What is your name? ____________________________ (4 points)

This test has the following sections:
I. True/False ...................... 60 points; (30 questions, 2 points each)
II. Multiple Choice .............. 36 points; (6 questions, 6 points each)
-------
96 + 4 points for name = 100 points total

This test is worth 10% of your final grade. You must put your answers on the bubble form. All code is in Java unless stated otherwise. This test is open book and open notes. 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. You have 50 minutes.

I. True/False: (2 points each) On your bubble form fill out a for true and b for false.

T F 1. A Java application must have a method called main to start running the program.

T F 2. Scratch programs submitted to the Scratch website run the same when run from the web site or when run from a local computer.

T F 3. A variable that is declared as a constant in Java has the keyword const placed in front of the type name, such as in this example: const int x=7;

T F 4. The printf command in Java is particularly useful for determining how many decimal places to display for a float or double value.

T F 5. Within Eclipse a Java program file name must be the same as the class name within that file. In other words a class called Date must be stored in a file called Date.java

T F 6. To display the quote character as part of a System.out.println statement put two quotes in a row.

T F 7. Equality of Strings is done using the compareTo() method.

T F 8. To see if variable x is equal to or greater than variable y, we could use:

```java
if( x => y) {
    System.out.println( "x => y");
}
```

T F 9. Changing the indentation of a program changes what a program means to the Java compiler.

T F 10. Single-line style comments using // can be nested inside block-style comments using /* */

T F 11. The following code results in multiple lines of output.

```java
System.out.print( "Yes");
System.out.print( " or ");
System.out.print( "No");
```
T  F   12.  The following code results in multiple lines of output.

\begin{verbatim}
System.out.print( "Yes\n or \n No");
\end{verbatim}

T  F   13.  If two methods had the return types, names and parameters as shown below it would cause a compiler error. Assume the other parts of the method exist and are written correctly.

\begin{verbatim}
public int f1()
public int f1( char c)
\end{verbatim}

T  F   14.  If two methods had the return types, names and parameters as shown below it would cause a compiler error. Assume the other parts of the method exist and are written correctly.

\begin{verbatim}
public int f2()
private int f2()
\end{verbatim}

T  F   15.  If two methods had the return types, names and parameters as shown below it would cause a compiler error. Assume the other parts of the method exist and are written correctly.

\begin{verbatim}
int f3( int x)
int f3( int y)
\end{verbatim}

T  F   16.  If two methods had the return types, names and parameters as shown below it would cause a compiler error. Assume the other parts of the method exist and are written correctly.

\begin{verbatim}
int f4( int x)
char f4( int x)
\end{verbatim}

T  F   17.  Any Java program that could be written using methods could be rewritten with all the code in \textit{main()}.  

T  F   18.  The following variable declaration would compile: \texttt{int Int;}

T  F   19.  The following variable declaration would compile: \texttt{int 2B;}

T  F   20.  The following variable declaration would compile: \texttt{char c="";}

T  F   21.  To end a Java program the \texttt{break} statement may be used.

T  F   22.  A loop that prompts for input and makes moves in a Tic-Tac-Toe program would best be implemented as a \texttt{for} loop.

T  F   23.  A loop that displays menu options and handles user input would best be implemented using a \texttt{do} loop.

T  F   24.  Excessive comments in a program can slow it down.
T F 25. The section of code shown below would compile and run and give as output: 3.5 Done

```java
int x = 7;
int y = 2;
System.out.print( x/y);
System.out.println(" Done");
```

T F 26. The section of code shown below would compile and run and give as output: Yes Done

```java
String first = "Bears";
String last = "Packers";
if( first.equals( last))
    System.out.println("Yes ");
else
    System.out.println("No ");
System.out.println("Done");
```

T F 27. The section of code shown below would compile and run and give as output:

```java
x and y are: 4 5
```

```java
int x = 3;
int y = 5;
int z = x+++y;
System.out.println( "x and y are: " + x + " + " + y);
```

T F 28. The section of code shown below would compile and run and give as output: Yes Done

```java
boolean x = false;
if( x = true)
    System.out.print("Yes ");
else
    System.out.print("No ");
System.out.println("Done");
```

T F 29. The following code prints the words: watch out

```java
char c='e';
switch (c){
    case 'a': System.out.print("w");
    case 'e': System.out.print("a");
    case 'i': System.out.print("t");
    case 'o': System.out.print("c");
    case 'u': System.out.print("h");
}
System.out.println(" out");
```

T F 30. The output of the following code is: Smaller Done

```java
int x = 3;
if( x < 5)
    System.out.println("Smaller ");
    if( x < 3 ) {
        System.out.print("than 4 ");
    }
else
    System.out.println("Bigger ");
System.out.println("Done ");
```
II. Multiple Choice (6 pts. each)

31. Consider method *looping* shown below. For positive numbers, how would you best describe its return value?

```java
public int looping( int x, int y)
{
    int z=1;
    for( int i=0; i<y; i++) {
        z = z * x;
    }
    return z;
}
```

a)  $x + y$
b)  $x * x$
c)  $x * y$
d)  $x$
e)  None of the above

32. Consider the code shown below. What value does it return when called with *repetition(2,6)* ?

```java
public int repetition( int x, int y)
{
    int z=0;
    for( int i=0; i<y; i++) {
        x=i;
        if( i%2 == 0) {
            x = i * -1;
        }
        z = z + x;
    }
    return z;
}
```

a)  Some value less than 1
b)  1
c)  2
d)  3
e)  None of the above
33. Consider the code shown below. If its output is:
   12 16 20
   15 20 25
   18 24 30
   21 28 35
what are the values for variables start, end, first and last?

   for( int i=start; i<=end; i++) {
      for( int j=first; j<last; j++) {
         System.out.printf("%4d", i+j);
      }
      System.out.println();
   }

   a) start = 7, end=3, first=6, last=4
   b) start = 4, end=7, first=3, last=6
   c) start = 7, end=4, first=6, last=3
   d) start = 3, end=7, first=4, last=6
   e) None of the above

34. Consider the code shown below. What does this method return?

   String changeUp( String aWord)
   {
      char c,d;
      String newWord = "";

      for( int i=aWord.length()/2-1; i>=0; i--){
         c = aWord.charAt( i);
         d = aWord.charAt( aWord.length() - i - 1);
         newWord = d + newWord + c;
      }
      return newWord;
   }

   a) the letters from parameter aWord in reverse order
   b) the letters from parameter aWord in reverse order when the length of aWord is even
   c) the letters from parameter aWord in reverse order when the length of aWord is odd
   d) every other letter from the letters in parameter aWord
   e) None of the above
35. Consider the code shown below. What does this method do?

```
void mystery( int x)
{
    for( int i=2; i<=x; i++) {
        boolean flag = true;
        for( int j=2; j<=i/2; j++) {
            if( i%j == 0) {
                flag = false;
                break;
            }
        } 
        if( flag) {
            System.out.print( i + " ");
        }
    }
}
```

a) displays part of a multiplication table
b) displays all odd numbers up until the value of x
c) displays prime numbers up until the value of x
d) displays factors of x
e) None of the above

36. Assume an instance of class Confusing is created, and that instance is used to call method `doIt()`. What is the output?

```
class Confusing
{
    int x = 1;
    int y = 3;
    void first( int x)
    {
        int temp = x;
        y = x/2;
        x = temp;
    }//end method first
    void second( int x, int z)
    {
        first( x);
        x = y + 1;
    }//end method second
    public void doIt()
    {
        int x = 2;
        y++;
        first( y);
        second( y,x);
        System.out.print("Answer is: " + this.y);
    }//end method doIt()
}
```

a) Answer is: 3
b) Answer is: 4
c) Answer is: 5
d) Answer is: 7
e) None of the above