

*Note: all of the information below I'm either presenting from having learned it in Promineo Tech's course materials for the current week, or simply from my experiences writing similarly structured loops in Javascript. There is one example I borrowed from my course instructor, which I identify as such when using it.*

**1. What are the different loops Java provides us with?**

- A. The "for loop"
- B. The "while loop"
- C. The "do... while" loop

**2. What is the syntax of one of the loops in your above answer? How and when should you use this loop? Give an example.**

The "for loop" takes the following syntax:

```
for (index declaration; test expression; index iteration) {  
    ... some code;  
}
```

...where the index declaration is usually initialized to an int, the test expression involves the declared index variable being greater than or less than some number or stored value (meaning the code in the loop will execute as long as that test expression remains true), and the index iteration uses an assignment operator or assignment operator shorthand, usually to increment/decrement the value of the index variable. In practice, a "for loop" usually looks like this:

```
for (int i = 0; i <= someNumber; i++){  
    System.out.println("Iteration number " + i);  
}
```

Notice how I made the test expression use another stored value, "someNumber", presumably declared elsewhere before the code above. "For loops" are useful in situations where the total number of iterations is unknown to the programmer, or will likely be different between executions of the code, and can be stored in a variable whose value is assigned from some user input or other external data. An example given by my instructor Chuck Kefrieter during our last class was the size of a user's shopping cart: if we're using a for loop to add up the total cost of all items in a shopping cart, we would presumably set the test expression part of the for loop syntax to execute the code up until it has gone through the total number of items in the shopping cart, which is a value that is likely to change from execution to execution.

**3. What is the syntax of another one of the loops in your above answer? How and when should you use this loop? Give an example.**

The "while loop" takes the following syntax:

```
while (test expression) {  
    ... some code;  
    [some sort of variable reassignment];  
}
```

...where the code block after the while statement will continually execute as long as the test expression remains true. The test expression *may* test an iterator variable that will be reassigned (i.e. like an int value is incremented or decremented in a "for loop") during the execution of the code block, but may also depend on more complex operations and/or access of external data. Here are two examples:

```
int targetValue = 5;  
int iterator = 0;  
  
while (iterator < targetValue) {  
    System.out.println("We are not yet at our target value.");  
    iterator++;  
}
```

Notice that--for this particular example--in order for the iterator variable to actually reach the target value, the code block will have to execute the same number of times as we have assigned to the variable "targetValue". In a "for loop", it would only execute up to `targetValue - 1` times. This difference in the two looping mechanisms illuminates their use cases: often, a programmer using a "while loop" does so because they are aware of the explicit number of iterations that will occur.

While loops can also lend themselves to more complex situations. Take the following code as an example (assume that the methods `getTemperature()` and `getHumidity()` are declared elsewhere, and access some external data):

```
boolean yuckyDay = false;  
while (yuckyDay == false) {  
    double temp = getTemperature();  
    double humid = getHumidity();  
  
    boolean hotAndHumid = temp >= 80.0 && humid >= 75.0;  
    boolean isRaining = humid >= 98.0;
```

```

        if (hotAndHumid || isRaining) {
            yuckyDay = true;
        }
    }
}

```

I'll forego a lot of explanation and let the code speak for itself, with the exception of pointing out that our test expression and the following code block don't utilize iteration, and will continue to execute until the day is yucky, or forever, if the previously-declared data access methods fail (like just returning the initial value they stored on all subsequent calls, or something like that).

#### 4. What is the syntax of another one of the loops in your above answer? How and when should you use this loop? Give an example.

The "do-while loop" functions much like the "while loop", with the exception that its test expression comes *after* its code block. The practical consequence is that the code block always executes at least once before the test expression is evaluated. It looks like this:

```

do {
    ... some code;
} while (test expression);

```

It can be useful in situations where we want to continue executing the code block one more time after the test expression evaluates to false, like so:

```

boolean yuckyDay = true;

do {
    double temp = getTemperature();
    double humid = getHumidity();

    boolean hotAndHumid = (temp >= 80.0 && humid >= 75.0);
    boolean isRaining = (humid >= 98.0);

    if (!(hotAndHumid || isRaining)) {
        yuckyDay = false;
        System.out.println("Time to go outside!");
    } else {
        System.out.println("Waiting for a nice day...");
    }
} while (yuckyDay == true);

```

...here, the programmer wants to display a message to the user (or, in

general, execute some kind of code) just once after the test expression has evaluated to false. This kind of situation is a potential use case.

## **5. What is your favorite thing you learned this week?**

The brief introduction to reading and working with user input was especially interesting. I'm accustomed to using Javascript to access the DOM, usually getting the value of an HTML `<input>` element if I need some information from the user. It's cool to see how Java handles this with "System.in" and what the basics of that code might look like. I Definitely need to know more about this for the end goal of building an API.

Also, I learned that next week's research assignment will be in markdown, because I want syntax highlighting in my code blocks.