# Software Installation

Download and install Java and Eclipse IDE

1. [JDK 7](http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. [Eclipse Kelper IDE for Java Developer](http://www.eclipse.org/downloads)

# Creating the Project

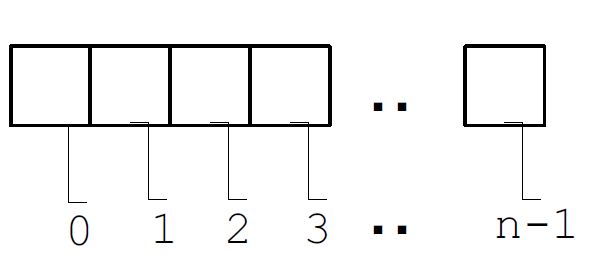
1. Click **File** 🡪 **New** 🡪 **Java Project**. In the Project Name, key in **P01**. Click [Finish].
2. If you have drop the file wrongly in P01, you can still drag the file **Helper.java** to the **src** folder.

# Array Models

In this section, you will learn about **different models of arrays.**

Synopsis

An array is an indexed list of values. You can make an array of any type int, double, String, etc.. All elements of an array must have the same type.



An array is defined using **TYPE[].**

int[] value; int[] is a type.

*Example*: int[] numbers = {1, 3, 5, 7, 9} ;

Curly braces can be used to initialize an array. It can ONLY be used when you declare the variable.

The elements of an array will be stored as shown below.

Elements in an array

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 5 | 7 | 9 |

Index numbers of array elements

# Variables and Arrays

**CCSU – Declaration of a Variable**

<http://chortle.ccsu.edu/java5/Notes/chap46/ch46_1.html>

**CCSU – Data Types**

<http://chortle.ccsu.edu/java5/Notes/chap08/ch08_2.html>

**CCSU – Names for Variables**

<http://chortle.ccsu.edu/java5/Notes/chap09A/ch09_6.html>

**CCSU – Arrays**

http://chortle.ccsu.edu/java5/Notes/chap46/ch46\_1.html

1. Click **File** 🡪 **New** 🡪 **Class**. In the Name, key in **ArrayDemo**. Click [Finish]. Key in the program as follows:

public class ArrayModel {

public static int sum = 0; // Declare a variable sum of type integer and

//initialize it to zero.

public static void main(String[] args) {

/\* program execution begins here \*/

int[] number = {1, 3, 5, 7, 9}; //Declare an array of integers

//Intialize the array

// for loop iterates 5 times

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

sum += number[i];

System.out.println("Iteration i = " + i + "\n Value =" + number[i]);

System.out.println("Sum = " + sum);

} //end of for loop

System.out.println("Sum of the numbers = " + sum);

} // end of method main

} //end of class Array Demo

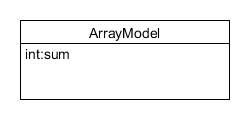
1. Click **Run** 🡪 **Run**. The program outputs appears in the Console window. Answer the following questions related to this program.
   1. What are the variables in the program ?

Sum

I

number

* 1. Write the Class Diagram for this program ?



* 1. Write down the inputs to this program ?

Sum

* 1. Write down the output of this program ?

Value of i and sum

**Iteration i = 0**

**Value =1**

**Sum = 1**

**Iteration i = 1**

**Value =3**

**Sum = 4**

**Iteration i = 2**

**Value =5**

**Sum = 9**

**Iteration i = 3**

**Value =7**

**Sum = 16**

**Iteration i = 4**

**Value =9**

**Sum = 25**

**Sum of the numbers = 25**

* 1. What is the method that helps to print the value on your screen ?

System.out.println

* 1. How many classes and methods are there ? Name them

No of classes 🡪 \_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_

Name of the class 🡪 \_\_\_\_\_ArrayModel\_\_\_\_\_\_\_\_\_\_

No of Methods 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of the method 🡪\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Write down the declaration of an array of integers in this program?

int[] number = {1, 3, 5, 7, 9};

# Array List Models

1. Create a class called **ArrayListModel** and key in the following codes.

import java.util.ArrayList; //import the ArrayList methods from the java library

public class ArrayListModel {

/\* program execution begins here \*/

public static void main(String[] args) {

int sum = 0; // Declare a variable sum which is of type integer and

//initialize it to zero.

ArrayList<Integer> number = new ArrayList<Integer>();

//Declare an array list of integers and Intialize the array

number.add(1);

number.add(3);

number.add(5);

number.add(7);

number.add(9);

// Enhanced for loop iterates 5 times and stops automatically

// for each integer i in the array “number”

for (int i: number) {

sum += i;

System.out.println("Value of i = " + i);

System.out.println("Sum = " + sum);

} //end of for loop

System.out.println("Sum of the numbers = " + sum);

} // end of method main

} //end of class Array Demo

1. Run the program and provide the output obtained.

Value of i = 1

Sum = 1

Value of i = 3

Sum = 4

Value of i = 5

Sum = 9

Value of i = 7

Sum = 16

Value of i = 9

Sum = 25

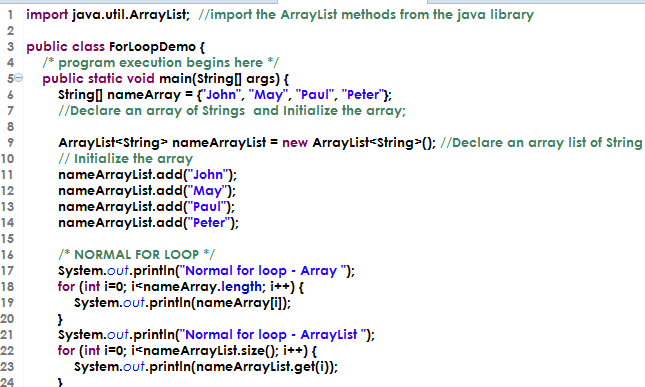
Sum of the numbers = 25

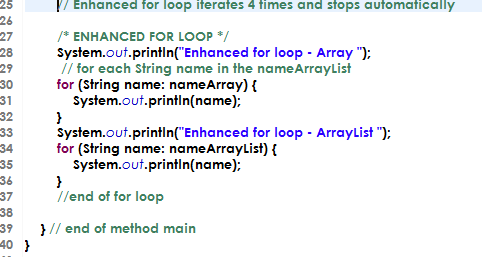
**Enhanced for loop**

<http://www.java.tips.org/java-se-tips/java.lang/the-enhanced-for-loop.html>

# Loop

1. Open the [ForLoopDemo.java](file:///D:\C306%20Data%20Structures%20and%20Algorithms\Problems\Draft\Problem%2001%20-%20Fashion%20Outlets%20-%20Arrays\Staff\p01\student\src\ForLoopDemo.java) file in Eclipse and run the program.





1. Examine the similarities and differences in using the *basic* for loop to print the values in an array and an Arraylist.

|  |  |  |
| --- | --- | --- |
| **Basic for Loop** | Array | ArrayList |
| Similarities | Can only store one data | Can only store one data |
| Differences | .Length  Fixed [i] | .size  dynamic (get)  Method |

1. Compare the *enhanced* for loop with the *basic* for loop. What are the differences in both approaches?

|  |  |  |
| --- | --- | --- |
| **Enhanced for Loop** | Array | ArrayList |
| **Similarities** | Length or size are not used | Length or size are not used |
| **Differences** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. What are the pros and cons of using the *basic* and *enhanced* for loops?

|  |  |  |
| --- | --- | --- |
| **Category** | Basic for loop | Enhanced for loop |
| Pros | Can choose the start and end position  The incrementing value can be varied in a simple for loop  Order of the elements can be varied | Simple, easy  Don’t need to specify anything (counter, conditions,etc) |
| Cons | Need to specify size, conditions,etc | Traverse in one direction .Cannot reverse the values |

# Analysing the Problem

In this section we will analyse the solution to the problem statement. We will first identify the Inputs to the program, process the inputs and then provide the result.

# Input Data

1. What are inputs that the Stock Inventory program require from the user in order perform the required calculations ?

Input is the inventory data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Shirt** | **Blouse** | **Sock** |
| **Outlet1** | 2 | 6 | 0 |
| **Outlet2** | 4 | 3 | 2 |
| **Outlet3** | 0 | 1 | 2 |
| **Outlet4** | 10 | 20 | 5 |
| **Outlet5** | 50 | 40 | 0 |

Each outlet has 3 items

* 1. Process the Data

Each store has 5 outlets

Structure the data obtained from the inventory.

1. Which approach did your team choose to use ? Array or ArrayList?

|  |
| --- |
| ArrayList |

1. Build the class and Identify the members in each class. Provide the data type and variable name for each member in the class.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * **Main class -> (StockInventory class)** * **Outlet class**  |  | | --- | | Outlet name  Data type: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Variable name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  * **Stock class**  |  | | --- | | Stock code - shirt/Blouse/socks  Data type: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Variable name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Stock count - shirt/Blouse/socks  Data type: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Variable name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |      |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Shirt** | **Blouse** | **Sock** | | **Outlet1** | 2 | 6 | 0 | |

1. Build the class diagram.

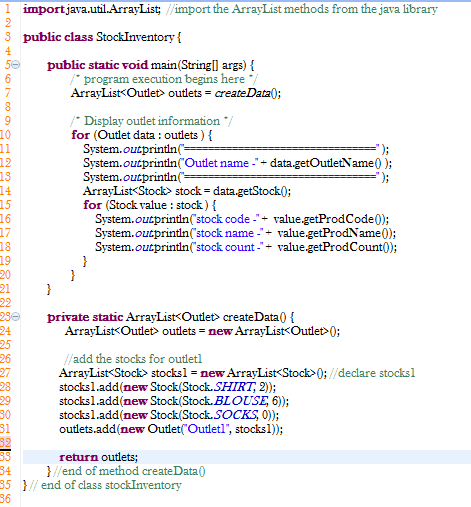
**Class Diagram**

1. Sum the inventory data in an Array / ArrayList of outlet1 to obtain the number of items in outlet1.
2. Sum the inventory data in an Array / ArrayList of each outlet and then sum the values obtained from each outlet to obtain the total number of items from all outlets
3. Find the outlet that has no stock for socks from the ArrayList and Count the number of outlets that has no stock.
4. Find the outlet that has no stock for any one item from the ArrayList and Count the number of outlets that has no stock.

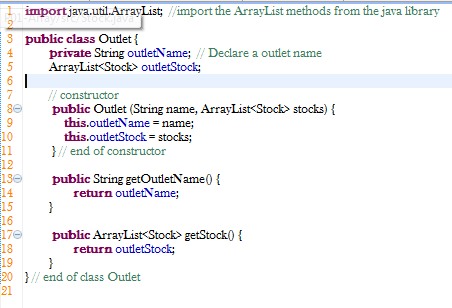
# Process the data

1. Create the three files StockInventory.java, Outlet.java and Stock.java and copy and analyse the code and explain how the code is structured ?

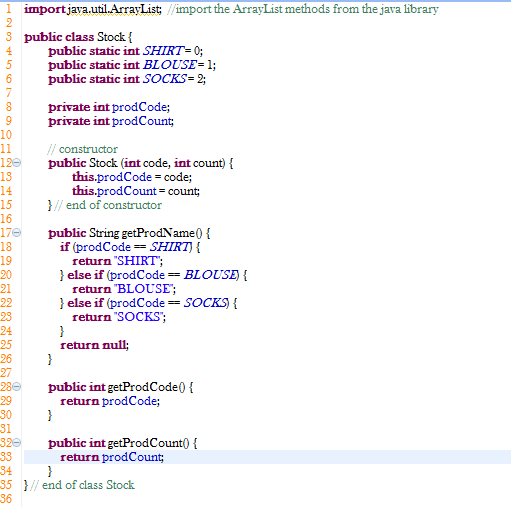
StockInventory.java



Outlet.java



Stock.java



# Display Results

1. Add five outlets and inventory information. (Note: print the outlets ArrayList and verify the data )
2. Display inventory for each outlet.
3. Total number of items for all outlets.
4. Total number of outlets and outlet’s that has no stock for socks.
5. Total number of outlets and outlet’s that has no stock for any one item.

## Advanced Challenges

1. List down the outlet which has less than 10 items in stock, for all the items.
2. List down the items that is least stocked, across all the outlets and the number of the least item in the stock ?

*End of Worksheet*