**Object Oriented Development**

Module 4: Loops

**This document includes the answers to the exercises**

© FDM Group Ltd 2020. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 15 / 10 / 20 | Nick Lawton | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## **Please note:**

Be careful about looking at the solutions too quickly; make sure you’ve given yourself time to wrestle with the concepts you just learned before looking at a solution. Also, there are several ways to solve many of the exercises, and the solutions only show one possible way to complete each exercise.

# Section 1 – For Each loops

## Array of courses

Create an array of Strings called courses which contains the following items: “UNIX”, ”SQL”, ”Java”, ”Python”, ”Web Apps”. Use a for each loop to display all of the courses. Modify the loop to display a sentence instead of just the course name. e.g. “I love learning UNIX”

“I love learning SQL”

## Copying to an ArrayList

Create an ArrayList of type String called coursesList. Use a for each loop to copy all the items from the courses array you created in the previous question into the ArrayList.

## Displaying courses

Use a for each loop to print out all of the courses stored in the ArrayList you created in question the previous question.

## Total price

Create an ArrayList of type Double called prices. Add the following prices into the ArrayList: 4.50, 1.99, 9.75, 3.15, 10.25. Use a for each loop to add up all the prices.

Display the total price.

## Copying to an array

Create an empty double array with 5 elements. Use a for each loop to copy the prices from the ArrayList you made in the previous question into the array.

## Adding numbers from a String

Create a String containing the following text: “10,7,25,3,8”.

Use the String’s split() method to create a String array containing the numbers from the String.

Now use a “for each” loop to add up the numbers in the array and display the total.

## Reversing a String

Create a String containing the following text: “desrever neeb sah txet siht”.

Use the String’s toCharArray() method to create an array of chars containing each of the letters from the String.

Finally use a “for each” loop to create a new String which contains all the letters of the original String in reverse order.

Display the new String.

# Section 2 – For Loops

## 2.1 One million

Use a for loop to display a list of all the numbers from 1 to a million.

## 2.2 Summing a million

Use a for loop to add up all the numbers from 1 to a million.

The answer should be 500000500000. Think about which type of variable would be suitable for storing a number of this size.

## 2.3 Odd numbers

Use a for loop to display a list of all the odd numbers from from 1 to 20.

## 2.4 Three times table

Use a for loop to display the three times table from 3 to 36 in the following format:

1 x 3 = 3

2 x 3 = 6

…….

…….

12 x 3 = 36

## 2.5 Incrementing alternate numbers

Create an int array called scores containing the following numbers: 25, 15,99,75,47,59,82

Use a for loop to increment every second item. When you’ve done this use another for loop to display all the values in the array. They should look like this:

25

16

99

76

47

60

82

## 2.6 Alternate letters

Use a for loop to display every second letter of the alphabet starting from b.

## 2.7 All times tables

Use a for loop within a for loop to display all of the times tables from 2 to 12. The output should be in this format:

2 times table:

1 x 2 = 2

2 x 2 = 4

……..

……..

12 times table:

1 x 12 = 12

……

12 x 12 = 144

# Section 3 - While Loops

## 3.1 One million

Use a while loop to display all the numbers up to 1 million.

## 3.2 Letters of the alphabet

Use a while loop to display the lower case letters of the alphabet from a to z.

## 3.3 Fibonacci sequence

The Fibonacci sequence starts like this:

0, 1, 1, 2, 3, 5, 8, 13, 21

The first number is 0, the second number is 1. After that each number is the sum of the two numbers before it.

Use a while loop to display all the Fibonacci numbers under 1000.

# Section 4 – Controlling a loop with an if statement

## 4.1 Finding a city

Create a String array called cities which contains the following cities: Glasgow, London, Birmingham, Cardiff, Manchester, Newcastle

Write a loop which loops through the names until it finds a name starting with ‘B’. At this point the loop should terminate.

For each city you should diplay the city name and the message ‘match’ or ‘doesn’t match’. The output should look like this:

Glasgow – doesn’t match

London – doesn’t match

Birmingham – match

## 4.2 First number over 100

Create an int array called numbers which contains the following numbers: 6, 15, 75, 32, 89, 24, 103, 17, 5, 250, 63, 91.

Write a loop which searches through the array until it finds the first number over 100. It should display this number and then terminate.

## In this example it should display 103.

ANSWERS

# Section 1 – For Each loops

## 1.1 Array of courses

Create an array of Strings called courses which contains the following items: “UNIX”, ”SQL”, ”Java”, ”Python”, ”Web Apps”. Use a for each loop to display all of the courses.

e.g. “I love learning UNIX”

“I love learning SQL”

String[] courses = {"UNIX", "SQL", "Java", "Python", "Web Apps"};

**for** (String course : courses){

System.***out***.println("I love learning "+course);

}

## Copying to an ArrayList

Create an ArrayList of type String called coursesList. Use a for each loop to move all the items from the courses array you created in the previous question into the ArrayList.

ArrayList<String> courseList = **new** ArrayList<String>();

**for** (String course : courses){

courseList.add(course);

}

## Displaying courses

Use a for each loop to print out all of the courses stored in the ArrayList you created in question the previous question.

**for** (String course : courseList){

System.***out***.println(course);

}

## Total price

Create an ArrayList of type Double called prices. Add the following prices into the ArrayList: 4.50, 1.99, 9.75, 3.15, 10.25. Use a for each loop to add up all the prices.

Display the total price.

ArrayList<Double> prices = **new** ArrayList<Double>();

prices.add(4.50);

prices.add(1.99);

prices.add(9.75);

prices.add(3.15);

prices.add(10.25);

**double** total = 0.0;

**for** (**double** price : prices){

total += price;

}

System.***out***.println(total);

## Copying to an array

Create an empty double array with 5 elements. Use a for each loop to copy the prices from the ArrayList you made in the previous question into the array.

**double**[] priceArray = **new** **double**[5];

**int** index = 0;

**for** (**double** price : prices){

priceArray[index] = price;

index ++;

}

## Adding numbers from a String

Create a String containing the following text: “10,7,25,3,8”.

Use the String’s split() method to create a String array containing the numbers from the String.

Now use a “for each” loop to add up the numbers in the array and display the total.

String string = "10,7,25,3,8";

String[] numbers = string.split(",");

**int** total = 0;

**for** (String number : numbers){

total += Integer.*parseInt*(number);

}

## System.*out*.println(total);

## Reversing a String

Create a String containing the following text: “desrever neeb sah txet siht”.

Use the String’s toCharArray() method to create an array of chars containing each of the letters from the String.

Finally use a “for each” loop to create a new String which contains all the letters of the original String in reverse order.

Display the new String.

String string = "desrever neeb sah txet siht";

**char**[] letters = string.toCharArray();

String reversed = "";

**for** (**char** letter : letters){

reversed = letter + reversed;

}

System.***out***.println(reversed);

This could also be done with a for loop:

String string = "desrever neeb sah txet siht";

String reversed = "";

**for** (**int** index = string.length()-1; index >=0; index --){

reversed += string.charAt(index);

}

System.***out***.println(reversed);

# Section 2 – For Loops

## 2.1 One million

Use a for loop to display a list of all the numbers from 1 to a million.

**for** (**int** x = 1; x <= 1000000; x++){

System.***out***.println(x);

}

## 2.2 Summing a million

Use a for loop to add up all the numbers from 1 to a million.

The answer should be 500000500000. Think about which type of variable would be suitable for storing a number of this size.

**long** total = 0;

**for** (**int** x = 1; x <= 1000000; x++){

total += x;

}

System.***out***.println(total);

## 2.3 Odd numbers

Use a for loop to display a list of all the odd numbers from from 1 to 20.

**for** (**int** x = 1; x <= 20; x+=2){

System.***out***.println(x);

}

## 2.4 Three times table

Use a for loop to display the three times table from 3 to 36 in the following format:

1 x 3 = 6

2 x 3 = 6

…….

…….

12 x 3 = 36

**for** (**int** x = 3; x<= 36; x+=3){

System.***out***.println((x/3)+" x 3 = "+x);

}

Alternatively:

**for** (**int** x = 1; x<= 12; x++){

System.***out***.println(x+" x 3 = "+(x\*3));

}

## 2.5 Incrementing alternate numbers

Create an int array called scores containing the following numbers: 25, 15,99,75,47,59,82

Use a for loop to increment every second item. When you’ve done this use another for loop to display all the values in the array. They should look like this:

25

16

99

76

47

60

82

**int**[] scores = {25,15,99,75,47,59,82};

**for** (**int** index = 1; index < scores.length; index +=2){

scores[index] ++;

}

**for** (**int** index = 0; index < scores.length; index ++){

System.***out***.println(scores[index]);

}

## 2.6 Alternate letters

Use a for loop to display every second letter of the alphabet starting from b.

**for** (**char** letter = 'b'; letter <= 'z'; letter +=2){

System.***out***.println(letter);

}

## 2.7 All times tables

Use a for loop within a for loop to display all of the times tables from 2 to 12. The output should be in this format:

2 times table:

1 x 2 = 2

2 x 2 = 4

……..

……..

12 times table:

1 x 12 = 12

……

12 x 12 = 144

**for** (**int** table = 2; table <=12; table++){

System.***out***.println(table+" times table");

**for** (**int** multiple = 1; multiple <=12; multiple++){

**int** result = multiple \* table;

System.***out***.println(multiple+" x "+table+" = "+result);

}

System.***out***.println();

}

# Section 3 - While Loops

## 3.1 One million

Use a while loop to display all the numbers up to 1 million.

**int** count = 1;

**while** (count <= 1000000){

System.***out***.println(count);

count ++;

}

## 3.2 Letters of the alphabet

Use a while loop to display the lower case letters of the alphabet from a to z.

**char** letter = 'a';

**while** (letter <= 'z'){

System.***out***.println(letter);

letter++;

}

## 3.3 Fibonacci sequence

**int** previousNumber = 1;

**int** currentNumber = 0;

**while**(currentNumber <= 1000){

System.***out***.println(currentNumber);

**int** temp = currentNumber;

currentNumber += previousNumber;

previousNumber = temp;

}

# Section 4 – Controlling a loop with an if statement

## 4.1 Finding a city

Create a String array called cities which contains the following cities: Glasgow, London, Birmingham, Cardiff, Manchester, Newcastle

Write a loop which loops through the names until it finds a name starting with ‘B’. At this point the loop should terminate.

For each city you should diplay the city name and the message ‘match’ or ‘doesn’t match’. The output should look like this:

Glasgow – doesn’t match

London – doesn’t match

Birmingham – match

String[] cities = { "Glasgow", "London", "Birmingham", "Cardiff", "Manchester", "Newcastle" };

**for** (String city : cities) {

**if** (city.charAt(0) == 'B') {

System.***out***.println(city + " - match");

**break**;

} **else** {

System.***out***.println(city + " - doesn't match");

}

}

## 4.2 First number over 100

Create an int array called numbers which contains the following numbers: 6, 15, 75, 32, 89, 24, 103, 17, 5, 250, 63, 91.

Write a loop which searches through the array until it finds the first number over 100. It should display this number and then terminate.

In this example it should display 103.

**int**[] numbers = {6, 15, 75, 32, 89, 24, 103, 17, 5, 250, 63, 91};

**for** (**int** number : numbers){

**if** (number > 100){

System.***out***.println(number);

**break**;

}

}

Click on the link to give feedback on this exercise:

<https://forms.office.com/Pages/ResponsePage.aspx?id=glOkWCW86EGcIlkUGYi-mhn5SZBIBvtIk9fURIMGJDZUNEVEMVNMVllOSkZaOUgxVFBZU1ZJWVAySy4u>