|  |  |
| --- | --- |
| **Alternative Evidence Gathering Template – Internal Assessment** |  |
| These templates must only be used to record student achievement and report results where remote assessment is the only practical option and the collection of direct assessment evidence from students has not been at all possible. ‘Alternative Evidence’ is student evidence for internally assessed standards that has been seen or heard within the teaching and learning programme. These templates do not signal a reduction in what is accepted for each grade, but rather a means of summarising evidence for reporting. These templates must be viewed in conjunction with the standard and assessment advice forwarded to schools to ensure that valid, credible and reliable assessment and learning has occurred before the standard is awarded. While physical evidence of student work does not need to be attached, the assessor decisions made must also be verified internally before reporting results. The template needs to be completed in accordance with the requirements in the Subject Learning Outcomes. |  |
|  |
| Student ID | Student 1 | Subject | Digital Technologies | Level | 1 |
| Notes |  | Standard No. | 92004 | Version | 2 |
| Standard Title | Create a computer program | Credits | 5 |
|  |  |  |
| **Achieved** | **Merit** | **Excellence** |
| Create a computer program. | Create a well-structured computer program. | Create a flexible and robust computer program. |
|  |  |  |
| **Key requirements (list):** | A | M | E | **Describe or attach the evidence considered.**  | **Explain how the judgement was made.** |
| Use a suitable programming language to construct a program that performs a specified task. |[ ]   |  |  |  |
| Test and debug the program to ensure it works on expected cases. |[ ]   |  |  |  |
| Document the program with comments. |[ ]   |  |  |  |
| Use succinct and descriptive variable names. |  |[ ]   |  |  |
| Document the program with comments that clarify the purpose of code sections. |  |[ ]   |  |  |
| Test and debug the program to ensure it works on expected and boundary cases. |  |[ ]   |  |  |
| Use conditions and control structures effectively. |  |  |[ ]   |  |
| Use constants, variables, or derived values in place of literals to make the program flexible. |  |  |[ ]   |  |
| Test and debug the program to ensure it works on expected, boundary, and invalid cases. |  |  |[ ]   |  |
|  |  |  |  |  |  |
| **Sufficiency statement** | **Internal Verification**  |
| Achievement | All of A is required [x]  | Assessor: Date:  |
| Merit | All of A and M is required [x]  | Verifier: Date:  |
| Excellence | All of A, M and E is required [x]  | Verifier’s school:  |
| MARK OVERALL GRADE | N [ ]  | A [ ]  | M [ ]  | E [ ]  | Comments:  |

For the purpose of national external moderation, please follow the external moderation guidelines on the NZQA website.