

ASSESSMENT

CREATE A COMPUTER PROGRAM

AS92004v2

Level 1

Credits 5

STUDENT TO FILL OUT

First and last name:

NSN number:

Date:

Signature:

ASSESSMENT RESULT (tick one)

Achieved ☐ Merit ☐ Excellence ☐ Not Achieved ☐

Resubmission requirements (if any)

Resubmission Date:

ASSESSOR NOTE:

Ensure you read all the information in this assessment before you give it to students. The Assessment Schedule and import assessor information are found at the end of this assessment.

STUDENT STATEMENT

What you need to know

- There is one assessment task that consists of several items.
- You can use your classroom notes or any other relevant resources to help you complete this assessment.
- Make sure your answers are your own work.

STUDENT TO FILL OUT

- I have read and understood the 'What you need to know' above.
- My assessor has answered any questions I have about the assessment.
- I know about any requirements, such as health and safety, that I must follow.
- I completed all work for this assessment myself. I did not get help from anyone else.
- I know how to ask for a review of my assessment result.

Student Signature

Date

ASSESSMENT TASK

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

In this assessment, you need to create a computer program. Your assessor will give you the brief for the computer program they want you to develop.

What you will be assessed on

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

- Write code for the program using a suitable programming language to allow it to perform the specified task
- Document the program (variable names and comments)
- Test and debug the program

Use the checklist below to ensure you meet all the requirements of the assessment. As you complete an item, tick it off in the 'Candidate to tick' column.

Item	Candidate to tick	Assessor to tick
1. Write code for a program to perform the task specified in the brief. The computer program needs to include the following.		
<ul style="list-style-type: none">• Variables storing at least two different types of data (eg numeric, string/text, Boolean).		
<ul style="list-style-type: none">• All of the following control structures:<ul style="list-style-type: none"><input type="checkbox"/> Sequence<input type="checkbox"/> Selection<input type="checkbox"/> Iteration (loops)		
<ul style="list-style-type: none">• Input from a user, sensor, or other external source.		
<ul style="list-style-type: none">• Data stored in at least one list.		
<ul style="list-style-type: none">• Output needs to be produced. This could be text, audio, or any other form of suitable output.		

Continues next page

Item	Candidate to tick	Assessor to tick
2. [EXCELLENCE GRADE] Use constants, variables, or derived values – rather than literals.		
3. [EXCELLENCE GRADE] Ensure the program uses conditions and control structures effectively to: <ul style="list-style-type: none"> • make the program more efficient • reduce/eliminate errors. Examples of ways to do this include the following. <ul style="list-style-type: none"> • Use conditional statements (if/else) in an efficient way (avoid duplication and nested conditionals where relevant) • Use naturally terminating loops (where possible). • Include input validity checks. • Ensure the program correctly handles expected, boundary, and invalid values. 		
4. Document the program with comments.		
5. [MERIT GRADE] Document the program with succinct (short) variable names that clearly describe the data held by the variable.		
6. [MERIT GRADE] Document the program with comments that clearly explain the purpose of code sections.		
7. Test and debug the program to ensure it works on expected cases.		
8. [MERIT GRADE] Test and debug the program to ensure it works on: <ul style="list-style-type: none"> <input type="checkbox"/> expected cases <input type="checkbox"/> boundary cases. 		
9. [EXCELLENCE GRADE] Test and debug the program to ensure it works on: <ul style="list-style-type: none"> <input type="checkbox"/> expected cases <input type="checkbox"/> boundary cases <input type="checkbox"/> invalid cases. 		

Continues next page

Evidence to be submitted

You need to submit the following to your assessor.

- The completed program – in electronic format.
- Records of how you tested and debugged the program. This should be in the form of one of the following.
 - A test plan: This should include:
 - test cases
 - expected results
 - actual results
 - improvements made based on testing (ie how you addressed any errors, etc).

You could add screenshots to your testing plan (eg showing the code you have tested and changes you made because of testing).

- A video walk-through in which you describe:
 - the test cases
 - what happened when you performed each test case
 - how you addressed any errors.

Make sure the following are clearly indicated on all evidence you submit:

- Your name
- Your NSN number
- The Achievement Standard number (ie 92004v2).

..... **END OF ASSESSMENT**

ASSESSMENT SCHEDULE

AS92004v2 Level 1 Credits 5

Create a computer program

IMPORTANT: Assessors need to refer to the **Assessment Sample Answers** document and **Efiles** for exemplars that provide an indication of the type and level of response expected for each grade.

Evidence/Judgements for Achievement	Evidence/Judgements for Achievements with Merit	Evidence/Judgements for Achievement with Excellence
<p>The candidate created a computer program. This included doing the following.</p> <p>Using a suitable programming language to construct a program that performs a specific task [Item 1]</p> <p>Example:</p> <p>The candidate has used a suitable programming language to develop a program that performs the task specified in the brief.</p> <p>The program:</p> <ul style="list-style-type: none"> <input type="checkbox"/> uses variables storing 2 different types of data (string – eg user’s name; numeric – eg to keep track of the user’s score) <input type="checkbox"/> is written in a procedural manner, consisting of sequence, selection, and iteration control structures <input type="checkbox"/> gets input from a user <input type="checkbox"/> uses a collection (list) (eg to store questions/answers) <input type="checkbox"/> produces output (eg text output via GUI). 	<p>The candidate created a well-structured computer program. This included doing the following.</p> <p>As for Achievement</p>	<p>The candidate created a flexible and robust computer program. This included doing the following.</p> <p>As for Achievement, PLUS:</p> <p>Using constants, variables, or derived values in place of literals to make the program flexible [Item 2]</p> <p>Example:</p> <p>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 block of code so that they can be easily found and changed as needed.</p>
<p>Yes <input type="checkbox"/> No <input type="checkbox"/></p>		<p>Yes <input type="checkbox"/> No <input type="checkbox"/></p>

Continues next page

Evidence/Judgements for Achievement	Evidence/Judgements for Achievements with Merit	Evidence/Judgements for Achievement with Excellence
		<p>Using conditions and control structures effectively [Item 3]</p> <p>Example (to make the program efficient and reduce/eliminate errors):</p> <ul style="list-style-type: none"> • The program is concise and easily readable, without redundant code. • The program has a flexible structure using data stored in lists. Actions, built-in functions, conditions, and control structures are used effectively, with redundant code or unnecessary repetition avoided. • The candidate has used parallel arrays (a list with the quiz questions, and a list with the quiz answers). This makes checking for the correct answer more efficient. • The candidate used a naturally terminating <code>for</code> loop to iterate over the list rather than duplicating code for each question. • The program includes input validity checks to ensure the program correctly handles expected, boundary, and invalid values. This included presence checks (to check the user has input data), and range checks (to check the input falls within a specified/expected range). <div> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>

Continues next page

Evidence/Judgements for Achievement	Evidence/Judgements for Achievement with Merit	Evidence/Judgements for Achievement with Excellence
<p>Documenting the program with comments [Item 4]</p> <p>Example:</p> <p>The candidate has included some comments. However, these are not always meaningful and descriptive. Some comments are added to code that is self-explanatory. For example:</p> <pre>q_score = 0 #q_score set to zero</pre> <div data-bbox="147 627 808 699"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>	<p>As for Achievement, PLUS:</p> <p>Using succinct and descriptive variable names [Item 5]</p> <p>Example:</p> <p>The candidate has used concise and meaningful variable names that clearly describe the data held. For example:</p> <p>player_name and player_score (for the player's name and score respectively).</p> <div data-bbox="831 627 1536 699"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div> <p>Documenting the program with comments that clarify the purpose of code sections [Item 6]</p> <p>Example:</p> <p>The comments in the program are clear and meaningful. These comments help to clarify the purpose of code sections (rather than every individual line of code). For example:</p> <pre># The code block below runs if the # player gets the answer correct on # the first try. if question_try == 1: player_score += 1</pre> <div data-bbox="831 1259 1536 1331"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>	<p>As for Achievement with Merit</p>

Continues next page

Evidence/Judgements for Achievement	Evidence/Judgements for Achievement with Merit	Evidence/Judgements for Achievement with Excellence
<p>Testing and debugging the program to ensure it works on expected cases [Item 7]</p> <p>Example:</p> <p>The candidate has provided evidence that they have tested and debugged the program for expected cases.</p> <p>The program is broadly functional but may have minor errors (eg asking for and using the player's name may not work correctly).</p> <div data-bbox="147 711 761 785"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>	<p>As for Achievement, PLUS:</p> <p>Testing and debugging the program to ensure it works on expected and <u>boundary</u> cases. [Item 8]</p> <p>Example:</p> <p>The candidate has provided evidence that they have tested and debugged the program for:</p> <p><input type="checkbox"/> expected cases (see Achieved)</p> <p><input type="checkbox"/> boundary cases (eg testing for condition</p> <pre>if user_age < 14:</pre> <p>using test inputs of 13, 14, and 15 to check program behaviour is as expected.)</p> <div data-bbox="786 801 1476 874"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>	<p>As for Achievement with Merit, PLUS:</p> <p>Testing and debugging the program to ensure it works on expected, boundary, and <u>invalid</u> cases. [Item 9]</p> <p>Example:</p> <p>The candidate has provided evidence that they have tested and debugged the program for:</p> <p><input type="checkbox"/> expected cases (see Achieved)</p> <p><input type="checkbox"/> boundary cases (see Merit)</p> <p><input type="checkbox"/> invalid cases (eg presence check – to check whether user has entered their name; range check – to check if user has entered an integer within a valid age range).</p> <div data-bbox="1503 880 2116 954"> Yes <input type="checkbox"/> No <input type="checkbox"/> </div>

IMPORTANT ASSESSOR INFORMATION

Refer to the unit standard specification at www.nzqa.govt.nz.

- A sample brief for this assessment is provided in the **Assessment Resources** folder. You may need to adjust the brief to ensure it is appropriate for your specific candidates and context. **IMPORTANT:** Any changes you make to the brief **must** still allow candidates to have sufficient opportunity to meet all requirements of the standard, up to Excellence grade.
- Sample answers are supplied separately in the **Assessment Sample Answers** document.
- **This document**, the **assessment sample answers**, and any supporting documents (eg scenario files, resource files, etc) must be supplied for external moderation when required.
- Candidate **privacy** and **confidentiality** must be respected always, including for external moderation.
- **Recording media** (eg video, audio) can be used for verbal responses or as required by the assessment. For more information refer to the NZQA website.