Lab 2

Contents:

Working with arrays - creating and initializing arrays, accessing arrays, enhanced for loop/ foreach loop, if statements, operators in Java.

Background:

Following is a side-by-side comparison of C and Java syntax.

Task	C Programming Language	Java Programming Language
Create an array and initialize it	`double array[5] = {2.0, 3.0, 7.0,`	`double[] array = {100.0, 2.0, 3.0, 7.0, 50.0};`
Accessing arrays	`printf("%d\n", array[1]);`	`System.out.println(array[1]);`

Array Traversal	`for (j = 0; j < 10; j++ ) { printf("Element[%d] = %lf\n", j, array[j] ); }`	`for (int j = 0; j < 10; j++ ) { System.out.println(array[j]); }`
		Using foreach loop,
		`for (double arrItem : array){ System.out.println(arrItem ); }`

The use of If statements in Java is similar to that of C.

Program:

```
public class GroceryListExample {
   public static void main(String[] args) {
      // Declare variables
      int numItems = 4;
      String[] itemNames = {"apples", "bananas", "bread",
      "milk"}; int[] itemQuantities = {3, 2, 1, 2};
      float[] itemPrices = {1.99f, 0.99f, 2.49f, 2.39f};
      boolean[] itemOnSale = {false, false, true, true};

      // Print the grocery list header
      System.out.println("Grocery List:");
      // Print each item and its information using for

      loop for (int i = 0; i < numItems; i++) {

         System.out.print("- " + itemNames[i] + ": ");
```

```
            System.out.print(itemQuantities[i] + " x ");
            System.out.print("$" + itemPrices[i] + " = ");
            double itemTotal = itemQuantities[i] * itemPrices[i];
            System.out.print("$" + itemTotal);
            if (itemOnSale[i]) {
               System.out.print(" (on sale!)");
            }
            System.out.println();
        }
    }
}
```

Exercises:

1. Rewrite the above program to use foreach loop instead of for loop.

2. Rewrite the program to include the following operations.

    a) Adding new item: add two new items at the end of the arrays by programmatically accessing the arrays.
    b) Inserting new item: insert an item to the *n*th position of the array, where *n* is a variable denoting the insertion point.
    c) Searching an item: given a variable named *key* denoting the item to be searched, traverse the array with a foreach loop and print the the item if found. Otherwise, print "Could not find the item" message.

References:

1. Java: The Complete Reference, Herbert Schildt, McGraw Hill
2. JDK Download Link - https://www.oracle.com/java/technologies/downloads/
3. Eclipse Download Link - https://www.eclipse.org/downloads/