What is your name?: ___________________________(2 points)

There are three sections:
   I. True/False. . . . . . . . . . . . . .54 points; (27 questions, 2 points each)
   II. Multiple Choice  . . . . . . . .44 points; (11 questions, 4 points each)
-------------------
   98 points + 2 for name = 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. All code is in Java unless stated otherwise. This test is open book and open notes. You have 50 minutes.

• For the True/False questions, if the answer is True, fill in T on your answer form. If the answer is False, fill in F 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. A Constructor is special initialization code that is run when an object is created.
T F 2. One way to tell the difference between a Constructor and a method is that a Constructor has no return type.
T F 3. The name of a Constructor must be the same as the class name if it is a public constructor. If it is a private constructor, then you can change the name.
T F 4. The output of the code below is the value: 10
   ```java
   int x=2, y=4, z=2;
   System.out.println(x + y * z);
   ```
T F 5. The output of the following lines of code is: Yes
   ```java
   boolean True = false;
   if (True = true) {
       System.out.println("Yes");
   }
   else {
       System.out.println("No");
   }
   ```
T F 6. The following code is valid (compiles and runs) in Java:
   ```java
   for( ; ; );
   ```
7. Every set of `if - else - if` statements can be alternatively represented using a `switch - case` statement.  

8. The `break` statement causes a program to stop execution.  

9. 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`.  

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

11. 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.  

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

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

13. 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.  

14. A `static` method can not access non-static instance variables.  

15. A method can call itself in Java.  

16. 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.  

17. The output of the statement below is:  `2b or not 2b`

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

18. For an input of 7, the following code when compiled and run in C gives as output:  `8`

   ```c
   int x;
   scanf("%d",x);
   printf("%d", x+1);
   ```

19. The following code in Java stores in variable `sum` the sum of positive odd integers less than 10:

   ```java
   int sum = 0;
   for(int x=-1, sum=0; x<=10; x+=2)
       if( x>0) sum+=x;
   ```
20. The following code prints the words: Grade is: A today

```java
int x = 93;
System.out.print("Grade is: ");
if (100 >= x >= 90)
    System.out.print("A ");
else
    System.out.print("not A ");
System.out.print("today");
```

21. The following code prints the words: val is: B

```java
char val='A';
switch (val){
    case ("A"): val='B';
    case ("B"): val='C';
    case ("C"): val='D';
    break;
};
System.out.println("val is: " + val);
```

22. The length() method is used for both strings and arrays. You must be sure to include the parenthesis.

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

```java
int x = 3;
int y = 8;
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;
}
```

24. Assume the code shown below, where method swapValues1 is called.
Output of this segment of code is: Values are: 2 5

```java
int[] numbers = {2,5};
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;
}
T    F       25. Assume the code shown below, where method `swapValues2` is called. Output of this segment of code is: Values are: 7 2

```java
int[] numbers = {2, 7};
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;
}
```

T    F       26. In C++, if variable `number` is declared as an int, the following would compile and execute correctly:

```cpp
cin << number;
```

T    F       27. In a C++ program, the return type of `main()` should be `int`
II. Multiple Choice (5 pts. each)

28. Consider the code given below. Its output is:

```java
int x = 2;
int y = 6;
int z = x++ + y;
System.out.println("Value is: " + x + y + z);
```

a) Value is: 268  
b) Value is: 368  
c) Value is: 278  
d) Value is: 279  
e) None of the above

29. 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 theNumbers  
b) The maximum value in array theNumbers  
c) The average value in array theNumbers  
d) The sum of the values in array theNumbers  
e) None of the above

30. Assume you are doing a numbers guessing game, where you are attempting to guess a number between 40 and 300 (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
31. What is the output of the program segment at right below when an instance of class `Verify` is created and used to call method `verifyIt()`?

a) -1  
b) 0  
c) 1  
d) 2  
e) None of the above

```java
class Verify
{
    int x = 0;

    void verifyIt()
    {
        if( second() && first() )
            x++;
        System.out.println( x);
    }

    boolean first()
    {
        x--;
        return false;
    }

    boolean second()
    {
        x++;  
        return true;
    }

}//end class Check
```

32. What is the output of the code given below when `problem32Driver()` is called?

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

```java
public void problem32Driver()
{
    String phrase = "eschew surplussage";
    // create character array from String
    char[] theArray = phrase.toCharArray();

    problem32( theArray);

    // display array contents
    System.out.print( theArray);
}

public void problem32(char[] w)
{
    int x = w.length;
    char c;
    for (int i=0; i<x; i++) {
        c = w[i];
        w[i] = w[x-i-1];
        w[x-i-1] = c;
    }
}
```
33. What is the output of the code given in the two columns below when an instance of class Confuse is created and used to call method startUp()?

```
class Confuse
{
    int x=3;
    int y=6;

    private void first(int y, int x)
    {
        x += 1; y += 2;
    }

    private void second(int x, int s)
    {
        x += 3; y = 7;
    }

    private void third(int y)
    {
        setXY( y);
        x += 2; y += 3;
    }

    private void setXY( int y)
    {
        x = y;
        this.y = x-1;
        second(y, x);
    }

    private void display()
    {
        System.out.println(x + y);
    }

    public void startUp()
    {
        int x=1, y=3;
        first(x,y);
        second(x,y);
        third(5);
        display();
    }
}
``` //end class Confuse

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

34. What is the output of the following code when it is called with methodCDriver();

```
public void methodCDriver()
{
    System.out.println( methodC("All generalizations are false"));
}

public char methodC( String s)
{
    if( s.length() <=1)
        return s.charAt(0);
    else
        return methodC( s.substring(0, s.length()-1));
}
``` 

a) A  
b) e  
c) t  
d) a  
e) None of the above.
35. Assume the code shown below right is stored in a file named `problems.java`

Assume that if it ran correctly, the output would be:

```
2 4 6 8 10
eschew surplusage
End
```

How many errors can you find that would prevent this code from running correctly if it were invoked in BlueJ by calling method `runIt()`?

a) 1, 2, or 3  
b) 4 or 5  
c) 6 or 7  
d) 8 or 9  
e) None of the above

```java
class Problem
{
    int counter = 5;
    int runIt()
        {
            runOnce();
            System.out.println("End");
        }
    void runOnce(int counter)
        {
            for( int i=0; i<=counter; i++);
            System.out.println( i*2);
            System.out.println("eschew ");
            System.out.println("surplusage");
        }
}
```

36. What is the output of the following code when it is called with `methodB(2,5)`:

```java
public int methodA( int x, int n)
{
    if (n == 0)
        return 0;
    else
        return x + methodA( x, n-1);
}
public int methodB( int x, int n)
{
    if (n == 0)
        return 1;
    else if (n > 0)
        return methodA( methodB( x, n - 1), x);
    return 0;
}
```

a) 7  
b) 10  
c) 25  
d) 32  
e) None of the above
37. What is the output of the code given below when called with:

```
ClassE classEinstance = new ClassE( 3);
```

```java
class ClassD
{
    private int x;

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

    private void method1( int x)
    {
        x++;    
        ClassE instance1instance1 =
            new ClassE( x);
    }
}
} // end of classD

class ClassE
{
    private int x=2;

    public ClassE(int x)
    {
        if ( this.x < 5) {
            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.
38. The following is a maze program similar to the ones we discussed in class:

class Maze
{
    private int start = 11;
    private int end = 55;
    private int[] maze =
        /*       + 0 1 2 3 4 5 6 7 8 9 */
        /* 0 */   {1,1,1,1,1,1,1,1,1,1,
        /*10 */   1,0,0,0,0,1,1,1,1,1,
        /*20 */   1,1,1,0,1,1,1,1,1,1,
        /*30 */   1,0,0,0,0,0,1,1,1,1,
        /*40 */   1,1,0,1,1,0,1,1,1,1,
        /*50 */   1,1,0,0,0,0,1,1,1,1,
        /*60 */   1,1,1,1,1,1,1,1,1,1},
    private int[] moves = {-1,-10,1,10};
    private int[] cameFrom = new int[70];

    public Maze()
    {
        makeMove(start);
    }

    public void makeMove(int current)
    {
        System.out.print(current +", ");
        if (current == end) {
            System.out.println("Got to destination!");
            return; // found solution, we're done
        }
        for (int i=0; i<4; i++) {
            int next = current + moves[i];
            if ((maze[next] != 1) && (next != cameFrom[current]) ) {
                cameFrom[next] = current;
                makeMove(next);
            }
        }
    }
    //end makeMove
}

What is the output from running this code when an instance of class Maze is created?

a) The solution path in order
b) The solution path in reverse order
c) A bunch of numbers, but not the solution path
d) The solution path with additional other numbers
e) None of the above