**School of Digital Media & Infocomm Technology (DMIT)**

**ST2111 Mobile Application Development I**

**Practical 5**

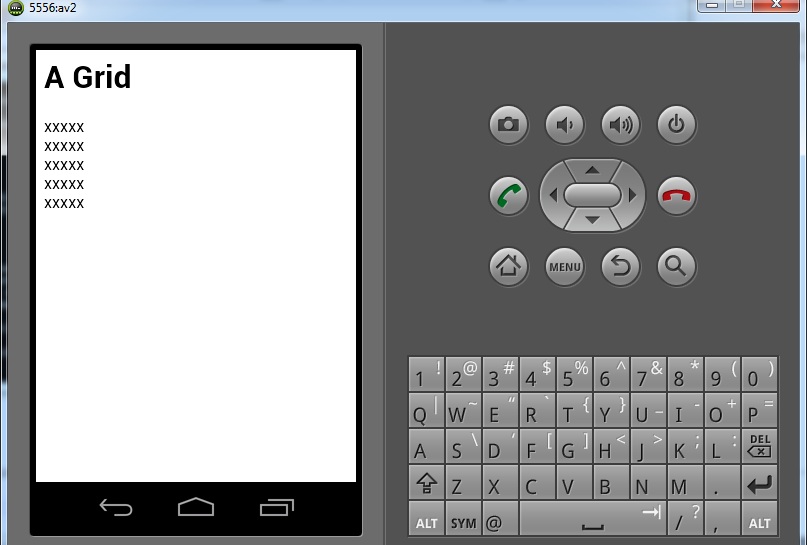
**More Loops**

|  |
| --- |
| Objectives:  After completing this lab, you should be able to:   * Write additional programs which utilize the for loop, while loop, do-while loop * Write programs which utilize nested loops |

**Exercise 1: Write Programs Involving Loops**

Use a nested for loop to show the tables for every multiplier from 1 to 10 (100 results total).

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-1** folder
3. Write a program that:
   * Prompts a user for length and width
   * The program should use a nested loop to display a grid of ‘X’ based on the height and width.



**Sample Output**

|  |
| --- |
| var height = prompt("How high do you want the grid?(1-10 is good)","10");  var width= prompt("How wide do you want the grid?(1-10 is good)","10");  var a\_line;  document.write("<h1>A Grid</h1>");  for (var height\_loop = 0; height\_loop < height; height\_loop++)  {  a\_line = "";  for (var width\_loop = 0; width\_loop < width; width\_loop++)  {  a\_line+="x";  }  document.writeln(a\_line + "<br>");  }  **Sample Code** |

**Exercise 2: Write Programs Involving Loops**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-2** folder
3. The formula for converting fahrenheit to celsius is:

F= 32 + (C \* 180/100)

1. Write a program that prints a conversion table from Celsius to Fahrenheit (0-100 degrees). Implement using for loop.
2. You can use **document.write** to format your output with **HTML** tags. See sample output below.

|  |  |
| --- | --- |
| **Celsius** | **Fahrenheit** |
| 0 | 32 |
| 1 | 33.8 |
| 2 | 35.6 |
| 3 | 37.4 |
| 4 | 39.2 |
| 5 | 41 |
| 6 | 42.8 |
| 7 | 44.6 |
| 8 | 46.4 |
| 9 | 48.2 |
| 10 | 50 |

**Sample output**

**Exercise 3: Write Programs Involving Loops**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-3** folder
3. Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen.

"0 is even"

"1 is odd"

"2 is even"

**Sample Output**

**Exercise 4: Break and Continue**

1. Create a copy of the **practical1-1** in the same **MAD1** folder.
2. Rename the copied folder as **practical5-4** folder
3. Write the output of each of the following listings below.
4. Explain the difference between the **break** and **continue** statements.

|  |  |
| --- | --- |
| **Loop Wi**  **th Break Statement** | **Output** |
| var text="";  for(var i = 0; i < 10; i++)  {  if (i == 3)  {  document.write("will be breaking");  **break;**  }  text += "The number is " + i + "<br>";}  document.write(text);  } |  |

|  |  |
| --- | --- |
| **Loop With Continue Statement** | **Output** |
| var text="";  for(var i = 0; i < 10; i++)  {  if (i == 3)  {  document.write("will be continue");  **continue;**  }  text += "The number is " + i + "<br>";}  document.write(text); |  |

**Answer:** The **break statement** breaks the loop and continues executing the code after the loop (if any). However, the **continue statement** breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

**Advanced Exercises (Optional). Test your level with these advanced exercises.**

**Exercise 5: Write the Fibonacci sequence using a loop**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-5** folder
3. Write a program that outputs the Fibonacci numbers less than 30,000. Each Fibonacci number is the sum of its two predecessors. The First two Fibonacci numbers are 1 and 1. Thus the sequence is

1,1,2,3,5,8,13,21,34

**Exercise 6: Check whether a number is a perfect number:**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-6** folder
3. A perfect number is a positive integer where the sum of all its positive divisors, except itself, is equal to the number itself.

For example 6 is a perfect number. 1,2 and 3 are its divisors and the sum of divisors=1+2+3=6.

28 is a perfect number as its divisors: 1 + 2 + 4 + 7 + 14 =28

1. Write a program which do the following:
   1. Prompt a user to enter an integer
   2. The program will use a loop to check whether a number is a perfect number.

**Exercise 7: List all perfect numbers from 1 to 10000**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical5-7** folder
3. Modify Exercise 6 to list all perfect numbers from 1 to 10000.
4. List the perfect numbers: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Exercise 8: Run Exercises in the Android Simulator**

You may run your exercises using the Phonegap Android Simulator to view the output. For example, if you wish to test out **practical5-1**, do the following:

1. From the [Start screen,](http://www.computerhope.com/jargon/w/windows8.htm) click **Command Prompt** to open the Windows console.
2. Navigate to **MAD1** folder. (Note: If your MAD1 folder is in **c: drive**, type **cd\** to bring you to the root directory first, then type **cd mad1** to go to MAD1 folder. If your MAD1 folder is in **d: drive**, type **d:** to change to d: drive first, then type **cd mad1** to go to MAD1 folder.**)**
3. Create the new app by typing **phonegap create practical5-1app**and press enter.
4. Open a text editor and open the file located in **mad1\practical5-1app\www\index.html**.
5. Replace the text with the code in **Listing D** below.
6. Copy **script.js** from **practical5-1** to the **www** directory.
7. Type **phonegap run android**.
8. Your JavaScript program is run in the Android Simulator.

**Listing A. index.html**