Westlake Boys High School

Assessment

Digital Technologies

AS91896v1 Level 2 Credits 6

Use advanced programming techniques to develop a computer program.

Student Name:

Grade Awarded:

Not Achieved	Achieved	Merit	Excellence

Overall comments:

Teacher Name: _____

Signature: _____

Date: ____/____/2025

The work that you submit for this assessment needs to be your **own**.
You may not use AI to generate the entire program.
See the NEXT page for explicit information

Assessment Task

Important: Read through **all** the instructions in this task before you start working on it.

In this assessment, you need to develop a computer program by using advanced programming techniques.

What you will be assessed on:

You will be assessed on how well you do the following.

- Write code for the program to allow it to perform the specified task Use advanced techniques to develop the program.
- Set out the program code.
- Document the program.
- Test and debug the program.

Use of AI in this assessment:

DO:	DON'T
You may use AI tools to clarify programming concepts, syntax, or debugging approaches.	Do not use AI to generate the entire program or portions of code.
If you encounter an error you can't resolve, AI tools can help explain what the error message means or suggest ways to fix it. *	Do not use AI tools to debug your program without referencing the tool used/outcome in your testing documentation.
You can use AI to brainstorm potential test cases or scenarios to ensure your program meets the brief.	Do not use AI to bypass the learning process or tasks intended to assess your skills and knowledge.
You can use AI as a 'Peer review' tool, asking it to check your conventions of the language. *	

Important Note:

You must be able to explain every line of code in your program. Submitting your assessment without understanding of the program will bring the authenticity of your work into question.

* *If you use AI in this manner, it must be documented in your testing documentation.*

Scenario:

You must develop a **Task Management System** that allows a small team to track tasks, team members, and task assignments.

The program should have a dictionary to store task information. Each task should have an ID, a title, a description, an assignee (team member), a priority rating (1-3), and a status. The program should have an additional dictionary to store team member information. Each team member should have a unique identifier, a name, an email, and a list of assigned tasks.

The program needs to be able to:

- **Add a new task to the project's task list:**
  - Each task needs to have an ID, a title, a description, an assignee (team member), a priority rating (1-3), and a status.
  - The ID must be sequential and automatically assigned to the task (The user will not add this)
    - **NOTE**: Tasks do not need to be assigned to a team member straight away

- **Update the task including:**
  - Task status (e.g., in progress, completed, blocked, not started)
  - Assign to a team member – This needs to update the task dictionary and the team member task list.
    - **NOTE:** When the task is completed then it should be removed from the team member's task list.

- **Search for a task or team member. The user should be given an option of which they would like to search for:**
  - Tasks can be displayed by their title and then when chosen, the program outputs the details of the task.
  - Users will be able to search for a team member. The program should then output the team member's details and task list.

- **Generate a report of the project's progress, including the number of tasks completed, the number of tasks in progress, the number of tasks blocked, and the number of tasks not started.**

- **Output the task collection in a readable format.**

The program should use the Easygui library as the chosen GUI.

See below for the information that needs to be used to populate your dictionaries.

Task Dictionary:

TASK ID	TITLE	DESCRIPTION	ASSIGNEE	PRIORITY	STATUS
T1	Design Homepage	Create a mockup of the homepage	JSM	3	In Progress
T2	Implement Login page	Create the Login page for the website	JSM	3	Blocked
T3	Fix navigation menu	Fix the navigation menu to be more user-friendly	None	1	Not Started
T4	Add payment processing	Implement payment processing for the website	JLO	2	In Progress
T5	Create an About Us page	Create a page with information about the company	BDI	1	Blocked

Team Member Dictionary:

MEMBER ID	NAME	EMAIL	TASKS ASSIGNED
JSM	John Smith	John@techvision.com	[“T1”,” T2”]
JLO	Jane Love	Jane@techvision.com	[“T4”]
BDI	Bob Dillon	Bob@techvision.com	[“T5””]

Assessment Schedule

AS91896v1 Level 2 Credits 6

Use advanced programming techniques to develop a computer program.

Evidence/Judgements for Achievement	Evidence/Judgements for Achievement with Merit	Evidence/Judgements for Achievement with Excellence
The learner used advanced programming techniques to develop a computer program. This included doing the following.	The learner used advanced programming techniques to develop an ***informed*** computer program. This included doing the following.	The learner used advanced programming techniques to develop a ***refined*** computer program. This included doing the following.
☐ **Writing code for a program that performs the task specified in the supplied brief.** *Example*: The learner has written a program that can perform the task specified in the brief. The program: - is written in a procedural manner, consisting of sequence, selection, and iteration control structures, and gets inputs from a user - uses variables storing different types of data (string e.g. the tasks title; numeric – e.g. variables storing several tasks for each status) - produces output (e.g. output of the full task list or individual team member details	**As for Achievement**	☐ **Ensuring the program is flexible and robust.** *Example*: - The program has a flexible structure using data stored in nested dictionaries and lists. Functions, conditions, and control structures are used effectively, with redundant code or unnecessary repetition avoided. - The program includes input validity checks to ensure the program correctly handles expected, boundary, and invalid values. This could include presence checks (to check the user has input data), range checks (to check the input falls within a specified/expected range), and data types checks (e.g. to check if the user has pressed the 'Cancel'

- uses the following advanced programming techniques:
  o modifying data stored in a list and/or dictionary (e.g. deleting tasks from a member's task list, changing status, etc.)
  o creating functions that use parameters and/or return values
  o using a GUI that allows the user to interact with the program by inputting values, clicking a button, etc.
 	
There may be unnecessary or redundant code.
 	
NOTES/EVIDENCE:
		button, with None returned).
		
		Where possible, the program uses constants, variables, and derived values, rather than literals. These are set out at the start of the program or relevant functions so that they can be easily found and changed if necessary.
		
		NOTES/EVIDENCE:

☐ **Setting out the program code clearly.** *Example*: - The code is indented as appropriate and blank lines are used between blocks of related code and functions. NOTES/EVIDENCE:	☐ **Ensuring that conventions for the programming language are followed.** *Example*: The learner has followed most conventions for the language used for the program (Python). *For example:* - Variable names are consistently styled (e.g. lowercase with underscores). - Line length is kept to a maximum of 79 characters for regular code, and 72 characters for block comments and docstrings. - Block comments are used for blocks of code, rather than an inline comment on the first line of the block. - Block comments are indented to the same level as the block of code they describe. - Comments are written as full sentences beginning with a capital letter. - Docstrings are included for each function. These are added after the function definition, are indented, and clearly describe the purpose of the function. NOTES/EVIDENCE:	☐ **Ensuring the program is a well-structured, logical response to the task.** *Example*: The program is concise and easily readable, without redundant code. - Functions are used rather than duplicating code. - The learner used a for loop to iterate over a list of categories rather than duplicating code for each category. NOTES/EVIDENCE:

☐ **Documenting the program with comments.**
 	
Example:
 	
The learner has included some comments. However, these are not always meaningful and descriptive. Some comments are added to the code that is self-explanatory.
For example:
`name = ""  # name set to empty string`
 	
NOTES/EVIDENCE:
	☐ **Setting out the program code clearly.**
	
	Example:
	
	The learner has used meaningful variable names which clearly indicate the data held and function names are relevant and appropriate to the purpose of the function. The comments in the program are clear and meaningful. These comments help to describe the purpose (function) of the code and the code's behaviour.
	For example:
	`# Calls function to validate name input.`
	`name = validate_name(name)`
	
	NOTES/EVIDENCE:	

☐ **Testing and debugging the program to ensure it works on a sample of expected cases.** *Example*: The learner has provided evidence that they have tested the program using expected user input. This was not done systematically, and the recording of this testing was minimal NOTES/EVIDENCE:	☐ **Testing and debugging the program in an effective manner to ensure it works on a sample of expected and relevant boundary cases.** *Example*: The learner has used a test plan or some other appropriate method(s) to plan their testing beforehand and to record the results of this testing. The testing is done systematically, testing one block of related code (e.g. a function) for expected and boundary cases. NOTES/EVIDENCE:	☐ **Testing and debugging the program comprehensively to ensure it works on expected, boundary, and invalid cases.** *Example*: The learner has systematically tested all input cases, including expected, boundary, and invalid cases. They have methodically conducted and recorded their testing and results. NOTES/EVIDENCE: