

Web Programming 1

Web Programming 1(WEPR1)		
Test Number	1	
Year	2023	
Internal Examiner	David Fourie	
Total Marks	120	
Release Date and time	07 July 2023 @08:00	
Submission Date and time	14 July 2023 @23:45	
Section A	Practical	100 Marks
	Comments	10 Marks
	Code executes successfully	10 Marks

Instructions to Candidates:

1. Read carefully what needs to be done in this practical assessment.

Important Notes:

- This test paper consists of 13 pages
- This test contributes 50% towards the final mark.

Section A

(100)

US115367
US115392

You are the new JAVASCRIPT programmer at a startup company called "New Future pty (LTD)".

The senior project manager and business analyst approached you to fill a gap identified and write a Java program that implements a task management application. The application should allow users to perform the following actions:

- Add a task with a title and description.
- Mark a task as completed by providing its ID.
- List all tasks with their IDs, titles, and completion status.
- Delete a task by providing its ID.
- Exit the application.

You should create two classes: Task and TaskManager. The Task class should have properties for ID, title, description, and completion status. The TaskManager class should manage a collection of tasks and provide methods for adding, marking, listing, and deleting tasks.

Implement the application by creating a function runTaskManager() that handles user input and performs the corresponding actions based on their choices. The application should run in a command-line interface (CLI) and use the readline module for user input.

Write the complete JavaScript code for the task management application, including the 'Task' and 'TaskManager' classes, the 'runTaskManager()' function, and the necessary input/output handling. Make sure to include comments to explain the major steps and logic in your code.

Test the application by adding tasks, marking tasks as completed, listing all tasks, and deleting tasks. Handle any potential errors or invalid user input gracefully.

Provide the code along with the comments explaining the functionality and logic used in your implementation.

Output should be as follow:

Step 1 (Add a task)

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 1
Enter task title: New task
Enter task description: This is the new task
Task added successfully.

Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice:
```

You should be able to add multiple tasks

Step 2 (Mark a task as completed)

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 2
Enter task ID to mark as completed: 1
Task marked as completed.

Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice:
```

There should be an X next to the task indicating that the task was completed as seen in the next step when list tasks is selected.

Step 3 (List tasks)

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 3
Tasks:
[ ] 1: ABC
[X] 2: DEF
[X] 3: New task
```

If there is no tasks the following should display:

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 3
No tasks found.
```

Step 4 (Delete a task)

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 4
Enter task ID to delete: 1
Task deleted successfully.
```

If there is no tasks the following should display:

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 4
Enter task ID to delete: 4
Task not found.
```

Step 5 (Exit)

```
Task Management Application
-----
1. Add a task
2. Mark a task as completed
3. List tasks
4. Delete a task
5. Exit
Enter your choice: 5
PS C:\David\Modules\Software Programming 1\Summative Assessment\Project Solution>
```

Copy and paste your code here (Please upload the actual code in notepad as well when uploading this document as evidence):

```

let readline = require('readline'); //imports the readline module

let readLine = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const taskArray = []; //Array to store tasks
let taskIdNumber = 1; //initial number for the task id number

function createTaskId() {
  return taskIdNumber++; //increments the task id number from the previous id number
}

function displayOptions() { //function to display options
  console.log('\n Task Manager Application\n');
  console.log('1. Add a Task');
  console.log('2. Mark a Task as Completed');
  console.log('3. List All Tasks');
  console.log('4. Delete a Task');
  console.log('5. Exit');
}

function addTask() { //function for adding Tasks into the array
  readline.question('Enter task title: ', title => {
    readline.question('Enter task description: ', description => {
      let taskId = createTaskId();
      taskArray.push({ id: taskId, title, description, completed: false });
      console.log(`Task "${title}" added with ID: ${taskId}`);
      displayOptions();
      userInput();
    });
  });
}

function markTaskAsCompleted() { //this function marks task as completed or not
  readline.question('Enter the task ID to mark as completed: ', taskId => {
    taskId = Number(taskId);
    const task = taskArray.find(task => task.id === taskId);
    if (task) {
      task.completed = true;
      console.log(`Task "${task.title}" marked as completed.`);
    } else {
      console.log('Task not found.');
```

```

    userInput();
  });
}

function userInput() { //function that checks user's inputs
  readline.question('Enter your choice: ', option => {
    switch (option) {
      case '1':
        addTask();
        return;
      case '2':
        markTaskAsCompleted();
        return;
      case '3':
        listTasks();
        return;
      case '4':
        deleteTask();
        return;
      case '5':
        console.log('Exiting Task Management Application. ');
        readline.close();
        return;
      default:
        console.log('Invalid choice. Please enter a valid option. ');
        displayMenu();
        handleUserInput();
        return;
    }
  });
}

displayOptions();
userInput();

```

SUBTOTAL: [100]

TOTAL: [100]

Mark allocation for student			
Section	Sub-section	Maximum Mark	Learner mark
Section A	Question 1	100	
	Code comments used	10	
	Code executes successfully (No errors shown)	10	
Total:		120	

PRE-ASSESSMENT AGREEMENT

Assessment Preparation: Preparing the Candidate

Student name and surname	Tafadzwa Chiripanyanga	Date	17/09/2023
		Time	16:00
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.		Summative 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.

	<p>Source feedback from candidate on assessment process.</p> <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 17/09/2023

Overall Assessment Decision	Competent	Not yet competent
Student's Signature	Tafadzwa	Date: 17/09/2023
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:	Summative Assessment 1
Assessment Code:	WEPR1_SA1
Assessment Type:	Summative

Feedback report	1st Attempt		2nd Attempt	
	C	NYC	C	NYC
Unit standard Number(s)				
US115367				
US115392				

General feedback to learner (Attempt 1)
Supply comprehensive feedback why learner is found NYC

Learner Number:	258196		
Learner name and surname:	Tafadzwa Chiripanyanga	Date:	17/09/2023
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:	258196		
Learner name and surname:	Tafadzwa Chiripanyanga	Date:	17/09/2023
Learner Signature:			
Lecturer name and surname:		Date:	
Lecturer Signature:			
Assessor name and surname:		Date:	
Assessor Signature:			
Moderator name and surname:		Date:	
Moderator Signature:			