**School Of computing**

**Practical 6: Functions**

**ST0502 Fundamentals of Programming**

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

* Use pre-packaged JavaScript functions
* Create and invoke a function
* Passs arguments to a function

1. Given the following JavaSript program, write the output for each of the following. (Each part is to be treated independently.)

|  |
| --- |
| function printMe() {      for (var i = 0; i < 3; i++)          console.log("My name is Dodo");  }  function printYou() {      for (var x = 0; x < 2; x++)          console.log("Your name is Bobo");  }  function printUs() {      var i = 1;      evenCounter = 0;      do {          console.log("We are FOPians");          if (i % 2 == 0)              evenCounter++;          i++;      } while (i < 5);      console.log(evenCounter);  }  // Please refer to Part(a) to (d) below for  // statements used to invoke selected function(s). |

|  |  |  |
| --- | --- | --- |
|  | Statements to invoke selected function(s) | Output |
| a) | printMe();  printYou();  console.log("End of program"); | My name is Dodo  My name is Dodo  My name is Dodo  Your name is Bobo  Your name is Bobo  End of program |
| b) | printUs();  console.log("End of program"); | We are FOPians  We are FOPians  We are FOPians  We are FOPians  2  End of program |
| c) | We are FOPians  We are FOPians  We are FOPians  We are FOPians  2  Your name is Bobo  Your name is Bobo  End of program  var x = 4;  printUs();  if (x == 4)      printYou();  else      printMe();  console.log("End of program"); |  |
| d) | var x = 45; y = 65; z = 30;  var a = 0;  printMe();  printus;  printYou;  a \*= x++ \*\* z-- + y--\*z \*\* 4;  console.log(a); | My name is Dodo  My name is Dodo  My name is Dodo  0 |

1. What is the output of the following statements?

|  |  |  |
| --- | --- | --- |
|  | Functions without Return statements | Output |
| (a) | function halfValue(x) {      x = x / 2;  }  function computeTot(x, y, z) {      var total = 0;      total = x + y + z;      console.log(total);  }  computeTot(4, 5, 6);  computeTot(10, 2, 1);  computeTot(20, 10); | 15  13  undefined |
| (b) | printWhat(4, 1, 5);  function printWhat(a, b, c) {      a += b;      if (a % c == 0) {          console.log("Hello");      } else if (b < c) {          console.log("Monkey");      } else {          console.log("Zoo");      }      console.log("Lion");  } | Hello  Lion |
| (c) | function cal\_Value(v, w) {      switch (v) {          case 1: w += v;          case 2: w -= v;              break;      }      console.log('Value of w is ' + w);  }  cal\_Value(1, 2); | Value of w is 2 |

|  |  |
| --- | --- |
|  |  |
| (d) | //Assuming user enters : FOP, FED, FOC, EGC1  var input = require('readline-sync');  function createArr(x) {     var modules = new Array();     for (var i = 1; i <= x; i++) {        modules.push(input.question('Enter module ' + i + ': '));     }     console.log(modules[1] + modules[0]);  }  createArr(4); |
|  | Output :  Enter module 1:  Enter module 2:  Enter module 3:  Enter module 4:  FOPFED |

1. What is the output of the following statements?

|  |  |
| --- | --- |
|  | Function with Return statement |
| (a) | function calculateVol(x) {      var vol = x \* x \* x;      return vol;  }  function calculateArea(length) {      var area = length \* length;      return area;  }  console.log("Area is :" + calculateArea(3));  console.log("Volume is :" + calculateVol(3)); |
|  | Output:  Area is 9  Volume is 27 |
|  | Function with Return statement |
| (b) | function checkArr() {      var arr1 = [-1, 0, 'one', '2', true, false, 5, 9];      if (arr1.length % 2 == 0)          return true      else          return false;  }  var result = checkArr();  console.log("Array has even elements : " + result); |
|  | Output:  Array has even elements: true |

1. Write a function ***allMultiple()*** that takes in three integers, *num1*, *num2* and *x*. The function prints out all the integers from *num1* to *num2* that are a multiple of *x*. (Assume *num1* will always be smaller or equal to *num2*). For example, a function call on ***allMultiple(10, 35, 5)*** prints 10 15 20 25 30 35.

You may use process.stdout.write to print the numbers on one line

|  |
| --- |
| function allMultiple(num1,num2,x){     for(;num1<=num2;num1++){        if(num1%x == 0)           process.stdout.write(num1 + ' ')     }  } |
|  |

5) The Math library has a prebuilt function called **random()** that generates random floating-point numbers in the range of 0 (inclusive) to 1(exclusive).

a) Write the codes that will generate a random integer in the range of 0 to 68.

b) Write the codes that will generate a random integer in the range of 19 to 87.

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

questions.

6) Modify Practical 3b Q3, ***Multiplication*** ***Table*** program.

Write a function ***printTable*** to print the “5 times” multiplication table. The function does not take in any parameter and does not return any value. Invoke the function to test its functionality.

for(i = 1;i<=12;i++){

   prinTable()

}

function prinTable(){

      console.log(5 + ' x ' + i  +  ' = ' + (i\*5))

}

7) Modify Practical 3a Q8, ***Prize Money*** program.

a) Write a function ***getInput*** that prompts for the rank of the contestant and returns the value entered. The function does not take in any parameter and returns an integer value.

1. Write a function ***printPrize*** that takes in ***rank*** as its parameter. It checks and displays the prize money based on the rank. The function does not return any value.
2. Invoke the ***getInput*** and ***printPrize*** functions to test the program.
3. var input = require('readline-sync');
4. function getInput() {
5. var rank = parseInt(input.question('Please enter your rank: '))
6. return rank
7. }
9. function printPrize(rank) {
10. var prize;
11. switch (rank) {

      case 1:

1. prize = 1000;
2. break;
3. case 2:
5. prize = 800;
6. break;
7. case 3:
8. prize = 700;
9. break;
10. case 4:
11. case 5:
12. prize = 300;
13. break;
14. default:
15. prize = 20;
16. }
17. console.log('Your prize money is: $' + prize);
18. }
19. printPrize(getInput());

8) Modify Practical 3a Q3, Display bigger number.

a) Write an integer function ***getInput*** as follows:

function getInput(**s**)

The function takes in a String parameter that will be used as part of the input prompt, e.g. "Please enter **1st** integer" and "Please enter **2nd** integer". It returns an integer value corresponding to the user input.

b) Write a function ***findMax*** that takes in 2 integer values ***num1*** and ***num2***. It compares which number is bigger and returns one of the following string values:

"1st number is bigger"

"2nd number is bigger"

"The 2 numbers are equal"

c) In the main part of the program, invoke ***getInput(“1st”)*** and ***getInput(“2nd”)*** to get the values of ***num1*** and ***num2***. Invoke ***findMax*** and display the string returned.

function getInput(s) {

   num = input.question('Please enter ' + s + ' integer: ');

   return num

}

function findMax(num1,num2){

   if(num1>num2){

      console.log('1st number is bigger')

   }

   else if (num2>num1){

      console.log('2nd number is bigger')

   }

   else{

      console.log('The 2 numbers are equal')

   }

}

var input = require('readline-sync');

var num1 = getInput('1st')

var num2 = getInput('2nd')

findMax(num1,num2)

findMax(getInput('1st'),getInput('2nd'))

9. You are given 2 functions below : ***readYear*** and ***isLeapYear***

|  |
| --- |
| function readYear(str) {      var yr = input.questionInt("Enter " + str + " year : ");      return yr;  }  function isLeapYear(year) {      return (year % 4 == 0 && year % 100 != 0 || year % 400 == 0); |

* ***readYear*** function takes in a string to read the starting year and ending year.
* ***isLeapYear*** function checks for leap year, returns a boolean value

Write the main program to do the followings:

1. Invoke ***readYear*** function to take in a string to read the starting year and ending year.
2. Validate the ending year cannot be smaller than the starting year.
3. Use a ***for loop*** and to invoke ***isLeapYear*** function to check the years from the starting year to the ending year and display the message as shown.

**Sample output:**

|  |
| --- |
| Enter start Year: **2000**  Enter end Year: **2004**  2000 is a leap year  2001 is not a leap year  2002 is not a leap year2003 is not a leap year  2004 is a leap year  varinput= *require*('readline-sync');  function *readYear*(str) {      varyr= *input.questionInt*("Enter "+str+" year : ");  *return* yr;  }  function *isLeapYear*(year) {  *return* (year % 4 == 0 && year % 100 != 0 || year % 400 == 0);  }  year = *readYear*("start")  yearEnd = *readYear*("end")  *for*(year;year<=yearEnd;year++ ){      if(*isLeapYear*(year))  *console.log*(year + ' is a leap year')      else  *console.log*(year + ' is not a leap year')  } |

10) Write a program, **using nested loop**, that generates 5 sets of 4 single-digit numbers (0 to 9) randomly as follows:

|  |
| --- |
| **9 3 3 9**  **3 9 8 8**  **2 3 7 8**  **0 1 6 6**  **7 5 2 5** |

**[Optional Questions]**

11) Complete the missing parts of the following program by writing the appropriate functions.

|  |
| --- |
| var input = require("readline-sync");  var num = readInput();  var pat = choosePattern();  switch(pat) {  /\*part f\*/  }  function readInput() {  /\*part a\*/  }  function choosePattern(){  /\*part b\*/  }  function printPattern1(n){  /\*part c\*/  }  function printPattern2(n){  /\*part d\*/  }  function printPattern3(n){  /\*part e\*/  } |

1. Code the ***readInput()***function to prompt the user for the number of rows. The function then returns the number entered.

|  |
| --- |
| Enter the number of rows: **4** |

1. Complete the ***choosePattern()*** function to prompt the user for the pattern he wish to be printed. The function returns the number entered to the calling code.

|  |
| --- |
| 1. Print Pattern 1  2. Print Pattern 2  3. Print Pattern 3  4. Exit  > **3** |

1. Complete the ***printPattern1(n)*** function to print ***Pattern 1*** as shown (assume ***n*** is 4).

Pattern 1

1 1 1 1

2 2 2 2

3 3 3 3

4 4 4 4

1. Complete the ***printPattern2(n)*** function to print ***Pattern 2*** as shown (assume ***n*** is 4).

Pattern 2

1 2 3 4

1 2 3 4

1 2 3 4

1 2 3 4

1. Complete the ***printPattern3(n)*** function to print ***Pattern 3*** as shown (assume ***n*** is 4).

Pattern 3

1 2 3 4

2 4 6 8

3 6 9 12

4 8 12 16

1. Complete the switch statement based on the pattern selected to invoke ***printPatternX*** function coded to produce the output.

Sample output 1:

|  |
| --- |
| Enter the number of rows: **4**  1. Print Pattern 1  2. Print Pattern 2  3. Print Pattern 3  4. Exit  > **1**  1 1 1 1  2 2 2 2  3 3 3 3  4 4 4 4 |

Sample output 2:

|  |
| --- |
| Enter the number of rows: **4**  1. Print Pattern 1  2. Print Pattern 2  3. Print Pattern 3  4. Exit  > **3**  1 2 3 4  2 4 6 8  3 6 9 12  4 8 12 16 |

**- END -**