

Web Programming 1

Web Programming 1 (WEPR1)	
Assignment Number	1
Assignment Name	Formative Assessment
NQF Level	N/A
Credits	N/A
Due Date	
Marks	<p>Total marks = 190</p> <p>Formative assessments through the semester contribute towards the student's module mark and are used to assess progress and identify areas for improvement. This formative assessment will contribute 25% towards final mark.</p> <p>Take note of the following with regards to late submissions:</p> <ul style="list-style-type: none">a. One (1) day late (-5%)b. Two (2) days late (-10%)c. Three (3) days late (-15%)
Individual / Group Assignment	Individual
Lecturer Information	
Lecturer	
Lecturer E-mail	

Learning Objective:

Formative assessment 1 will cover the following concepts:

- a. Javascript with React.JS

Attributes/Competencies Assessed:

Unit standards:

- a. 115367 - Demonstrate logical problem solving and error detection techniques
- b. 115392 - Apply principles of creating computer software by developing a complete programme to meet given business specifications

Scope:

The scope of this formative assessment is based on a solid knowledge of coding techniques used in JavaScript and React.JS supported by theoretical components..

Technical Aspects:

The number of pages for this formative assessment is 15 and the following font and size should be used in your report:

- a. Font: Arial

b. Size: 12 and 14 for headings

c. Font colour: Black

Save and upload the report as a .PDF(**No backgrounds**) with the following naming convention:

a. Student no_StudentName_StudentSurname_ModuleCode_FA1(**Only ZIP folder uploads**)

Ensure adequate referencing is used when using information from either books or internet. Plagiarism is a serious offence and can result in 0% for the assessment when excessive work is copied without proper referencing.

Please complete the following and sign as requested for Portfolio of Evidence (POE)

a. Save code with screenshots of each question and upload when completed

b. Pre-Assessment agreement (Save, sign and submit as PDF)

c. Assessment Feedback Agreement (Save, sign and submit as PDF)

Mark allocation for report

See Mark allocation sheet below

Question 1

(20)

Unit standard	Specific outcome	Assessment criterion
115367	1	1
	1	2
	1	3
	1	4

1. Identify and discuss in short the different problem solving techniques.

(8)

ANSWER:

➤ Top-Down

- **Stepwise design-** works like a staircase with each step taken going down a problem is solved further and further till you reach the baseline or a point of no further problem solving could be (it all starts with a big “picture of the problem” and through the events that picture is broken down to the smallest unit)

➤ Bottom-Up

- Its starting with the physical component (physical things that can be seen) working your way up till you reach the cause of the problem and I would also compare this process to walking up a staircase , with each level depicting a different component of the system and you can only pass to the next level after you have checked that , that level you are on isn't the cause of the problem and you keep on walking up till you reach the level where the problem is .

➤ System Approach

- A technique used to solve problems through formulating steps or processes to be followed namely :
 - I. Defining the problem –get to know the problem
 - II. Developing alternative solutions – finding ways to solve the problem
 - III. Selecting solutions –finding the most effective solution to the problem
 - IV. Designing the solution- creating the solution
 - V. Implementing the solution –inserting or place the solution
 - VI. Reviewing the solution – view whether your solution got an impact or not

2. Identify and discuss situations where specific problem solving techniques would be more suitable than others.

(6)

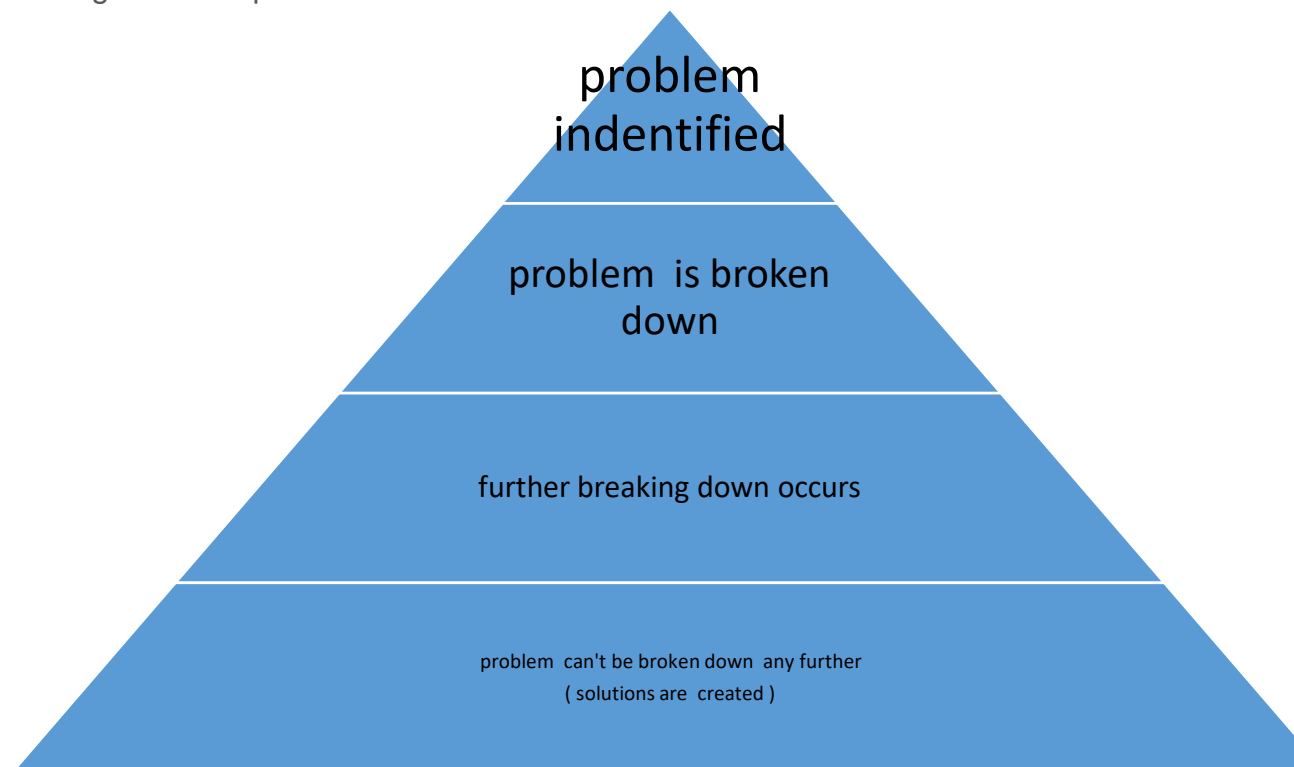
ANSWER:

- **Top-Down** technique would be more suitable for approach for problems being faced by one or a few number of people because the lower level usually affects everyone for example network infrastructure .
- **Bottom-Up** is favored to solve physical problems and more complex problems because this technique check each level or layer of the system thoroughly before moving to the next level or layer .
- **System Approach** –preferred for business domains due to the fact though it looks at the systems as a whole it mainly focus on the components and subsystems that make up the entire system

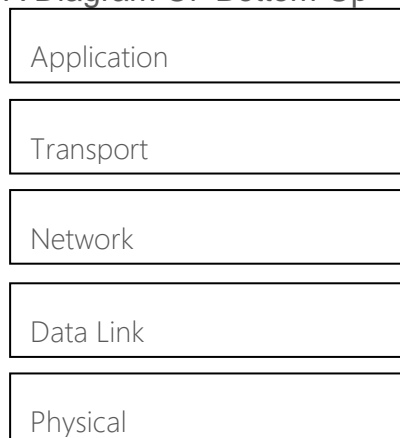
3. Considering question 1 (a) and the problem-solving techniques you have identified, create a visual representation or diagram for each of the problem-solving techniques mentioned. (6)

ANSWER:

I. A Diagram Of Top-Down

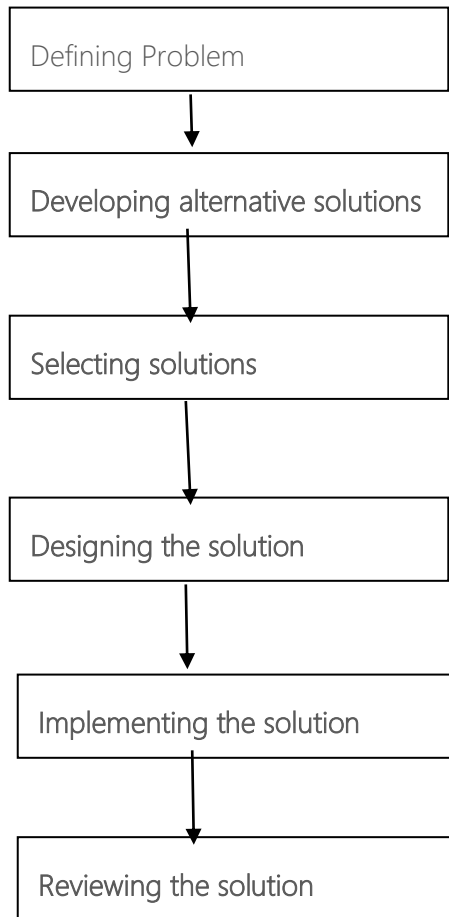


II. A Diagram Of Bottom-Up



Starts here

III. A Diagram Of System Approach



Question 2

(40)

Unit standard	Specific outcome	Assessment criterion
115367	4	1
	4	2
	4	3

1. Identify and discuss the three (3) types of errors.

(17)

ANSWER:

➤ Syntax Errors

- Grammatical errors in programming languages , for example :
 - Misspelled variables
 - Incorrect format in selection and loop statements
 - Mismatching parentheses , curly braces , square brackets

- Missing semicolons
- Runtime Errors
 - Occurs when a program that doesn't have syntax error instruct a computer to carry out tasks, but that computer isn't capable of running them.
 - Examples include : trying to divide a number by zero or trying to open a file that doesn't exist .
- Logic Errors
 - Occurs due to a design flaw ("a design that fails to meet requirements or serve customer needs") in your program .
 - Examples include : add when it should be subtracting or dividing when it should be multiplying and wrong display of information

2. Identify and discuss in short "Error Isolation Techniques". (5)

ANSWER:

- Methods or ways to separate device, component or software module that is causing errors (also known as error diagnosis)
 - This error isolation technique is achieved through building in test circuits that either test through dividing the operations into sub components that can be tested and monitored separately and after this whole process, pieces are put back together again either manually or automatically

3. Identify and discuss the various testing techniques. (18)

ANSWER:

1. Glass Box Testing

- Tests and examines the program's structure and it obtain or get data form program logic or code .
 - It is also name as : open box testing, clear box testing, path driven testing, structural testing or logic driven testing.
 - Glass box testing covers three areas namely :
 - i. Statement coverage –exercises programming statements with minimum tests
 - ii. Branch coverage –runs a series of tests ensuring all the program branches are tested at least once .
 - iii. Path coverage- tests all possible paths

2. White-box Testing

- Tests software and that in turn test the internal structures or how the application works, as opposed to its functionality.
 - Also known as : clear box testing, transparent testing, structural testing and glass box testing .
 - It looks at the internal perspective of the system and programming skills to design test cases .
 - Can be used or applied at the unit, integration and system levels of software testing processes .

3. V-model Testing

➤ It verification and validation model

- Executes processes in a sequence
- It is carried through a number of phases namely:
 - i. Requirements – system test plan is created (test plan focuses on functionality of the system)
 - ii. The High Level Design (HLD)-mainly focuses on the systems' design and architecture . An integration testing is also made in this phase
 - iii. The Low Level Design (LLD)- where the actual software components are created . In this Phase , the logic of each and every systems' component is defined and also component test are made in this phase .
 - iv. The implementation phase- where all coding occurs .
 - v. Coding – module design is coded by a programmer and unit testing occurs here .

Question 3

(66)

Unit standard	Specific outcome	Assessment criterion
115392	1	1
	1	2
	1	3
	1	4
	2	1
	2	2

1. Tic-tac-toe is a classic game played on a 3x3 grid. The objective is to be the first to form a line of three of your symbols (X or O) either horizontally, vertically, or diagonally. (50)
 - a. Here's a step-by-step description of how the game works:
 - i. The game begins with an empty 3x3 grid.
 - ii. Two players take turns, one using X and the other using O.
 - iii. A player selects an empty cell on their turn and places their symbol (X or O) in that cell.
 - iv. The turn then passes to the other player.
 - v. Steps iii and iv are repeated until a player wins or the game ends in a draw.
 - vi. A player wins if they have three of their symbols in a row, column, or diagonal.
 - vii. If all cells are filled and no player has won, the game ends in a draw.

- b. Write the game using JavaScript and React.JS. (Paste your code below and also upload the working file that contains your code.

ANSWER:

```
JS Appjs X # styles.css index.html JS indexjs
src > JS Appjs > Board
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { useState } from "react"; 4.1k (gzipped: 1.8k)
3
4 function Grid(value, onGridClick){
5
6   return <button className="grid" onClick={onGridClick}>{value}</button>
7 };
8 }
9
10 export default function Board() {
11   const [player1IsNext, setPlayer1IsNext] = useState(true);
12   const [grid, setGrid] = useState(Array(9).fill(null));
13
14   function handleClick(i) {
15     if (grid[i] || calculateWinner(grid)) {
16       return;
17     }
18     const player2Grid = grid.slice();
19     if (player1IsNext) {
20       player2Grid[i] = "X";
21     }
22     else {
23       player2Grid[i] = "O";
24     }
25   }
26   setGrid(player2Grid);
27   setPlayer1IsNext(!player1IsNext);
28 }
29 const winner = calculateWinner(grid);
30 let status;
31 if (winner) {
32   status = 'well done ' + ' ' + 'player' + winner + ' ' + 'you have won';
33 }
34 else {
35   status = 'Next player: ' + (player1IsNext ? 'player1': 'player2');
36 }
37 return(
38   <React.Fragment>
39     <div className="status">{status}</div>
40     <div class="board">
41       <div className ="container">
42         <Grid value={grid[0]} onGridClick={() => handleClick(0)} />
43         <Grid value={grid[1]} onGridClick={() => handleClick(1)} />
44         <Grid value={grid[2]} onGridClick={() => handleClick(2)} />
45       </div>
46     </div>
47     <div className ="container">
48       <Grid value={grid[3]} onGridClick={() => handleClick(3)} />
49       <Grid value={grid[4]} onGridClick={() => handleClick(4)} />
50       <Grid value={grid[5]} onGridClick={() => handleClick(5)} />
51     </div>
52     <div className ="container">
53       <Grid value={grid[6]} onGridClick={() => handleClick(6)} />
54       <Grid value={grid[7]} onGridClick={() => handleClick(7)} />
55       <Grid value={grid[8]} onGridClick={() => handleClick(8)} />
56     </div>
57   </React.Fragment>
58 );
59 }
60
61 function calculateWinner(grid){
62   const lines=[
63     [0, 1, 2],
64     [3, 4, 5],
65     [6, 7, 8],
66     [0, 3, 6],
67     [1, 4, 7],
68     [2, 5, 8],
69     [0, 4, 8],
```



```
JS App.js x # styles.css index.html JS index.js
src > JS App.js > Board
70 [2, 4, 6]
71 ];
72
73 for (let i = 0; i < lines.length; i++) {
74   const [a, b, c] = lines[i];
75   if (grid[a] && grid[a] === grid[b] && grid[a] === grid[c]) {
76     return grid[a];
77   }
78 }
79 return null;
80 }
```

```
JS App.js # styles.css x index.html JS index.js
src > # styles.css > code
1 * {
2   box-sizing: border-box;
3 }
4
5 body {
6   font-family: sans-serif;
7   margin: 20px;
8   padding: 0;
9 }
10
11
12 h1 {
13   margin-top: 0;
14   font-size: 22px;
15   color: coral;
16   font-family: Georgia, 'Times New Roman', Times, serif;
17 }
18
19
20
21 code {
22   font-size: 1.2em;
23 }
24
```

```
JS App.js # styles.css x index.html JS index.js
src > # styles.css > code
25 * {
26   box-sizing: border-box;
27 }
28
29 body {
30   font-family: sans-serif;
31   margin: 20px;
32   padding: 0;
33   color: rgb(12, 58, 227);
34 }
35
36 .grid {
37   background: #4fe2e5;
38   border: 1px solid #000000;
39   float: left;
40   font-size: 40px;
41   font-weight: bold;
42   line-height: 34px;
43   height: 90px;
44   margin-right: -1px;
45   margin-top: -1px;
46   padding: 0;
47   text-align: center;
```

```

JS App.js # styles.css X index.html JS index.js
src > # styles.css > code
48 width: 90px;
49 border-radius: 5px solid black;
50 color: whitesmoke;
51 }
52
53 .container{
54 margin: 0px;
55 border-radius: 5px;
56 }
57
58
59
60 .container:after {
61 clear: both;
62 content: '';
63 display: table;
64 }
65
66 .status {
67 margin-bottom: 10px;
68 font-size: 30px;
69 color: rgb(50, 204, 137);
70 font-family: Georgia, 'Times New Roman', Times, serif;
71 }

```

```

JS App.js # styles.css X index.html JS index.js
src > # styles.css > code
72 .game {
73 display: flex;
74 flex-direction: row;
75 }
76
77 .game-info {
78 margin-left: 20px;
79 }

```

```

JS App.js # styles.css X index.html X index.js
public > index.html > @html > @body > @h1
1 <doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6 <meta name="theme-color" content="#000000">
7 <!--
8 manifest.json provides metadata used when your web app is added to the
9 homescreen on Android. See https://developer.google.com/web/fundamentals/engage-and-retain/web-app-manifest/
10 -->
11 <link rel="manifest" href="/PUBLIC_URL/manifest.json">
12 <link rel="shortcut icon" href="/PUBLIC_URL/favicon.ico">
13 <!--
14 Notice the use of /PUBLIC_URL in the tags above.
15 It will be replaced with the URL of the public folder during the build.
16 Only files inside the public folder can be referenced from the HTML.
17 Unlike "/favicon.ico" or "favicon.ico", "/PUBLIC_URL/favicon.ico" will
18 work correctly both with client-side routing and a non-root public URL.
19 Learn how to configure a non-root public URL by running 'npm run build'.
20 -->
21 <title>React App</title>
22 </head>
23 <body>
24 </body>
25 </html>

```

```

JS App.js # styles.css X index.html X index.js
public > index.html > @html > @body > @h1
24 </head>
25 <body>
26 <div style="text-align: center;">Tic-Tac-Toe</div>
27 <div id="root"></div>
28 <!--
29 This HTML file is a template.
30 If you open it directly in the browser, you will see an empty page.
31
32 You can add webfonts, meta tags, or analytics to this file.
33 The build step will place the bundled scripts into the <body> tag.
34
35 To begin the development, run 'npm start' or 'yarn start'.
36 To create a production bundle, use 'npm run build' or 'yarn build'.
37 -->
38 </body>
39 </html>
40

```

```

JS App.js # styles.css X index.html JS index.js X
src > JS index.js > ...
1 import React, { StrictMode } from "react"; 6.9k (gzipped: 2.7k)
2 import { createRoot } from "react-dom/client"; 505 (gzipped: 323)
3 import "./styles.css";
4
5 import App from "./App";
6
7 const root = createRoot(document.getElementById("root"));
8 root.render(
9   <StrictMode>
10     <App />
11   </StrictMode>
12 );

```

2. Do white-box testing for the above solution.

(16)

ANSWER:

1) Unit testing

- Tests units of a program for example for a tic-tac-toe game unit testing will test if a player meet the winning conditions or not through constantly checking the handleClick function

2) code coverage

- covers all the critical parts of a code for example it make sure that inputs are inserted in the expected position and also tests the Boolean statement in a code and ensures the program is running smoothly .

3) component Testing

- tests different components of a program : test 3x3 grid , test result display ,test the restart button

Question 4

(64)

<i>Unit standard</i>	<i>Specific outcome</i>	<i>Assessment criterion</i>
115392	2	3
	2	4
	3	1
	3	2
	3	3
	4	1
	4	2
	4	3
	5	1
	5	2
	5	3

	6	1
	6	2
	6	3
	6	4

1. Write the pseudocode for the Tic-Tac-Toe game:

(23)

ANSWER:

totPlayerX input=0

totplayerO input=0

begin

while playerX's or playerO's input != gridNumber [123,456,789 ; 147,258,369 ; 159,357]

for totInputNumber=9

totInputNumber= totplayerX input + totplayerO input

playerX input= X

print= "playerO's turn"

playerO input=O

print= "playerX's turn"

endfor

endwhile

if playerX's or PlayerO's input= gridNumber

gridNumber=[123,456,789 ; 147,258,369 ; 159,357] then

print="Win"(to the player matching the gridNumber)

endif

if playerX's or playerO's input !=gridNumber then

gridNumber=[123,456,789 ; 147,258,369 ; 159,357]

print="Tie" (to both playerX and playerO)

endif

if playerX's print="win"

print="playerO lost" (to playerO)

elseif playerO's print="win"

print="playerX lost" (to playerX)

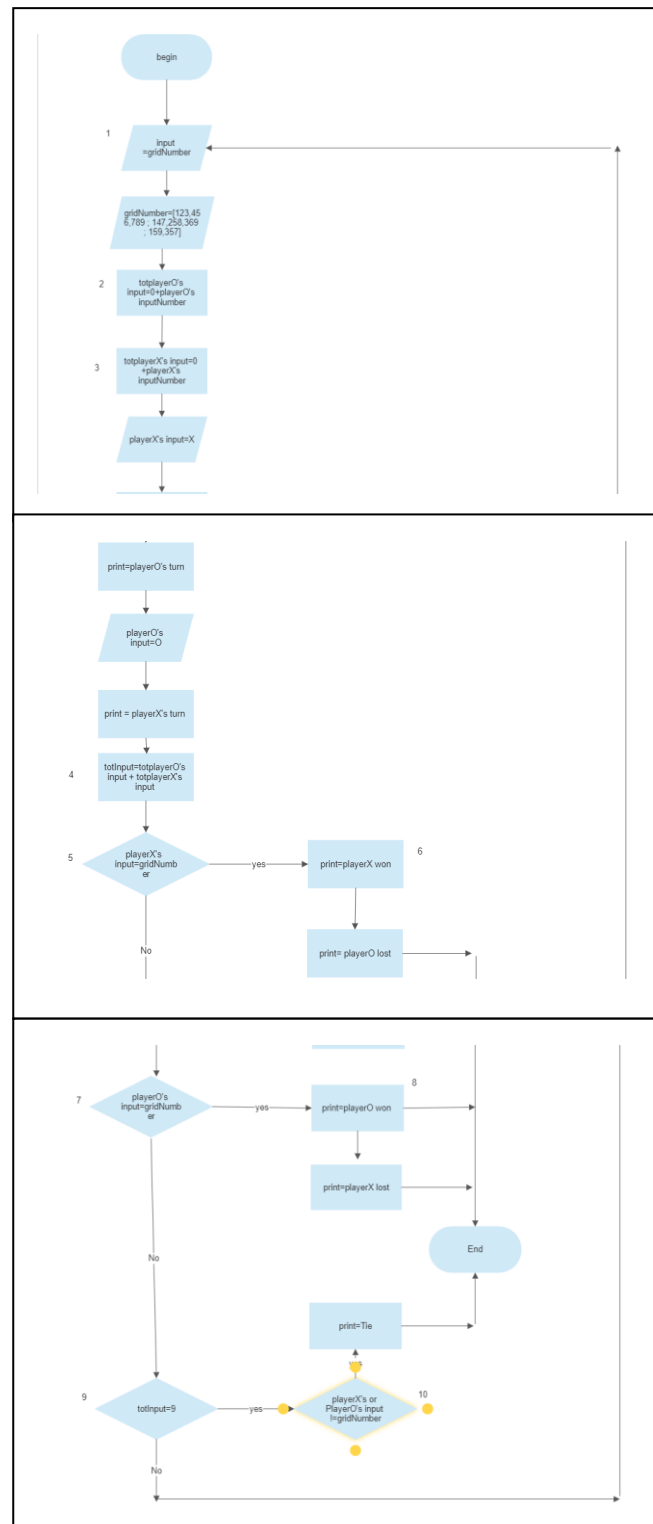
endif

end

2. Draw a program flowchart for the above pseudocode.

(18)

ANSWER:



3. As part of your testing, create a trace table from your program flowchart.

(13)

ANSWER:

Below is a trace table of a Tic Tac Toe game

moves	Input=gridNumber	totX's input	totO's input	totInput=totO's input +totX's input	playerX=gridNumber	playerO=gridNumber	totInput=9	playerX playerO !=gridNumber	or	results
1	X									
2			0							
3		1								
4				1						
5					no					
7						no				
9							no			
1	X									
2			4							
3		4								
4				8						
5					yes					
6										Print=playerX won Print=playerO lost
1	O									
2			3							
3		4								
4				7						
5					no					
7						yes				
8										Print=playerO won Print=playerX lost
1	O									
2			4							
3		5								
4				9						
5					no					

7					no				
9							yes		
10									Print=Tie

4. Create a short testing plan on how the users must use your program with appropriate screenshots for the user. (10)

ANSWER:

- Output expected results
 - Input =output (user's input must equal their output)
- Output evaluation
 - If a user's moves meet the winning conditions game must stop and display the winner
- Responsive and feasibility
 - Program must run on all devices and convenient to use(must be simple to use)
- User training
 - Users must be introduced and educated about the program before using it

Referencing:

<https://reqtest.com/en/knowledgebase/white-box-testing-example/>

<https://www.youtube.com/watch?v=c8dXnuVwmA8>

<https://www.youtube.com/watch?v=EEWaLOvCprk&pp=ygUZdGljlHRhYyB0b2Ugd2l0aCBzZWJdCBqcw%3D%3D>

<https://www.w3schools.com/react/default.asp>

Mark allocation for student			
Section	Sub-section	Maximum Mark	Learner mark
Body of the report	Question 1.1	8	
	Question 1.2	6	
	Question 1.3	6	
	Question 2.1	17	
	Question 2.2	5	
	Question 2.3	18	
	Question 3.1	50	
	Question 3.2	16	
	Question 4.1	23	
	Question 4.2	18	
	Question 4.3	13	
	Question 4.4	10	
Deductions	1 day late	-5	
	2 days late	-10	
	3 days late	-15	
Total:		190	

PRE-ASSESSMENT AGREEMENT

Assessment Preparation: Preparing the Candidate

Student name and surname	Tafadzwa Chiripanyanga	Date	7/01/2023
		Time	4pm
Assessor name and surname		Venue	online
How to prepare the candidate	Document Requirements	Agree (tick)	Action Required
Explain to the candidate why you are meeting and the purpose of the assessment.	Assessment Policy Assessment process	✓	
Discuss the assessment plan in detail.	Assessment strategy	✓	
Explain assessment process, show assessment instruments to candidate and describe assessment conditions.	Assessment instruments	✓	
Identify the role-players during assessment.	Assessors Moderator	✓	
Describe the evidence required to be declared competent.	Examples of evidence	✓	
Explain how evidence will be judged.	Mark allocation explained	✓	
Explain to the candidate how to prepare: Give candidate assessment task description.	Assessment task description	✓	
Confirm with the candidate what he/she should bring to the assessment.	Detailed briefing on exact requirements to be given to candidate in writing	✓	
Ensure that candidate understands the procedures of all assessment practices.	Appeals Policy Appeals procedure Assessment Policy Assessment Procedure Moderation Policy Moderation procedure Verification Policy Verification Procedure	✓	

Ask the candidate if he/she foresees any problems or identify any special needs.	List needs	✓	
--	------------	---	--

Agreed Assessment Plan			
Student name and surname:		Tafadzwa Chiripanyanga	
Assessor name and surname:			
Module name:		Web Programming 1	
Unit Standard/s:		US115367 US115392	
Type of Assessment i.e. Formative assignment, Formative test, Formative Practical, Summative etc.		Formative Assessment 1	
Special Assessment Requirements:		N/A	
Event	Date, time and location	Resources required	Evidence to be generated
Assessments due date		Assessments	Completed documentation
Complete activity on MyAIE and upload to MyAIE			Completed Portfolio of Evidence
Submit Portfolio of Evidence			

Assessor Roles and Responsibility	
Roles	Assessor Guide Feedback Agent Reviewer
Responsibilities	Consult candidate re-assessment, assessment process and plan. Agree assessment process and plan with candidate. Forward documentation to candidate: plan, guide and assessment instruments. Assess candidate with the use of different instruments. Provide feedback on assessment findings. Support candidate through assessment process. Source feedback from candidate on assessment process.

	<p>Review assessment process and outcome.</p> <p>Use assessment process as opportunity to transform assessment activities and outcomes.</p>
--	---

Candidate Roles and Responsibility	
Roles	<ul style="list-style-type: none"> • Learner • Feedback agent • Reviewer
Responsibilities	<ul style="list-style-type: none"> • Be available for assessment. • Be actively involved in the consultative process. • Learn from the assessment process. • Provide feedback to the assessor in terms of the assessment as learning activity. • Provide feedback to the assessor on the efficacy of the assessment process. • Review own role and assessor role in the assessment process.
Assessment Instruments	<ul style="list-style-type: none"> • Portfolio of Evidence • Questionnaire • Report • Presentation • Reflexive questions • Work sample • Practical's • Group Activity • Research activities

Assessment Process	
<ul style="list-style-type: none"> • Evaluation of POE addressing Essential Embedded Knowledge in unit standards. • Evaluation of Research Projects and other evidence addressing specific unit standards. • Consultation: assessment plan and assessment activities and instruments. Pre-assessment moderation and interviews conducted at this stage. • Observation: feedback on assessment against specific outcomes, critical outcomes in unit standards. • Feedback: to candidate regarding sufficiency of evidence and possible interview to gain supplementary evidence. • Feedback to candidate regarding assessment findings as well as review process. 	
Feedback	Written feedback to be given to all stakeholders at the end of the assessment process, as well as verbal feedback to the candidate

	during assessment activities.
Recording Process	Process and findings to be recorded and submitted for record keeping purposes as well as moderation and verification.
Review Process	The review process is the responsibility of the assessor and the candidate. Joint reviewing will take place after feedback has been given to the candidate.
Right to appeal	The candidate must be advised of the right to appeal.
Resources Required	Assignments <ul style="list-style-type: none"> • POE • Assessments • Guides
I confirm that: <ul style="list-style-type: none"> • I have been consulted on and have agreed to the training and assessment process as detailed in the assessment guide. • I have been advised of my right to appeal against any assessment that is unfair, unreliable, invalid or impracticable. • I have read and understood the appeal procedure. • I know that assessments may be moderated or verified by an external party. • The purpose of the assessment has been clearly explained to me. • The criteria have been discussed with me, and I know I will be assessed against these criteria. • I know when and where I will be assessed, and I was given fair notice. • I know how the assessment will be done, and any other requirements related to the assessment. 	

Signed: tafadzwa

Date: 7/01/2023

Overall Assessment Decision	Competent	Not yet competent
Student's Signature	<i>tafadzwa</i>	Date: 7/01/2023
Assessor's Signature		Date:
Moderator's Signature		Date:

ASSESSMENT FEEDBACK AGREEMENT

Assessment feedback: Feedback to learner

Qualification Name:	
Qualification SAQA Number:	
Subject Name:	Web Programming 1
Subject Code:	WEPR1
Assessment Name:	Formative Assessment 1
Assessment Code:	WEPR1_FA1
Assessment Type:	Formative

Feedback report	1st Attempt		2nd Attempt	
	C	NYC	C	NYC
Unit standard Number(s)				
US115367:				
SO1;AC1				
SO1;AC2				
SO1;AC3				
SO1;AC4				
SO4;AC1				
SO4;AC2				
SO4;AC3				
US115392:				
SO1;AC1				

SO1;AC2				
SO1;AC3				
SO1;AC4				
SO2;AC1				
SO2;AC2				
SO2;AC3				
SO2;AC4				
SO3;AC1				
SO3;AC2				
SO3;AC3				
SO4;AC1				
SO4;AC2				
SO4;AC3				
SO5;AC1				
SO5;AC2				
SO5;AC3				
SO6;AC1				
SO6;AC2				
SO6;AC3				
SO6;AC4				

General feedback to learner (Attempt 1)

Supply comprehensive feedback why learner is found NYC

Learner Number:	258196		
Learner name and surname:	Tafadzwa Chiripanyanga	Date:	7/01/2023
Learner Signature:	<i>tafadzwa</i>		
Lecturer name and surname:		Date:	
Lecturer Signature:			
Assessor name and surname:		Date:	
Assessor Signature:			
Moderator name and surname:		Date:	
Moderator Signature:			

Note to learner
<p>Review the feedback provided by your lecturer to check that you have been found competent in this assessment. If there are any areas where you have been found not yet competent, you must redo those parts of the assessment and resubmit within the stipulated time frame.</p> <p>The section below will only be completed in cases where the learner was asked to resubmit parts of the assessment where they were found not yet competent.</p>

General feedback to learner (Attempt 2)
Supply comprehensive feedback why learner is found NYC

Learner Number:			
Learner name and surname:		Date:	
Learner Signature:			
Lecturer name and surname:		Date:	
Lecturer Signature:			
Assessor name and surname:		Date:	
Assessor Signature:			
Moderator name and surname:		Date:	
Moderator Signature:			