What is your name?: ___________________________(0 points)

There are three sections:
   I. True/False. . . . . . . . . . . . . .60 points; (30 questions, 2 points each)
   II. Multiple Choice  . . . . . . . .40 points; (8 questions, 5 points each)
---------------
100 points total

This test is worth 15% of your final grade. For the true/false and multiple choice problems you must put your answers on the bubble form. Put your answers to the short answer questions on these sheets, which must also be turned in. This test is open book and open notes. You have 50 minutes.

• For the True/False questions, if the answer is True, fill in a on your answer form. If the answer is False, fill in b on your answer form.
• For the multiple choice problems, select the best answer for each one and select the appropriate letter on your answer sheet.
• Be careful - more than one answer may seem to be correct. Some questions are tricky.

I. True/False: (2 points each)

T    F       1. If a Scanner is used for input in a program, then it must be declared just inside the class, but outside of the main method.

T    F       2. Any code written with multiple if - else statements can be written using a switch - case statement.

T    F       3. Both the break statement and the continue statement are ways to jump out of a loop and resume execution at the code following the loop.

T    F       4. The output of the statement below is: 2b or not 2b

```
    int x = 4/2;
    char c = 'b';
    System.out.printf( "%db or not 2%c", x, c);
```

T    F       5. The following code is valid (compiles and runs) in Java:

```
    while( true)
    ;
```

T    F       6. The following code in Java stores in variable sum the sum of odd integers less than 20:

```
    for(int x=1, sum=0; x<=20; x++)
    if( ! x%2) sum+=x;
```
7. In a compound if statement of the form
   
   ```java
   if ( condition1 && condition2 && condition3) ...
   ```
   all three conditions must be evaluated each time to determine whether or not the code inside the if statement will be executed.

8. The following code prints the words: `Between a rock and a hard place`
   ```java
   int x = 5;
   System.out.print("Between ");
   if (7 > x > 3)
     System.out.print("a rock ");
   else
     System.out.print("a sponge ");
   System.out.print(" and a hard place");
   ```

9. The following code prints the words: `c = C`
   ```java
   char c='B';
   switch (c){
     case ('A'): c='B';
     case ('B'): c='C';
     case ('C'): c='D';
     break;
   }
   System.out.println("c = " + c);
   ```

10. If we want to compare characters in two different Strings, we must first pull the characters out into individual char variables before the comparison can be done.

11. A Constructor is special initialization code that is run when an object is created.

12. One way to tell the difference between a Constructor and a method is that a Constructor has no return type.

13. The name of a Constructor must be the same as the class name.

14. If you don’t define a default Constructor, the system will supply one for you.

15. The output of the code shown below is: 7513
   ```java
   int y = 5;
   int x = 7;
   int z = x+++y;
   System.out.println("" + x + y + z);
   ```

16. When implementing a menu in your code, good programming practice suggests that a while loop is preferable to a do loop.

17. The output of the code below is the value: 18
   ```java
   int x=3, y=6, z=2;
   System.out.println(x + y * z);
   ```
18. Assume the code shown below, where method *swapValues* is called. Output of this segment of code is:  
Values are: 2 7

```java
int x = 7;
int y = 2;
swapValues(x, y);
System.out.println("Values are: "+ x + " "+ y);
// ... other code

public void swapValues(int num1, int num2)
{
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
```

19. Assume the code shown below, where method *swapValues1* is called. Output of this segment of code is:  
Values are: 3 8

```java
int[] numbers = {3, 8};
swapValues1(numbers[0], numbers[1]);
System.out.println("Values are: "+ numbers[0] + " "+ numbers[1]);
// ... other code

public void swapValues1(int num1, int num2)
{
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
```

20. Assume the code shown below, where method *swapValues2* is called. Output of this segment of code is:  
Values are: 4 9

```java
int[] numbers = {9, 4};
swapValues2(numbers);
System.out.println("Values are: "+ numbers[0] + " "+ numbers[1]);
// ... other code

public void swapValues2(int[] numbers)
{
    int temp = numbers[0];
    numbers[0] = numbers[1];
    numbers[1] = temp;
}
```

21. The following code prints the words:  
First one

```java
String word = "First";
switch (word){
    case ("First"): System.out.println("First one"); break;
    case ("Second"): System.out.println("Second one"); break;
    case ("Third"): System.out.println("Third one"); break;
}
```
22. You can have Constructors with different method names, but for this to work you must leave off the return value and declare them as public.

23. According to our text, the name of a Java class (e.g. Date) should be the same as the filename (e.g. Date.java)

24. If you don’t implement the toString() method in a class, then using an instance of that class in a System.out.println(...) statement displays a number.

25. The output of the program segment below is the text: Nope ***

```java
boolean x = false;
if (x = true)
    System.out.print("Nope");
else
    System.out.print("Yup");
System.out.print(" *** ");
```

26. Assuming we have a String named theWord, in Java we can check for its size using: theWord.length()

27. Every method must have the return type declared unless the method doesn’t return anything, in which case the return type can be left off.

28. A static method can not access non-static instance variables.

29. A method can call itself in Java.

30. If method A( ) calls method B( ) in Java, and method B( ) calls method C( ), then method C( ) may not call methods A( ) or B( ), since that would create an endless loop.
II. Multiple Choice (5 pts. each)

31. Consider the code given below. If its output is:

\[
\begin{array}{cccc}
8 & 12 & 16 & 20 \\
10 & 15 & 20 & 25 \\
12 & 18 & 24 & 30
\end{array}
\]

what are the values for variables \( n, r, \) and \( k \)?

for( int j=k; j<=n; j++) {
    for( int i=2; i<=r; i++) {
        System.out.printf("%5d",i*j);
    }
    System.out.println();
}

a) int n=6, r=5, k=4;
b) int n=5, r=4, k=6;
c) int n=3, r=4, k=4;
d) int n=4, r=3, k=4;
e) None of the above

32. Assume you are doing a numbers guessing game, where you are attempting to guess a number between 0 and 260 (this is a binary search). After each guess, you are told if you need to guess higher or lower. What is the maximum number of guesses you would need to find the number?

a) 8 
b) 9 
c) 10 
d) 11 
e) None of the above

33. Consider the program method given below.

```java
public void problem33( int[] theNumbers)
{
    int x = theNumbers[ 0];
    for (int i=0; i<theNumbers.length; i++) {
        if (theNumbers[i] < x)
            x = theNumbers[i];
    }
    System.out.println( x);
}
```

Its output is:

a) The minimum value in array \textit{theNumbers} 
b) The maximum value in array \textit{theNumbers} 
c) The average value in array \textit{theNumbers} 
d) The sum of the values in array \textit{theNumbers} 
e) None of the above
34. What is the output of the code given in the two columns below when a `Confuse` object is created and method `startUp()` is called?

<table>
<thead>
<tr>
<th>class Confuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>private int x=0;</td>
</tr>
<tr>
<td>private int y=0;</td>
</tr>
<tr>
<td>public Confuse()</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>x = 5; y = 7;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>private void first(int s, int t)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>x += 1; y += 2;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>private void second(int y)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>x += 4; y = 6;</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>private void third(int s, int t)</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>s = 5; t = 2;</td>
</tr>
<tr>
<td>setXY( s, t);</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>public void startUp()</td>
</tr>
<tr>
<td>{</td>
</tr>
<tr>
<td>int x=0;</td>
</tr>
<tr>
<td>first( x, y);</td>
</tr>
<tr>
<td>second( x);</td>
</tr>
<tr>
<td>third( x, y);</td>
</tr>
<tr>
<td>display();</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

| private void setXY( int x, int y) |
| { |
| this.x += x; |
| this.y += y; |
| second( this.y); |
| } |
| private void display() |
| { |
| System.out.println(x + " " + y); |
| } |

| a) 4 6 |
| b) 7 11 |
| c) 13 17 |
| d) 6 5 |
| e) None of the above |

35. What is the output of the code given below when `problem35Driver()` is called?

| public void problem35Driver() |
| { |
| String phrase = "jumbo shrimp"; |
| // create character array from String |
| char[] theArray = phrase.toCharArray(); |
| problem35( theArray); |
| // display array contents |
| System.out.print( theArray); |
| } |

| public void problem35(char[] w) |
| { |
| int x = w.length\2; |
| char c; |
| for (int i=0; i<x; i++) { |
| c = w[i]; |
| w[i] = w[x-i-1]; |
| w[x-i-1] = c; |
| } |

| a) The contents of the original array in reverse order |
| b) The contents of the original array in the original order |
| c) The contents of the original array with half of the characters reversed |
| d) The original array with characters rearranged so they are neither in the original nor reversed order |
| e) None of the above |
36. What is the output of the following code when it is called with \textit{methodA}(3,2)\ ?

\begin{verbatim}
public int methodA( int x, int y)
{
    System.out.println( methodB( y, x));
}

public int methodB( int x, int n)
{
    if (n == 0)
        return 0;
    else
        return x + methodB( x, n-1);
}
\end{verbatim}

\begin{enumerate}
\item[a)] 5
\item[b)] 6
\item[c)] 8
\item[d)] 9
\item[e)] None of the above
\end{enumerate}

37. What is the output of the following code when it is called with \textit{methodC}(512)\ ?

\begin{verbatim}
public int methodC( int x)
{
    System.out.println( methodD( x));
}

public int methodD( int x)
{
    if (x == 0)
        return 0;
    else
        return 1 + methodD( x/2);
}
\end{verbatim}

\begin{enumerate}
\item[a)] 0
\item[b)] 2
\item[c)] 9
\item[d)] 64
\item[e)] None of the above.
\end{enumerate}
38. What is the output of the code given below when called with:

ClassE classEinstance = new ClassE(2);

```java
class ClassD
{
    private int x;

    public ClassD( int value)
    {
        x = value;
        method1( x);
    }

    private void method1( int x)
    {
        x++;  
        ClassE instance1 = new ClassE( x);
    }
} // end of classD

class ClassE
{
    private int x=2;

    public ClassE(int x)
    {
        if ( this.x < 4) {
            ClassD instance1 = new ClassD( x);
        }
        x++;  
        System.out.print(" "+x);
    }

    public void method1( int x)
    {
        x = x + 3;
    }
} // end of classE
```

a) 2 5  
b) 5 2  
c) 5 3  
d) 3 5  
e) None of the above.