

## Model I. (25 min) Arrays

Consider the following series of Java statements that initialize the first several Fibonacci numbers. (The Fibonacci numbers represent a series 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so forth.)

```
// Code Example #1
int a1 = 0;
int a2 = 1;
int a3 = 1;
int a4 = 2;
int a5 = 3;

// Code Example #2

// declaration
int[] numbers = new int[5];

// initialization
numbers[0] = 0;
numbers[1] = 1;
numbers[2] = 1;
numbers[3] = 2;
numbers[4] = 3;
```

### Critical Thinking Questions

1. What is the data type of the variables beginning with letter 'a'?  
**integer**
2. How many variables beginning with letter 'a' have been created?  
**5**
3. If we wanted to represent the first 25 numbers in the Fibonacci sequence, how many variables would this require?  
**25**
4. What is the data type of the variable 'numbers'?  
**array**
5. What is the value of 'numbers[2]'?  
**1**
6. What is the value of 'numbers[4]'?  
**3**

7. When we declare the data type of a variable with the '[' notation, we are indicating the variable is an *array*. What variable in Model I is an array?

**numbers**

8. What is the size of the array that is declared in Model I?

**5 numbers (0-4)**

9. The number between the '[' and ']' refers to the *index* of the array. What index does an array begin with?

**0**

10. Does the size of the array affect the starting index of the array?

**No**

11. What is the ending index of the array in Model I?

**3**

12. What is the ending index of an array of size 10?

**34**

13. Does the size of the array affect the ending index of the array?

**Yes**

14. What changes to the declaration in Code Example #2 would be required if we wanted to store the first 25 numbers in the Fibonacci sequence in numbers?

**int[] numbers = new int[25];**

15. Declare an array of size 50 with the name 'averages' of type double.

**double[] averages = new double [50];**

16. Compare Code Example #1 and Code Example #2 in Model I. What is the primary benefit of using arrays?

**You don't have to declare each individual value.**

17. What would be output from the following statements?

```
int i = 0;
System.out.println(numbers[i]);
i++;
System.out.println(numbers[i]);
```

**0**

**1**

18. Using either a while or do-while loop, develop pseudocode that would print out each of the values in an array named `values` of size 50.

```

public class POGIL6
{
    public static void main(String[] args)
    {
        int i = 0;
        int[] values = new int[50];

        while (i < 50)
        {
            values[i] = i;
            System.out.println(values[i]);
            i++;
        }
    }
}

```

## Model II. (20 min) For Loops

The **while** and **do-while** loops are used to loop as long as a Boolean condition is true. Both while and do-while loops are intended to be written such that the body of the loop ultimately causes the Boolean condition to become false.

**For** loops are appropriate when you know exactly how many times you want the body of the loop to run.

```

public class LoopExample {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            System.out.println("i = " + i);
        }

        System.out.println("goodbye");
    }
}

```

### Critical Thinking Questions

1. Identify the statement(s) that make up the **body** of the for loop.  
*i gets 1 added to it till i is equal to 5 and then it prints goodbye.*
2. If you were to run this program, what do you expect would be output?  
*i=0,i=1,i=2,i=3,i=4,goodbye*
3. Copy and paste this code example into BlueJ. Compile and run it. Does your expected output match the actual output?  
*yes*

For loops have three pieces: (1) initialization, (2) condition, and (3) increment.

4. What Java code is the **initialization**?

```
int i = 0
```

5. What Java code is the **condition**?

```
for (int i = 0; i < 5; i++)
```

6. What Java code is the **increment**?

```
System.out.println("i = " + i);  
}
```

```
System.out.println("goodbye");
```

7. Explain the logical error with the following for loop:

```
for (int i = 0; i > 5; i++) {  
    System.out.println("i = " + i);  
}
```

**"i" is air**

8. Complete the following statement: "A for loop continues to execute as long as the condition is **true**".

The following questions can best be answered using BlueJ

9. What would be output from the following for loop?

```
for (int i = 5; i <= 25; i += 5) {  
    System.out.println("i = " + i);  
}
```

10. Write a for loop that counts **up** from 1 to 100.

```
public class LoopExample {  
public static void main(String[] args) {  
    for (int i = 1; i <= 100; i++) {  
        System.out.println("i = " + i);  
    }  
  
    System.out.println("goodbye");  
}  
}
```

11. Write a for loop that counts **down** from 100 to 1.

```
public static void main(String[] args) {  
    for (int i = 100; i>0; i--)  
        System.out.println(i);  
}
```

12. Explain the logical error with the following for loop:

```
int j = 0;  
  
for (int i = 0; i < 5; j++) {  
    System.out.println("i = " + i);  
}
```

It will always be 0, so it continuously loops and prints "i=0"

### ***Model III. (20 min) Arrays and For Loops***

For loops are often used in coordination with arrays.

```
// declaration and initialization
int[] numbers = {0, 1, 1, 2, 3};

for (int i = 0; i < numbers.length; i++) {
    System.out.println("value = " + numbers[i]);
}
```

#### **Critical Thinking Questions**

1. Model I illustrates an alternative technique for initializing an array. What are the values of the following elements in the array `numbers`?

`numbers[0]` =0

`numbers[1]` =1

`numbers[2]` =1

`numbers[3]` =2

`numbers[4]` =3

2. The `length` property of an array determines its size (capacity). What does the value `numbers.length` evaluate to?

5

3. Assume there exists an array `averages` where the value `averages.length` is equal to 50. What is the starting index of the `averages` array? What is the ending index of the `averages` array?

0 is the start, 49 is the end

4. Assume we have declared the following array

```
int[] sequence = new int[50];
```

and initialized the first few elements in the array as follows:

```
sequence[0] = 0;
sequence[1] = 1;
```

```
sequence[2] = 1;
```

What assumptions can we make - if any - about array locations `sequence[3]` through `sequence[49]`?

**Sequence[3] would come next and sequence[49] would be the end.**

5. What would be the resulting value of `sequence[3]` after the following statement?

```
sequence[3] = sequence[2] + sequence[1]; sequence[3] = 2;
```

6. Write a for loop that prints out all of the elements in an array named `distances`:

```
int[] distances = {0, 1, 1, 2, 3};

for (int i = 0; i < numbers.length; i++) {
    System.out.println("value = " + numbers[i]);
}
```

7. Write a for loop that prints out each of the odd elements (the elements at positions 1, 3, 5, 7, 9, and so forth) in an array named `distances`:

```
for (int i = 0; i < numbers.length; i++) {

    if (i%2 != 0)
    {
        System.out.println("value = " + numbers[i]);
    }
}
```

8. Using BlueJ, create an array of integers of size 100 and initialize each element to -1. (Proceed carefully with this - there is a very easy way of accomplishing this, and a not-so-easy way!)

```
int value = -1;
int [] integers = new int [100];
integers [value] = value;
```