Final Exam

COMP 14-090 Summer I 2000

June 26, 2000

200 points


2. Write all answers on the test itself. Do not write any answers in a blue book or on scratch paper.

3. If you finish before 11:00, you may bring your test to the front of the room.

4. Keep your answers short and to the point. Longer is not necessarily better.

5. Budget your time carefully. Read over the entire exam before starting.

6. Write legibly. If I can’t read it, you can’t get credit for it.

7. Write your initials on the back of every page of this exam.

8. You do not need to put comments in any code you write. You will not be graded for coding style. But, if I can’t understand it you can’t get credit for it.

9. Make sure your answer clearly indicates the result (versus your scratch work). Draw a box around your answers.

10. Assume any code segment is embedded in a correct program.

11. Assume all variables have been appropriately declared before they are used. Their type will be obvious from the usage.

12. When showing program output, you do not have to indicate the exact spacing but do show when an output starts on a new line.

I pledge that I have neither received nor given unauthorized aid on this examination.

Signed: _________________________________________________________

Printed Name: ________________________________________________
<table>
<thead>
<tr>
<th>Question</th>
<th>Maximum</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

200 points  Total Score _________
Section I – Multiple Choice / Fill in the Blank [4 points each]

Circle the option that correctly fills in the blank.

1. A(n) ________ allows a single variable to reference multiple values.
   a) method    b) array    c) byte    d) primitive data type

2. ________ variables are non-static and are declared inside the class, but outside any method.
   a) Instance    b) Class    c) Local    d) Static

3. A(n) ________ connects machines in the same building or room.
   a) WAN    b) ARPA    c) TCP    d) LAN

4. The result of compiling Java source code is called ____________.
   a) compiler    b) executable    c) bytecode    d) machine code

5. class FinalExam
   {
       private int numQuestions;

       public double calcScore (int numMissed)
       {
           double percentage = (double) numMissed / numQuestions;
           return (percentage);
       }
   }

   The scope of numMissed is the ____________.
   a) class    b) method    c) variable    d) object
Section II – True or False [4 points each]

Evaluate each Boolean expression as true or false, given

```java
int count = 0, sum = 54;
double x = 4.3, y = 1.2;
boolean wrong = true;
```

6. `(wrong && (sum > 60))` ____
7. `((x > 5) || !(sum == 55))` ____
8. `!((y < 1.0) && (x > 4))` ____
9. `((count != 4) || (sum < 100) && (wrong))` ____
10. `(!((count <= 0) && (sum > 25)) || ((x < 10) || !(y > 2)) && (!wrong))` ____

Section III – Short Answer

11. [4 points] List the four parts of every loop.

   1. 
   2. 
   3. 
   4. 

12. [5 points] Describe the difference between formal and actual parameters.
Initials __________________________
13. [5 points] Describe the difference between private and public variables.

14. [6 points] Given the operation of the following sort, identify the type of sort (selection or insertion).

<table>
<thead>
<tr>
<th>Original array</th>
<th>3 7 1 6 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>After pass 1</td>
<td>3 7 1 6 2</td>
</tr>
<tr>
<td>After pass 2</td>
<td>1 3 7 6 2</td>
</tr>
<tr>
<td>After pass 3</td>
<td>1 3 6 7 2</td>
</tr>
<tr>
<td>After pass 4</td>
<td>1 2 3 6 7</td>
</tr>
</tbody>
</table>

15. [30 points] Translate the following English sentences into Java code.

   a) [10 points] Print all of the elements of an array of doubles called grades.

   b) [10 points] Print the minimum value in the grades array.

   c) [10 points] Print all of the odd integers between 0 and 100 inclusive.
16. [10 points] Write the exact output from the following code fragment:

```java
int[] array = {3, 6, 9, 12, 37};

int count = 0;
boolean found = false;

while (!found) {
    if (array[count++] < 10) {
        System.out.println ("Items left to look at:");
        for (int i = count; i<array.length; i++)
            System.out.print (array[i] + " ");
    } else
        found = true;
System.out.println();
}
```

17. [20 points] The class Can includes a public method, IsCrushed, which returns a boolean.

a) [5 points] Write the code to declare a variable that can refer to an object of the Can class.

b) [5 points] Write the code to create a new object of the Can class and assign it to the variable declared in part a.

c) [10 points] Write the code to call the IsCrushed method using the object you created in part b and print the results of the method call.
18. [15 points] Write the exact output from the following program:

class Arr
{
    private double[] array = {1.4, 2.2, 3.6, 4.7, 5.1};

    public int getLength()
    {
        return (array.length);
    }

    public double getElement (int ind)
    {
        return (array[ind]);
    }

    public void swap (int ind1, int ind2)
    {
        double temp = array[ind1];
        array[ind1] = array[ind2];
        array[ind2] = temp;
    }

    public static void main (String[] args)
    {
        Arr myArray = new Arr();

        for (int i=0; i<myArray.getLength()-2; i++)
            myArray.swap (i, i+2);

        for (int i=0; i<myArray.getLength(); i++)
            System.out.print (myArray.getElement(i) + " ");
    }
}
Initials ________________
19. [30 points] The method `mergeArrays` interleaves (merges, **not** concatenates) the contents of the two integer arrays (of the same size) into a new array that is returned from the method. (Note: the two arrays passed as parameters are unchanged.)

   a) [10 points] Write the exact output of the following code fragment:

   ```java
   int[] busy = {4, 6, 23, 15, 100};
   int[] work = {45, 78, 1, 43, 89};

   int[] busywork = mergeArrays (busy, work);

   for (int i=0; i<busywork.length; i++)
       System.out.print (busywork[i] + " ");
   ```

   b) [20 points] Write the `mergeArrays` method.
Initials ________________
20. [35 points] Write nested for loops that produce the following output. (Hint: don’t try to print the spaces and the # inside the same inner loop.)

```
  #
  ###
  #####
  #######
```