**PROJECT ONE ( Using Arrays in Java )**

**Objective** To create basic arrays in Java.

***PROJECT DESCRIPTION***

This project has you constructing an array process in Java which declares and initializes an integer array and prints each value to standard output. Create your Java program using the file name ArrayDemo.

***Information About this Project***

Arrays are useful to study in computer science since they can be manipulated to show processing times and process efficiency.

***Steps to Complete this Project***

**STEP 1 Run a Project in a Java IDE**

Run the given code below and take a screen snapshot of the correct output display of the code.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**class ArrayDemo {**

**public static void main(String[] args) {**

**int[] anArray; // declares an array of integers**

**anArray = new int[10]; // allocates memory for 10 integers**

**anArray[0] = 100; // initialize first element**

**anArray[1] = 200; // initialize second element**

**anArray[2] = 300; // etc.**

**anArray[3] = 400;**

**anArray[4] = 500;**

**anArray[5] = 600;**

**anArray[6] = 700;**

**anArray[7] = 800;**

**anArray[8] = 900;**

**anArray[9] = 1000;**

**System.*out*.println("Element at index 0: " + anArray[0]);**

**System.*out*.println("Element at index 1: " + anArray[1]);**

**System.*out*.println("Element at index 2: " + anArray[2]);**

**System.*out*.println("Element at index 3: " + anArray[3]);**

**System.*out*.println("Element at index 4: " + anArray[4]);**

**System.*out*.println("Element at index 5: " + anArray[5]);**

**System.*out*.println("Element at index 6: " + anArray[6]);**

**System.*out*.println("Element at index 7: " + anArray[7]);**

**System.*out*.println("Element at index 8: " + anArray[8]);**

**System.*out*.println("Element at index 9: " + anArray[9]);**

**}**

**}**

**PROJECT TWO ( Array Cloning in Java )**

**Objective** To utilize the arraycopy method of java

***PROJECT DESCRIPTION***

Modify your code in Project One to demonstrate how to clone or copy a Java array.

Starter code for this project will follow.

***Information About this Project***

Arrays are useful to study in computer science since they can be manipulated to show processing times and process efficiency.

***Steps to Complete this Project***

**STEP 1 Run a Project in a Java IDE**

The Java **System** class has an **arraycopy** method that you can use to efficiently copy data from one array into another:

**public static void arraycopy(Object src,**

**int srcPos,**

**Object dest,**

**int destPos,**

**int length)**

The two Object arguments specify the array to copy *from* and the array to copy *to* . The three **int** arguments specify the starting position in the source array, the starting position in the destination array, and the number of array elements to copy.

Use the following code snippet declaring an array of **char** elements, spelling the word " decaffeinated " . Note the code uses **arraycopy** to copy a subsequence of array components into a second array:

**public static void main(String[] args) {**

**char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',**

**'i', 'n', 'a', 't', 'e', 'd' };**

**char[] copyTo = new char[7];**

**System.*arraycopy*(copyFrom, 2, copyTo, 0, 7);**

**System.*out*.println(new String(copyTo) + copyTo[4]);**

Run your program and test your output, which spells the word **caffeine**.   
 Submit your results.

Next modify your code once again to display the array before the copy and again after the copy. Submit your results once again.

**PROJECT THREE ( Working Arrays in Assembler )**

**Objective** To create arrays in Assembler

***PROJECT DESCRIPTION***

For this project, you will create an array similar to **Project 1** and turn it into assembler program. Namely, declare a table to be of size 5 for example and populate each index with a value of 100 .

***Information About this Project***

Arrays are useful to study in computer science since they can be manipulated to show processing times and process efficiency.

***Steps to Complete this Project***

**STEP 1 Run a Project in MIPs**

Submit your ASM file and a snapshot of your program in action.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROJECT FOUR ( Questions and Answers )**

**Objective** To reflect upon the objectives and goals of this project.

***PROJECT DESCRIPTION***

Answer each of the following exercises, which are based on the theory and concepts of this computer laboratory assignment.

***Information About this Project***

It is always an excellent idea to overview any computer laboratory assignment or project

once it is completed.

***Steps to Complete this Project***

**STEP 1 Questions and Answers Concerning this Computer Laboratory Project**

Open MS Word and, within a new document, place your responses to these questions. Submit your completed MS Word document for credit.

**(1) ( Review Topics: Arrays in Java )**

Arrays are important for use in all computer languages. The following code segment using an array, manipulate the array or perform operations on the array and then to display information concerning the array.

Examine this code segment and explain what exactly will be accomplished when the statements are executed.

. **PROJECT FOUR ( Questions and Answers )**

|  |
| --- |
| **String[] words = {"computer", "storage drive", "software"};**  **int maxLength = 0;**  **String longest = null;**  **int size = words.length;**  **for (int i = 0; i < size; i++)**  **{**  **System.*out*.println(words[i]);**  **}**  **for (String s : words)**  **{**  **if (s.length() > maxLength)**  **{**  **maxLength = s.length();**  **longest = s;**  **}**  **}**  **System.*out*.format("longest string: '%s'\n", longest);** |

**(2) ( ArrayCopy in Java )**

Briefly summarize the function and / or purpose of the following program.

. **PROJECT FOUR ( Questions and Answers )**

|  |
| --- |
| **import java.lang.\*;**  **public class ArrayDemo**  **{**  **public static void main(String[] args)**  **{**  **char[] anArray1;   // declares an array of characters - original array**  **anArray1 = new char[7];**  **// allocates memory for 7 characters**    **anArray1[0] = 'h'; // initialize first element**  **anArray1[1] = 'i'; // initialize second element**  **anArray1[2] = 's'; // etc.**  **anArray1[3] = 't';**  **anArray1[4] = 'o';**  **anArray1[5] = 'r';**  **anArray1[6] = 'y';**      **System.*out*.println("\nArray before copying:\n");**  **for (int i = 0; i < anArray1.length; i++)**  **{**  **System.*out*.println("Element at index " + i + ": " + anArray1[i]);**  **}**    **char[] anArray2;**    **anArray2 = new char[5];**    **System.*arraycopy*(anArray1, 2, anArray2, 0, 5);**    **System.*out*.println("\nArray after copying:\n");**    **for (int i = 0; i < anArray2.length; i++)**  **{**  **System.*out*.println("Element at index " + i + ": " + anArray2[i]);**  **}**  **}**  **}** |

**(3) ( ArrayCopy in Java )**

Briefly summarize the necessary arguments **arraycopy()** method.

. **PROJECT FOUR ( Questions and Answers )**

**(4) ( Arrays in MIPS )**

Java uses square brackets [ ] for array indices. Does MIPS assembly language also use square brackets for arrays?

**(5) ( Arrays in MIPS )**

Summarize what is accomplished with this MIPS assembly program segment.

**loop:**

**sw $t3, arr($t1) # arr[i] = 100**

**addi $t0, $t0, 1 # increment loop counter**

**addi $t1, $t1, 4 # increment index by 4 bytes**

**blt $t0, $t2, loop # while i < 5**