CSC 1051 Algorithms and Data Structures I

Midterm Examination
October 9, 2014

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>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>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</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. 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!

a) How many bits/bytes are needed to store a color picture that is 200 pixels wide and 400 pixels high under each of the following schemes? Express your answer as approximate number of KB or MB, etc., as appropriate. Show your work and fill in the answers below:

Total # pixels: 80,000
200x400 = 80,000

<table>
<thead>
<tr>
<th>Scheme</th>
<th># bytes</th>
<th># bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitmap (1bit/pixel)</td>
<td>10 KB</td>
<td>80,000 bits or 80kbits</td>
</tr>
<tr>
<td>Greyscale (1byte/pixel)</td>
<td>80 KB</td>
<td>640,000 or 640Kbits</td>
</tr>
<tr>
<td>RGB (3 bytes/pixel)</td>
<td>240 KB</td>
<td>1,920,000 or ≈2Mbits</td>
</tr>
</tbody>
</table>

b) List all binary codes that can be made with 3 bits

000
010
011
001
111
100
101
110

c) With $n$ bits we can have $2^n$ different binary codes
2. [ 10] Refer to the program below. Next to each word in the list, choose the most fitting description:

- reserved word
- constant
- variable
- method

```java
import java.util.Scanner;

public class Age {
    // Reads the user's age and prints comments accordingly.
    public static void main (String[] args) {
        final int MINOR = 21;
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter your age: ");
        int age = scan.nextInt();
        System.out.println ("You entered: " + age);
        if (age < MINOR)
            System.out.println ("Youth is a wonderful thing. Enjoy.");
        System.out.println ("Age is a state of mind.");
    }
}
```
3. What output is produced by the following program?

```java
public class OneMoreTime {
    public static void main (String[] args)
    {
        System.out.print ("Howdy there... ");
        System.out.println ("I promise
            \"last time\"");
        System.out.print ("you have to do this ");
        System.out.println ("so\nplease count the \"s carefully!\"");

        System.out.println ("And numbers... "+20+30);
        System.out.println ("Is this different? ..."+(20+30));
    }
}
```

Output:

```
Howdy there... I promise
this is the "last time"
you have to do this so
please count the \\
's carefully!
And numbers... 2030
Is this different? ... 50
```

b) Write a single `println` statement that would output

```
"Hello"
  "Goodbye"
```

```
System.out.println ("\"Hello\"\n\t"Goodbye\"");
```
4. [ /10] Given the following declarations:

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

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

<table>
<thead>
<tr>
<th>Source code</th>
<th>Result stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>fResult = (num1 + 2)/ 2;</td>
<td>4</td>
</tr>
<tr>
<td>iResult = num1 % num2;</td>
<td>1</td>
</tr>
<tr>
<td>fResult = val1 / 2;</td>
<td>4.5</td>
</tr>
<tr>
<td>fResult = (double) num1 / 2;</td>
<td>3.5</td>
</tr>
<tr>
<td>status = part1 &amp;&amp; (num1 &gt; num2);</td>
<td>false</td>
</tr>
<tr>
<td>status = part1</td>
<td></td>
</tr>
</tbody>
</table>
Below is the code for the Snowman applet and the image it produces.

a) Sketch the image produced by modifying the code indicated by the arrow, as follows:

```java
final int TOP = 20;
```

b) Add some code to the program below to make the snowman look like he is holding a green ball, i.e.:
- add a “hand” – 10 pixel long horizontal line, start at arm
- add a green ball, 10 by 10 pixels, resting on the hand.
- be sure the ball is green, but the hand is black
- see illustration below

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

public class Snowman extends JApplet {
    public void paint(Graphics page) {
        final int MID = 150;
        final int TOP = 50;

        page.setColor(Color.cyan);
        page.fillRect(0, 0, 300, 175); // sky
        page.setColor(Color.blue);
        page.fillRect(0, 175, 300, 50); // ground
        page.setColor(Color.yellow);
        page.fillOval(-40, -40, 80, 80); // sun

        page.setColor(Color.white);
        page.fillOval(MID-20, TOP, 40, 40); // head
        page.fillOval(MID-35, TOP+35, 70, 50); // upper torso
        page.fillOval(MID-50, TOP+80, 100, 60); // lower torso

        page.setColor(Color.black);
        page.fillOval(MID-10, TOP+10, 5, 5); // left eye
        page.fillOval(MID+5, TOP+10, 5, 5); // right eye

        page.drawArc(MID-10, TOP+20, 20, 10, 190, 160); // smile

        page.drawLine(MID-25, TOP+60, MID-50, TOP+40); // left arm
        page.drawLine(MID+25, TOP+60, MID+55, TOP+60); // right arm

        page.drawLine(MID-20, TOP+5, MID+20, TOP+5); // brim of hat
        page.fillRect(MID-15, TOP-20, 30, 25); // top of hat

        //*** code for left hand holding green ball goes here ***

        page.drawLine(MID-60, TOP+40, MID-50, TOP+40); // left hand
        page.setColor(Color.red);
        page.fillOval(MID-60, TOP+30, 10, 10); // red ball in hand
    }
}
```

hand and ball width is the same = 10 pixels
6. [  / 10] What gets printed?

Please show output as it will appear or indicate “NO OUTPUT” or “INFINITE LOOP.”

```java
int a = 4;
while (a > 0) {
    System.out.println(a);
    a++;
}
```

Output:
```
4
5
6
... INFINITE LOOP
```

```java
int a = 4;
while (a > 0) {
    System.out.println(a);
    a = a - 2;
}
```

Output:
```
4
2
```

```java
int a = 1;
while (a < 4) {
    a++;
    System.out.println(a);
}
```

Output:
```
2
3
4
```

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

Output:
```
NO OUTPUT
```
7. [ / 10] Let's look at the problem of repeatedly obtaining input and performing a calculation, for example, computing the area of a circle given its radius, using the following algorithm:

```plaintext
Variables: 
  radius, area
Algorithm: 
  input radius 
  area = radius * radius * PI 
  print area
```

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

**a) Compute the areas of 5 circles (exact count).**

```plaintext
Variables: 
  radius, area, count
Algorithm: 
  count = 1 
  while (count <= 5) 
    input radius 
    area = radius * radius * PI 
    print area 
    count = count + 1
```

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

```plaintext
Variables: 
  radius, area
Algorithm: 
  input radius 
  while (radius != -1) 
    area = radius * radius * PI 
    print area 
    input radius
```
8. [ / 10] Suppose you look up a method in the Java API and find this method heading:

```
void getThisDone(String x, double y)
```

a) Fill in the following information about this method:

- Method name: `getThisDone`
- Method return type: `void`
- Required parameters for the method: two, a String and a double

b) Suppose this method is defined in class `Midterm` and you have an object `mid1` of that class. Give an example of the use of the dot operator to invoke this method. (Hint: You need to use `Midterm` or `mid1` but not both.)

```
getThisDone.mid1("hello world", 4.5);
```
9. [ / 10]

a) Suppose the String variable `message` is already initialized. Write a Java code fragment that prints out `message`, backwards.

```java
int count = message.length() - 1;
while (count >= 0)
{
    System.out.print(message.charAt(count));
    count--;
}
```

b) Write some code that uses the Math class to compute and print out the value of \( \cos(\pi/2) \).

```java
System.out.println(Math.cos( Math.PI /2) );
```

c) Write code to declare and instantiate an object of the Random class (call the object reference variable `rand`).

```java
Random rand = new Random();
```

d) Write an expression using the `nextInt` method that generates random numbers in the range 1 to 1000, including the endpoints. Use the version of the `nextInt` method that accepts a single integer parameter.

```java
rand.nextInt(1000) + 1
```
10. (______/ 10)
Construct an algorithm that inputs 10 positive numbers and prints the maximum of these numbers.

Example: If the numbers input are 44 7 31 22 53 16 21 48 72 60, the output should be: 
\[
\text{max} = 72
\]

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.

\[
\begin{align*}
\text{num} & = 0 \\
\text{count} & = 0 \\
\text{count} & = 1 \\
\text{max} & = 0 \\
\text{count} & = \text{count} + 1 \\
\text{num} & = \text{max} \\
\text{max} & = \text{num} \\
\text{print} & \text{ "max = " max} \\
\text{input num} & \\
\text{if} & (\text{num} > \text{max}) \\
\text{max} & = \text{num} \\
\text{count} & = \text{count} + 1 \\
\text{while} & (\text{count} \leq 10) \\
\text{while} & (\text{count} < 10) \\
\text{while} & (\text{num} \neq \text{max}) \\
\end{align*}
\]

\[
\begin{align*}
\text{count} & = 0 \\
\text{max} & = 0 \\
\text{while} & (\text{count} < 10) \\
\text{input num} & \\
\text{if} & (\text{num} > \text{max}) \\
\text{max} & = \text{num} \\
\text{count} & = \text{count} + 1 \\
\text{print} & \text{ "max = " max} \\
\end{align*}
\]

ALTERNATIVELY we can count from 1 to 10:

\[
\begin{align*}
\text{count} & = 1 \\
\text{max} & = 0 \\
\text{while} & (\text{count} \leq 10) \\
\text{input num} & \\
\text{if} & (\text{num} > \text{max}) \\
\text{max} & = \text{num} \\
\text{count} & = \text{count} + 1 \\
\text{print} & \text{ "max = " max} \\
\end{align*}
\]
Drawing a Line

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

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

---

Drawing a Rectangle

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

---

Drawing an Oval

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

---

Drawing an Arc

- An arc is defined by an oval, a start angle, and an arc angle:
**Random class**

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

- `int nextInt()`
  Returns the next pseudorandom, uniformly distributed `int` value from this random number generator’s sequence.

- `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.

**Some methods of the 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**

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

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

- `int length()`
  Returns the length of this string.

- `String toUpperCase()`
  Converts all of the characters in this `String` to upper case using the rules of the default locale.