NAME: Sedina Husanovic

|  |  |
| --- | --- |
| Model I. 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’? int

1. How many variables beginning with letter ‘a’ have been created? 5

1. If we wanted to represent the first 25 numbers in the Fibonacci sequence, how many variables would this require? 25(code ex 1)

1. What is the data type of the variable ‘numbers’? int

1. What is the value of ‘numbers[2]’? 1

1. What is the value of ‘numbers[4]’? 3

1. 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?

Code example #2 numbers

1. What is the size of the array that is declared in Model I? 5

1. The number between the ‘[‘ and ‘]’ (after the array name) refers to the *index* of the array. What index does an array begin with? 0

1. Does the size of the array affect the starting index of the array? no

1. What is the ending index of the array in Model I? 4

1. What is the ending index of an array of size 10? 9

1. Does the size of the array affect the ending index of the array? yes

1. 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];

1. Declare an array of size 50 with the name ‘averages’ of type double.

double[] averages = new double[50];

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

It is used to represent multiple data items of same type by using only a single name/variable.

1. What would be output from the following statements?

Whatever the value in index 0(zero); and the index value will increase by one; so the output is 1  
  
int i = 0;  
System.out.println(numbers[i]);  
i++;  
System.out.println(numbers[i]);

1. 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.

int i = 0;

while(i < 50)

{

System.out.println(value[i]);

i= i + 1;

}

*LE STOP}}*

|  |  |
| --- | --- |
| Model II. 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 II 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

1. The length property of an array determines its size (capacity). What does the value numbers.length evaluate to? 5

1. 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? Starting = 0

Ending = 49

1. 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;  
     
   Can you guess the values for sequence[3] through sequence[49]? Yes

long[] sequence = new long[50];

sequence[0] = 0L;

sequence[1] = 1L;

sequence[2] = 1L;

for(int i = 3; i < sequence.length; i++)

sequence[i] = sequence[i-1] + sequence[i-2];

for(int i =0; i < sequence.length; i++)

System.out.println(sequence[i]);

1. What would be the resulting value of sequence[3] after the following statement?  
     
   sequence[3] = sequence[2] + sequence[1];

The result is sequence[2] = 1 + sequence[1] = 1

1 + 1 = sequence[3] which is 2

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

for(int i = 0; distances.length; i++)

{

System.out.println(distances[i]);

}

1. 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 in an array named distances:

for(int i = 1; distances.length; i = i +2)

{

System.out.println(distances[i]);

}

1. Using Dr. Java, 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[] arrayInt = new int[100];

for(int i =0; i < arrayInt.length; i++)

{

arrayInt[i] = -1

}