**School Of computing**

**Practical 4:   
Repetitions**

**(do-while & while loops)**

**ST0502 Fundamentals of Programming**

**What you will learn / do in this lab**

* Use do … while loop
* Use while loop
* Use nested loops

1. Determine the output and/or identify the error in each of the following code segments:

a)

|  |
| --- |
| var r = 0; Loop ends here  while (r < 10); {  console.log ("r is " + r);  r += 3;  } |
| Output/ Error:  Infinite loop |

b)

|  |
| --- |
| var s = 11;  do {  s -= 2;  console.log ("s is " + s);  } while (s >= 1); |
| Output/ Error:  s is 9  s is 7  s is 5  s is 3  s is 1  s is -1 |

c)

|  |
| --- |
| var t = 0, u = 4;  while (t <= 3) {  t++;  for (var v = t; v < u; v++) {  console.log (v + t);  }  } |
| Output/ Error:  0 v=1 t=1  2 v=2 t=2  4 v=3 t=3  6 v=4 t=4  2  3  4  4  5  6 |

d)

|  |
| --- |
| var m = 10, n = 4;  while (m > 5) {  if (m % n != 1)  console.log(m);  m--;  } |
| Output/ Error:  10  9 (10 % 4 = 1 so it is not printed out)  8  7  6 |

e)

|  |
| --- |
| var msg = "";  for (var d = 0; d < 3; d += 1) {  if (d % 2 == 1) {  for (var e = 1; e <= d; e++) {  msg += e;  }  }  msg += "\n";  for (var f = 6; f > 2; f-=2) {  msg += f;  }  }  console.log(msg); |
| Output/ Error:  14 |

1. Using the while loop, write the program to generate each of the following output:

a)

|  |
| --- |
| Output:  6 12 18 24 30 |
| var a = 6  while ((a > 5) && (a<31)) {      process.stdout.write(a+' ');      a = a + 6;  }  Correct version ^  var number = 1  while (number<=5) {  s+=number \* 6 + ‘ ‘ ;  number++}  Console.log(s) |
|  |

b)

|  |
| --- |
| Output:  80-40-20-10-5 |
|  |

1. Using the do-while loop, write the program to generate each of the following output:

a)

|  |
| --- |
| Output:  2 4 8 16 32 64 |
| var a = 2;  do {      process.stdout.write(a+' ')      a = a\*2  } while ((a > 1) && (a<65)) |

b)

|  |
| --- |
| Output:  91 + 80 – 69 + 58 - 47 |
|  |

You may open up your Visual Studio Code (VSC) and work from there for the remaining

questions.

[Optional : After you have completed each question, try challenging yourself to include some data validation in your program. Do remember to prepare test cases to ensure your validation works for all possible cases.]

1. Write a **do-while** loop to prompt for the number of children from the user until a valid input in the range of 0 to 50 is captured. An appropriate error message should be displayed for invalid input.

Create a Test plan showing all possible test cases used to test your program.

Sample output:

|  |
| --- |
| Please enter the number of children: **-5**  Invalid number of children!  Please enter in the range of 0 to 50.  Please enter the number of children: **51**  Invalid number of children!  Please enter in the range of 0 to 50.  Please enter the number of children: **4**  Input Accepted! Program terminated...  var userinput = require('readline-sync')    do{      var numberCh = userinput.questionInt('Please enter the number of children:')      if (isNaN(numberCh) || numberCh<0 || numberCh>50)          console.log('Invalid number of children!\nPlease enter in the range of 0 to 50');      else          console.log('Input Accepted! Program terminated.....')    }while ((numberCh<0) || (numberCh>50)); |

1. Write a program using **do-while** loop that prompts user to enter a number. The number entered must be between 100 and 200 (both inclusive). If the number entered is not within the range, it displays an error message and prompts user to re-enter.

Create a Test plan showing all possible test cases used to test your program.

Sample output:

|  |
| --- |
| Enter a number: ***50***  Error! Please enter a number between 100 and 200.  Enter a number: ***245***  Error! Please enter a number between 100 and 200.  Enter a number: ***123***  Input Accepted: End of Program!  var input=require('readline-sync')  do{      var num = input.questionInt('Enter a number: ')      if (isNaN(num) || 100>num || num>200)          console.log('Invalid')  }while(100>num || num>200  ) |

1. Write a program to handle the following requirements:

* Prompts the user to enter an **integer**
* Display an appropriate message if the number is divisible by 3 but not 5.
* The user will be prompted repeatedly to enter the next number until athe user enters 0 as the input.
* Create a Test plan showing all possible test cases used to test your program.

Sample output:

|  |
| --- |
| Enter any number or (0) to quit: ***12***  12 is divisible by 3 but not 5? true  Enter any number or (0) to quit: ***30***  30 is divisible by 3 but not 5? false  Enter any number or (0) to quit: ***0***  Program Terminated…  var userInput = require('readline-sync')  for (var x = 1;x<3;x++) {      var num1 = userInput.question('Enter any number or (0) to quit: ');      if (num1 == 0)          console.log('Program Terminated...')      else          if (num1 % 3 == 0 && num1 % 5 != 0)              console.log(num1 + ' is divisible by 3 but not 5? true')          else if (num1 % 3 != 0 && num1 % 5 == 0)              console.log(num1 + ' is divisible by 3 but not 5? false')      } |

**[Optional Questions]**

1. Modify question 6, such that users are only allowed to enter a number that are between 50 and 100 (inclusive) or 0 to quit the program.

*Recap the use of Math.abs in Practical 3b, Q5.*

Sample output:

|  |
| --- |
| Enter any number or (0) to quit: ***188***  Error: Out of range!  Enter any number or (0) to quit: ***44***  Error: Out of range!  Enter any number or (0) to quit: ***63***  63 is divisible by 3 but not 5? true  Enter any number or (0) to quit: ***55***  55 is divisible by 3 but not 5? false  Enter any number or (0) to quit: ***0***  Program Terminated… |

1. Write a program that produces a multiplication table with 25 rows of computations. Allow the user to input the first and last base values for the multiplication table.

Display a column in the table beginning with the first base inputted value. The last column should be the ending base value entered. The first row should be for 1 times the beginning base, 1 times the (beginning base value + 1), through 1 times the ending base value. The last row should be for 25 times the beginning base, 25 times the (beginning base value + 1), through 25 times the ending base value. Base values can range from 2 through 7.

Display an aesthetically formatted multiplication table. An example of output produced when 2 and 7 are entered appears in the figure below.

Sample output:

|  |
| --- |
| Enter first base value: 2  Enter last base value: 7  n 2 3 4 5 6 7  1 2 3 4 5 6 7  2 4 6 8 10 12 14  3 6 9 12 15 18 21  4 8 12 16 20 24 28  5 10 15 20 25 30 35  6 12 18 24 30 36 42  7 14 21 28 35 42 49  8 16 24 32 40 48 56  9 18 27 36 45 54 63  10 20 30 40 50 60 70  11 22 33 44 55 66 77  12 24 36 48 60 72 84  13 26 39 52 65 78 91  14 28 42 56 70 84 98  15 30 45 60 75 90 105  16 32 48 64 80 96 112  17 34 51 68 85 102 119  18 36 54 72 90 108 126  19 38 57 76 95 114 133  20 40 60 80 100 120 140  21 42 63 84 105 126 147  22 44 66 88 110 132 154  23 46 69 92 115 138 161  24 48 72 96 120 144 168  25 50 75 100 125 150 175 |

**- END -**