**CM1100 Javascript Assessment 2**

**Friday 21st November 2019: 12:00 – 14:00**

**This is an Open Moodle Assessment**

**You are allowed to look at any resource on Moodle but you cannot use other resources (electronic or hardcopy).**

**However, this is an individual assessment, you should refrain from sharing documents, talking, using e-mail or communicating in anyway with other students.**

**Anyone caught attempting to break the examination rules will be graded F.**

The mark for the assessment will be converted into a grade using the following grading criteria.

|  |  |
| --- | --- |
| **Mark** | **Grade** |
| 70% and above | A |
| 60% - 69% | B |
| 50% - 59% | C |
| 40% - 49% | D |
| 35% - 39% | E |
| Below 35% | F |

Your submission should be written as a single .txt file and named with your second name and matric number, e.g. my file would be McDermott1234567.txt.

Notes:

* At the start of Q1, you should include a Statement of Compliance that is **complete** and **factually accurate**.
* You should include comments in your program to aid readability.
* Use sensible variable names to make your program readable.
* Make sure you declare all your variables correctly.
* Use explicit type conversion where necessary.
* The program that you submit is expected to run. If any part of your program doesn’t work, don’t delete that code but comment it out and add a further comment to explain that it doesn’t work and, if possible, why you think it doesn’t work.
* There are marks allocated for "program style and structure". You will gain marks if you structure your program using sensible functions. However, you will still gain **some** marks if you use sequential code (i.e. not functions) instead.

Q1 is worth 80 marks, Q2 is worth 20 marks and Q3 is worth 10 marks. Therefore, it is suggested that you spend about 80 minutes on Q1 and around 30 minutes on Q2 and Q3. This then gives 10 minutes time for checking your answers.

**Save your work in a .txt file at regular intervals so that you do not lose it if the runner crashes.**

**Q1. THE HORROR! THE HORROR! (80 marks)**



**I'M A TALENTLESS**

**D-LIST CELEBRITY…**

**RESCUE MY CAREER!**

*"Everyone gets everything he wants. I wanted a mission, and for my sins, they gave me one. Brought it up to me like room service. It was a real choice mission, and when it was over, I never wanted another."*

IT Support for "I'm a Talentless D-List Celebrity"

Deep in the Australian Jungle, unscrupulous former Kangaroo traders, Anthony Kurtz and Declan Kilgore, have used the power of TV to dazzle a group of desperate minor celebrities into joining their cult-like competition, dedicated to mind-numbing chatter and the consumption of oversized insects. This competition is then broadcast to a mesmerised audience throughout the world. In order to generate TV ratings, the individuals are subjected to a series of brutal, (mostly) insect-based tortures which determine their position in the competition and ultimately, which one of them survives. After each activity, one of the group members is eliminated - with extreme prejudice - from of the competition.

These unfortunate individuals, who had previously competed in the hit TV show "Strictly Dances with Wolves" during the infamous KenGate Scandal, are all too familiar with death, destruction and the desire to go to any length to avoid doing a day's work. But this is something much, much worse!

You have been called in as a consultant in order to test the computer system that controls the camp management and security. You should write some Javascript code which accomplishes the following tasks:

1. Create an array of names of the members of the group: "Flash", "Anne", "Chelsee",

"Corrie", "Des" and "Zed" (in that order).

(5 marks)

**//1(a)-----------------------------------------------------------**

**var names = ["Flash", "Anne", "Chelsee", "Corrie", "Des", "Zed"];**

**alert("The names of the contestants are " + names.toString());**

Marks for:

* Declaration, Assignment, Array syntax, Populating array, Test /1 each

1. The first activity in the show is requires the individuals to consume a number of beetles to try to avoid elimination (and so escape appearing as an item on the camp breakfast menu for the next day). You are required to test the cutting-edge beetle-logging software that has been installed to keep track of the number of insects eaten.

For each group member, you should generate and store a random integer in the range 0 to 1000, corresponding to the number of beetles consumed in one hour. The group member with the lowest score is eliminated. You should display a message stating which person has been eliminated and which five contestants still remain.

(10 marks)

**//1(b)-----------------------------------------------------------**

**var beetles = new Array(names.length);**

**for (let i = 0; i < beetles.length; i++){**

**beetles[i] = parseInt(Math.random()\*1001);**

**}**

**//alert(beetles.toString());**

**var min = beetles[0];**

**var personIndex = 0;**

**for (let i = 0; i < beetles.length; i++){**

**if (beetles[i] < min){**

**min = beetles[i];**

**personIndex = i;**

**}**

**//alert(personIndex);**

**}**

**alert("The minimum number of beetles is " + min + " consumed by " + names[personIndex] + "\nThat person is eliminated.");**

**names.splice(personIndex,1);**

**alert(names);**

**//1(b)--Using Functions--------------------------------------------**

**function makeBeetleArray(){**

**var anArray= new Array(names.length);**

**for (let i = 0; i < anArray.length; i++){**

**anArray[i] = parseInt(Math.random()\*1001);**

**}**

**return anArray**

**}**

**var beetles = makeBeetleArray();**

**console.log("-------------------------------------------------------");**

**console.log("The number of beetles eaten by each contestant are [" + beetles.toString() + "]\n");**

**function removeLowestElement(nameArray, numberArray){**

**var min = numberArray[0];**

**var personIndex = 0;**

**for (let i = 0; i < numberArray.length; i++){**

**if (min > numberArray[i] ){**

**min = numberArray[i];**

**personIndex = i;**

**}**

**}**

**console.log("The person removed was " + nameArray[personIndex]**

**+ " at position " + personIndex**

**+ " (starting at zero.)");**

**nameArray.splice(personIndex,1);**

**return nameArray;**

**}**

**var secondRoundContestants = removeLowestElement(names, beetles);**

Marks for

* Creating array of beetle numbers /3
* Finding minimum element of array /2
* Finding index of minimum element of array /1
* Removing associated contestant /3
* Output /1

1. To enhance the alienation process, upon entering the jungle, each member of the group is assigned a new name based on their Spirit Animal and an adjective that may (possibly) describe that creature.

The list of possible animals are:

* "wombat", "spider", "platypus", "treefrog", "cockroach", and "koala".

The list of adjectives are:

* "slimy", "venomous", "scaly", "malodorous", "screechy", "snappy" and "creepy".

You should write some code which will generate a message giving the name of each contestant together with the adjective and spirit animal. This should have the form:

<name> is a <adjective> <animal>.

For example, one message might read "Des is a venomous wombat".

Note that these combinations do not have to be unique (e.g. "creepy spider" could be assigned to more than one person).

The number of characters in the message (including spaces) should be found and the person with the lowest number should be eliminated from the competition. If there is a tie in the smallest number of letters, the first person found with that number is eliminated.

Display the name of the person eliminated and the four people left.

(15 marks)

**//1(c)-----------------------------------------------------------**

**let animals = ["wombat", "spider", "platypus", "treefrog", "cockroach",**

**"koala"];**

**let adjectives = ["slimy", "venomous", "scaly", "malodorous",**

**"screechy", "snappy", "creepy"];**

**let responses = [];**

**let responsesLength = [];**

**for (let i = 0; i < names.length; i++){**

**animalNumber = parseInt(Math.random()\*animals.length);**

**adjectiveNumber = parseInt(Math.random()\*adjectives.length);**

**responses[i] = names[i] + " is a " + adjectives[adjectiveNumber]**

**+ " " + animals[animalNumber];**

**}**

**alert(responses.toString());**

**for(let i = 0; i < responses.length; i++){**

**responsesLength[i] = responses[i].length;**

**}**

**alert(responsesLength.toString());**

**personIndex = 0;**

**min = responsesLength[0];**

**for (let i = 0; i < responsesLength.length; i++){**

**if (responsesLength[i] < min){**

**min = responsesLength[i];**

**personIndex = i;**

**}**

**}**

**alert("The minimum number of characters is " + min + " for "**

**+ names[personIndex]+ "\nThat person is eliminated.");**

**names.splice(personIndex,1);**

**alert("The remaining contestants are " + names.toString());**

**//1(c) Using Functions---------------------------------------------**

**let animals = ["wombat", "spider", "platypus", "treefrog", "cockroach",**

**"koala"];**

**let adjectives = ["slimy", "venomous", "scaly", "malodorous",**

**"screechy", "snappy", "creepy"];**

**function assignAnimals(nameArray){**

**let outputArray = [];**

**for (let i = 0; i < nameArray.length; i++){**

**animalNumber = parseInt(Math.random()\*animals.length);**

**adjectiveNumber = parseInt(Math.random()\*adjectives.length);**

**outputArray[i] = "\n" + nameArray[i] + " is a "**

**+ adjectives[adjectiveNumber] + " "**

**+ animals[animalNumber] ;**

**}**

**return outputArray;**

**}**

**let responses = assignAnimals(secondRoundContestants);**

**console.log(responses.toString());**

**console.log("\n--------------------------------------------------\n");**

**function makeResponseLengthArray(){**

**let responsesLengthArray = [];**

**for(let i = 0; i < responses.length; i++){**

**responsesLengthArray[i] = responses[i].length;**

**}**

**return responsesLengthArray;**

**}**

**let responsesLength = makeResponseLengthArray();**

**console.log(responsesLength);**

**let thirdRoundContestants**

**= removeLowestElement(secondRoundContestants, responsesLength);**

**console.log("\n---------------------------------------------------\n");**

**console.log(thirdRoundContestants);**

Marks for:

* Setting up arrays /4
* Creating responses /4
* Finding length of responses /2
* Finding minimum length /1
* Eliminating person with minimum length /3
* Output /1

1. Each remaining contestant is fitted with a noise sensor and sent to lie in a cage full of scorpions for an hour. Each sensor displays the loudness of their screams on a scale of 1 to 20. The person with the largest loudness score is then fed to the attack kangaroos that patrol the compound. If there is a tie, the last person found with that score is eliminated.

You should test the control software for sensor by simulating a loudness score for each person. This is done by randomly generating a value in the correct range for each remaining contestant and eliminating the person with the highest score.

You should display a suitable message and the remaining list of three contestants.

(5 marks)

**//1(d)-----------------------------------------------------------**

**let loudness = new Array(names.length)**

**for (let i = 0; i < loudness.length; i++){**

**loudness[i] = parseInt(Math.random()\*21 + 1);**

**}**

**alert(loudness);**

**personIndex = 0;**

**max = responsesLength[0];**

**for (let i = 0; i < responsesLength.length; i++){**

**if (responsesLength[i] > max){**

**max = responsesLength[i];**

**personIndex = i;**

**}**

**}**

**alert("The maximum loudness score was " + max + " for " + names[personIndex] + ".\nThat person is eliminated.");**

**names.splice(personIndex,1);**

**alert("The remaining contestants are " + names.toString());**

**//1(d)-----------------------------------------------------------**

**function getLoudnessArray(){**

**let loudnessArray = new Array(thirdRoundContestants.length);**

**for (let i = 0; i < loudnessArray.length; i++){**

**//Make loudness values negative so we can re-use**

**//"removeElement" function which removes the lowest value**

**//The lowest negative value would correspond to the**

**//largest value**

**loudnessArray[i] = -parseInt(Math.random()\*19 + 1);**

**}**

**return loudnessArray;**

**}**

**let loudness = getLoudnessArray();**

**console.log(loudness);**

**let fourthRoundContestants**

**= removeLowestElement(thirdRoundContestants, loudness);**

**console.log("\n------------------------------------------------------\n");**

**console.log(fourthRoundContestants);**

**console.log("\n------------------------------------------------------\n");**

Marks for:

* Setting up array /1
* Populating array /1
* Finding minimum /1
* Removing associated contestant /1
* Output /1

1. The next activity is to gather bananas while trying to avoid a swarm of killer bees. Each person will have four attempts to collect them and can carry, at most, six bananas on each attempt. The information on the number of bananas collected by each contestant in each round is to be held in a table.

You are required to test the software by generating appropriate entries in the table and performing the relevant calculations. The results should be displayed on the screen.

The person eliminated in this round is the one with the lowest number of bananas

State the person who is eliminated and the two remaining contestants.

(15 marks)

**//1(e)-----------------------------------------------------------**

**let bananas = new Array(names.length);**

**let numberOfRounds = 4;**

**for (let round = 0; round < names.length; round++){**

**bananas[round] = new Array(numberOfRounds);**

**}**

**var output = "";**

**for (let i= 0; i < bananas.length; i++){**

**for (let j = 0; j < bananas[0].length; j++){**

**bananas[i][j] = parseInt(Math.random()\*6 + 1);**

**//output += bananas[i][j] + " ";**

**}**

**//output += "\n";**

**}**

**//alert(output);**

**let bananaNumbers = [0,0,0];**

**for (let i = 0; i < bananas.length; i++){**

**for (let j = 0; j < bananas[0].length; j++){**

**bananaNumbers[i] += bananas[i][j];**

**}**

**}**

**alert(bananaNumbers);**

**personIndex = -1;**

**main = bananaNumbers[0];**

**for (let i = 0; i < bananaNumbers.length; i++){**

**if (min > bananaNumbers[i]){**

**min = bananaNumbers[i];**

**personIndex = i;**

**}**

**}**

**alert("The minimum number of bananas collected is " + min + " for "**

**+ names[personIndex]+ "\nThat person is eliminated.");**

**names.splice(personIndex,1);**

**alert("The remaining contestants are " + names.toString());**

**//1(e) Using Functions--------------------------------------------**

**function getBananas(nameArray){**

**let bananas = [];**

**let numberOfRounds = 4;**

**let bananasPerRound = 6**

**for (let round = 0; round < nameArray.length; round++){**

**bananas[round] = new Array(numberOfRounds);**

**}**

**var output = "";**

**for (let i= 0; i < bananas.length; i++){**

**for (let j = 0; j < bananas[0].length; j++){**

**bananas[i][j] = parseInt(Math.random()\*bananasPerRound + 1);**

**}**

**}**

**let bananaNumberArray = [0,0,0];**

**for (let i = 0; i < bananas.length; i++){**

**for (let j = 0; j < bananas[0].length; j++){**

**bananaNumberArray[i] += bananas[i][j];**

**}**

**}**

**return bananaNumberArray**

**}**

**let bananaNumbers = getBananas(fourthRoundContestants);**

**console.log(bananaNumbers);**

**let fifthRoundContestants**

**= removeLowestElement(fourthRoundContestants, bananaNumbers);**

**console.log("\n----------------------------------------------------\n");**

**console.log(fifthRoundContestants);**

**console.log("\n----------------------------------------------------\n");**

Marks for:

* Setting up array /3
* Populating array of for each contestant /4
* Form row-sums of the banana array /3
* Finding minimum /2
* Remove associated contestant /2
* Output /1

1. The final challenge to determine who is named Ruler of the Jungle is for the two remaining contestants to challenge each other to a contest of skill and intelligence, known as "rolling dice". Each of the finalists rolls a normal six-sided die and the contestant with the highest score wins. If there is a tie, the process is repeated until a clear winner emerges.

Test this procedure by simulating this contest and output the result, specifying who is to be crowned "Ruler of the Jungle".

(10 marks)

**//1(f)----------------------------------------------------------------**

**var sameScores = true;**

**let scores = [0,0];**

**let winner = "";**

**while(sameScores){**

**round = 1;**

**scores[0] = parseInt(Math.random()\*6 + 1);**

**scores[1] = parseInt(Math.random()\*6 + 1);**

**if (scores[0] > scores[1]){**

**winner = names[0];**

**alert(names[0] + " with " + scores[0] + " beat "**

**+ names[1] + " with " + scores[1]);**

**sameScores = false;**

**} else if (scores[0] < scores[1]){**

**winner = names[1];**

**alert(names[1] + " with " + scores[1] + " beat " + names[0]**

**+ " with " + scores[0]);**

**sameScores = false;**

**} else {**

**alert("No winner on round " + round);**

**round++;**

**}**

**}**

**alert("The Ruler of the Jungle is " + winner);**

**//1(f) Using Functions------------------------------------------------**

**function getWinner(nameArray){**

**var sameScores = true;**

**let scores = [0,0];**

**let theWinner = "";**

**let round = 1;**

**while(sameScores){**

**scores[0] = parseInt(Math.random()\*6 + 1);**

**scores[1] = parseInt(Math.random()\*6 + 1);**

**if (scores[0] > scores[1]){**

**theWinner = nameArray[0];**

**console.log(nameArray[0] + " with " + scores[0] + " beat "**

**+ nameArray[1] + " with " + scores[1]**

**+ " on round " + round);**

**sameScores = false;**

**} else if (scores[0] < scores[1]){**

**theWinner = nameArray[1];**

**console.log(nameArray[1] + " with " + scores[1] + " beat "**

**+ nameArray[0] + " with " + scores[0]**

**+ " on round " + round);**

**sameScores = false;**

**} else {**

**console.log("No winner on round " + round);**

**round++;**

**}**

**} // end while loop**

**return theWinner;**

**} // end function**

**let winner = getWinner(fifthRoundContestants);**

**console.log("\n---------------------------------------------------\n");**

**console.log("The Ruler of the Jungle is " + winner);**

**console.log("\n---------------------------------------------------\n");**

Marks for:

* Set up variables for contest /2
* Generate random scores /2
* Test for winner /2
* Redo if tie /2
* Output result /2

In addition to this: there are:

10 marks for a Statement of Compliance

Marks for:

* SoC covers all parts of question /2
* SoC is accurate /6
* SoC is formatted as required (see P1) /2

10 marks for Coding Style.

Marks for:

* Comments /2
* Coding style, e.g. declaration, semi-colons, etc /2
* Sensible variable names /2
* Sensible function names /2
* Appropriate Indentation /2

Total 80 marks

**Q2.** What does the following code do? Write a paragraph describing what this code does. Do not just make comments in the code. **DO NOT RUN THE CODE**.

What is the output on line 26 if the sentence entered is "Hello Roger"? **(20 marks)**

1. var input = prompt("Enter a sentence");
2. var temp = "";
3. var output = "";
5. for (var i = 0; i < input.length; i++){
6. if (input.charAt(i) === " "){
7. temp += "";
8. } else {
9. temp += input.charAt(i);
10. }
11. }
12. for (var i = 0; i < temp.length; i++){
13. var checked = false;
15. for (var j = 0; j < i; j++){
16. if (temp.charAt(i) === temp.charAt(j)){
17. checked = true;
18. }
19. }
20. if (checked){
21. output += temp.charAt(i).toUpperCase();
22. } else {
23. output += temp.charAt(i).toLowerCase();
24. }
25. }
26. alert(output);

An explanation of the code is as follows:

* The user is prompted for a sentence which is stored in the variable input. **/2**
* Two variables, temp and output and declared and initialised to the empty string. **/2**
* A for-loop is constructed which iterates through the characters of the input variable. **/1**
  + If the character at a particular index is a space, an empty string is added to the temp variable, otherwise the character at that index is added. **/1**
  + The result is that the variable temp is the same as the variable input except with all spaces removed. **/1**
* Another for-loop is set up which iterates over the characters of the variable temp using a counter i. **/1**
  + A boolean flag variable checked is declared and initialised to false. **/1**
  + Another for-loop is nested inside the current for-loop. **/1**
  + This has a count variable which iterates from 0 up to the index of the outer for-loop. **/1**
    - An if-statement checks to see if the character at index i is equal to that at the index j, i.e. if the character at the index i has occurred previously in the string. **/1**
      * If so, the boolean variable checked is set to true. **/1**
    - This check is carried out for all the characters in temp. **/1**
  + The inner for-loop is closed.
  + If checked is true, i.e. if, for a particular character, it has already appeared in the string temp, that character is capitalised and added to the variable output. (Note that small and capitals are treated as different characters) **/2**
  + If not, i.e. if checked is false, the character at i is added to the output variable but is made lower case. **/1**
* The output variable is alerted to the screen. **/1**
* The input "**Hello Roger**" results in the output "**helLorOgEr**" **/2**

**Q3.** Read the following program and rewrite it so that it is easier to understand for a fellow developer. You can run this code to try to work out what it does. **(10 marks)**

let a,b,c=0;d=true;

while(d){a=parseInt(Math.random()\*6)+1

b=parseInt(Math.random()\*6)+1;c++

alert(c+"gives"+a+"and"+b)

if((a+b)===8)d=false}

alert(c)

The code can be rewritten as

var dice1;

var dice2;

var countRolls = 0;

var keepRolling = true;

while(keepRolling){

dice1 = parseInt(Math.random()\*6)+1;

dice2 = parseInt(Math.random()\*6)+1;

countRolls++;

alert("Roll " + countRolls + " gives " + dice1 + " and " + dice2);

if((dice1 + dice2) === 8){

keepRolling = false;

}

}

alert("It took " + countRolls + " rolls to get an 8");

**2** marks for declarations, **2** marks for sensible variable names, **2** marks for use of camelCase, **2** marks for indentation, **2** marks for other formatting (semi-colons, spaces, etc).

**Q4**. Question 4 will be issued on Friday 22nd November and should be submitted by via the Moodle dropbox by 4pm on Tuesday 26th November. **It is worth 10 marks.**

**(RM will mark Q4)**