[
\begin{array}{cccc}
a / (b - c) / e + f & a / b - d * e + f \\
2 & 1 & 3 & 4 \\
\end{array}
\]

b) The code below is supposed to print the numbers from 1 to 10, but it has an error.

```java
int count = 1;
while (count <= 10) {
    System.out.println (count);
    count++;
}
```

i) Describe the error and how to correct it? __ Incorrect semicolon after while __
ii) If not corrected, what, if anything gets printed? ____ Nothing -- infinite loop
iii) Is this a syntax, runtime, or logical error? Logic

c) Given a Random object named `gen`, what range of values are produced by the following expressions?

- `gen.nextInt(6)` ______ 0 to 5
- `gen.nextInt(30) + 10` ______ 10 to 39
- `gen.nextInt(3) - 8` ______ -8 to -6

a) Given the following declarations:

```java
int iResult, num1 = 7, num2 = 3;
double fResult, val1 = 9.0;
boolean status, part1 = false;
```

What result is assigned by each of the following assignment statements?

<table>
<thead>
<tr>
<th>Source code</th>
<th>Result stored</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fResult = (num1 + 2)/ 2;</code></td>
<td>4.0</td>
</tr>
<tr>
<td><code>iResult = num1 % num2;</code></td>
<td>1</td>
</tr>
<tr>
<td><code>fResult = val1 / 2;</code></td>
<td>4.5</td>
</tr>
<tr>
<td><code>fResult = (double) num1 / 2;</code></td>
<td>3.5</td>
</tr>
<tr>
<td><code>status = part1 &amp;&amp; (num1 &gt; num2);</code></td>
<td>false</td>
</tr>
<tr>
<td>`status = part1</td>
<td></td>
</tr>
</tbody>
</table>

a) Fill in the blanks

```
String word = "overdrive";
int num = word.length();

num _9_ word.charAt(1)_v_
word.charAt(4)_d_ word.charAt(num-1)_e_
```

```
Output:
vararava
```

b) Suppose the String variable `word` is already initialized (similar to part (a), but not necessarily with the same value). Write a Java code fragment to prints `word`, BACKWARDS.

```
int count = word.length() - 1;
while (count >= 0)
{
    System.out.print(word.charAt(count) + "a");
    count--;
}
```

c) Suppose you would like to display 1000 “snowflakes” on your applet by drawing white ovals of varying sizes, 2-5 pixels in width and height. Fill in the blanks to complete the following code fragment to do this.

```
final int WIDTH = 500, HEIGHT = 300; // applet dimensions
Random rand = new Random();
int count = 1;
while (count <=1000)
{
    int x = rand.nextInt(WIDTH);  // Change this line
    int y = rand.nextInt(HEIGHT);  // Change this line
    int size = rand.nextInt(4) + 2;
    page.fillOval(x, y, size, size);
    count++;
}
```

Note: Alternative solution if we are fussy and insist that the snowflakes are fully within the picture:

```
int x = rand.nextInt(WIDTH - 5);
int y = rand.nextInt(HEIGHT - 5);
```

Also accepting solutions where the values 500 or 300 are used.
Drawing a Line

```java
page.drawLine (10, 20, 150, 45);
page.drawLine (150, 45, 10, 20);
```

Drawing a Rectangle

```java
page.drawRect (50, 20, 100, 40);
```

Drawing an Oval

```java
page.drawOval (175, 20, 50, 80);
```

Drawing an Arc

- An arc is defined by an oval, a start angle, and an arc angle.

REFERENCE MATERIAL
Random class

double nextDouble()
Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0

int nextInt(int n)
Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.

Math class

static double abs(double a)
Returns the absolute value of a double value.

static double cos(double a)
Returns the trigonometric cosine of an angle.

static double pow(double a, double b)
Returns the value of the first argument raised to the power of the second argument.

static double random()
Returns a double value greater than or equal to 0.0 and less than 1.0.

static long round(double a)
Returns the closest long to the argument.

static double sin(double a)
Returns the trigonometric sine of an angle.

static double sqrt(double a)
Returns the correctly rounded positive square root of a double value.

String class

char charAt(int index)
Returns the char value at the specified index.

int compareTo(String anotherString)
Compares two strings lexicographically.

int indexOf(int ch)
Returns the index within this string of the first occurrence of the specified character.

boolean isEmpty()
Returns true if, and only if, length() is 0.

int length()
Returns the length of this string.

String replace(char oldChar, char newChar)
Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.

boolean startsWith(String prefix)
Tests if this string starts with the specified prefix.

String substring(int beginIndex)
Returns a new string that is a substring of this string.

String substring(int beginIndex, int endIndex)
Returns a new string that is a substring of this string.

String toLowerCase()
Converts all of the characters in this String to lower case using the rules of the default locale.

String trim()
Returns a copy of the string, with leading and trailing whitespace omitted.