1— Which of the following is not a primitive data type?
   a. int
   b. String
   c. char
   d. boolean

2— If you wanted to store pi in a data type, which of these would be best?
   a. int
   b. float
   c. char
   d. long

3— Which modifier goes before a variable when you want an entire class to share only one copy?
   a. package
   b. class
   c. static
   d. void

4— Which of the following expressions would return 8.0?
   a. 5 / 2 * 4.0
   b. 5.0 / 2 * 4
   c. 5 / 2.0 * 4
   d. 5.0 / 2.0 * 4.0

5— Which of the following represents the proper typical structure of a for loop?
   a. for (initialize counter; update counter) {
            statement;
            statement;
            . . .;
          }
   b. for (initialize counter; test counter; update counter) {
            statement;
            statement;
            . . .;
          }
   c. for (update counter; initialize counter; test counter) {
            statement;
            statement;
            . . .;
          }
   d. for (test counter; initialize counter; update counter) {
            statement;
            statement;
            . . .;
6— Consider the following code segment,

```java
int x = 3; int y = 2;
double z = (double) x / y;
```

(double) is not necessary for this code to compile; **why** is (double) used in this code? What is z with (double) in place? What is z without (double)?

7— Consider the following code segment, which will be used to determine the number of years required for a $10,000 investment to grow to $25,000 at a simple interest rate of 3%.

```java
int years = 0;
double balance = 10000;
double interest = 0;
while( balance < 25000 ) {
    years++;
    interest = balance * .03;
    <BLANK>
}
System.out.println("Years = " + years);
```

What **line of code** should be placed where <BLANK> is so that the calculation works as intended?

8— Why do the following lines of code not compile? (Explain in one sentence each)

```java
double w, x, y and z;

System.out.println(Hello again!);

if( x = y ) {
    boolean myBoolean = true;
}  ```
9— The method below should print three times the number of "*" as the argument if the number is odd, and double the number of "*" if the number is even, that is:

\[
\begin{align*}
\text{tripleDoubleStar}( 1 ) & \quad > \quad *** \\
\text{tripleDoubleStar}( 2 ) & \quad > \quad **** \\
\text{tripleDoubleStar}( 3 ) & \quad > \quad ********* \\
\text{tripleDoubleStar}( 4 ) & \quad > \quad *******
\end{align*}
\]

However! It has three errors. Correct them!

```java
void tripleDoubleStar( int stars ) {
    if( stars % 2 == 1 ) {
        for( int i=0; i<stars; i++ ) {
            System.Out.Print( "***" );
        }
    } else if( stars % 2 == 2 ) {
        for( int i=0; i<stars.length; i++ ) {
            System.out.print( "**" );
        }
    }
}
```

10— Write a method that returns the nth element of the Fibonacci sequence (n should be an input)

The Fibonacci sequence as defined as:

- the first element is 0
- the second element is 1
- all other elements are the addition of the two previous elements.

That is:

\[
\begin{align*}
0 & \quad 1 & \quad 1 & \quad 2 & \quad 3 & \quad 5 & \quad 8 & \quad 13 & \quad 21 & \quad 34 & \quad 55 & \quad 89 & \quad 144 & \quad ...
\end{align*}
\]

That is, if your method was given the input 7, your method should return 13.
11—The array question.

Return the sum of the numbers in the array, returning 0 for an empty array. **Except** the number 13 is very unlucky, so it does not count towards the sum and any number immediately following 13 also does not count towards the sum.

\[
\begin{align*}
\{1, 2, 2, 1\} & > 6 \\
\{1, 1\} & > 2 \\
\{1, 2, 2, 1, 13\} & > 6 \\
\{1, 1, 1, 1, 13, 10\} & > 4
\end{align*}
\]

```java
public int sum13(int[] nums) {
    int sum = 0;
    for (int i = 0; i < nums.length; i++) {
        if (i == 0 && nums[i] != 13) {
            sum += nums[i];
        } else if (i > 0 && nums[i] != 13 && nums[i - 1] != 13) {
            sum += nums[i];
        }
    }
    return sum;
}
```

```java
public int sum13(int... list) {
    boolean bad = false;
    int sum = 0;
    for (int i : list) {
        if (i == 13) {
            bad = true;
        } else if (bad) {
            bad = false;
        } else {
            sum += i;
        }
    }
    return sum;
}
```
12— Describe two differences between superclasses and interfaces

13— The inheritance question.

Given the interface Animal, make the necessary classes/methods/objects to make the main method below correctly give the output (also described below).

```java
public interface Animal {
    /**
     * Animals must be able to make sounds
     *
     * @return String - the sound an animal makes
     */
    public String makeSound();
}

public static void main(String[] args) {
    Animal myAnimal = new Dog();
    System.out.println( myAnimal.makeSound() );
    myAnimal = new Cat("oscar");
    System.out.println( myAnimal.makeSound() );
    Cat myCat = new Cat("fluffy");
    System.out.println( myCat.makeSound() );
    System.out.println( myCat.purr() );
    System.out.println( myCat.introduceSelf() );
}

Output:
> woof
> meow
> meow
> meow
> purr
> My name is fluffy.
```