# Assessment Task Part B

|  |  |
| --- | --- |
| **Assessment Title:** | Programming |
| **Assessment Instructions:** | In this assessment the student must use the design produced in Part A to program the application as per the specified requirements.  This assessment task:   * involves two sub-tasks (Parts **B1 - Write Code** and **B2 - Debugging**). * is a written/practical task to demonstrate the student’s programming skills including debugging and error resolution. * is open book * may be conducted in class, at a workplace or online depending on the student group   Students must:   * write a complete functioning application that meets the Design Requirements provided in the Resource section of this document (Part B1) * provide a list of five different errors encountered while writing code, and an explanation of the tests and debugging procedures that they performed to determine the cause of each error and how it was fixed (Part B2) * complete all tasks to a satisfactory level to receive a satisfactory result |
| **Duration of the Assessment:** | 4 hrs |
| **Required Knowledge** | Introductory level knowledge of programming language, including:   * language data types, operators, expressions and variables * basic language syntax rules * sequence, selection and iteration constructs * industry programming standards and guidelines * commenting techniques * basic data structures * debugging techniques * testing methods * applications |
| **Resources required for this Assessment:** | |
| **Supplied by Institute/workplace** | * computer with programming environment installed (eg [Cloud9](https://aws.amazon.com/cloud9/)) * Git repository (eg [GitHub](https://github.com/)) * internet access * word processor for error resolution logging * Design Requirements located in the Resource section * Coding Standards located in the Resource section |
| **Supplied by student** | For online students:   * computer with programming environment installed (eg [Cloud9](https://aws.amazon.com/cloud9/)) * Git repository (eg [GitHub](https://github.com/)) * internet access * word processor for error resolution logging |

# Task B1: Write Code

This task requires the student to write a computer program that satisfies the requirements specified in the **Design Requirements** document, which was summarised in the checklist created in Task A2.

The program must follow programming standards and guidelines. Students must demonstrate skills in:

* Basic programming language syntax rules and basic data structures
* Using data types, operators, string manipulation, expressions and variables
* Sequencing your program using selection and iteration constructs
* Commenting techniques

**Product:** Written source code located in a Git repository. Student must provide assessor with access to the repository, and upload a zipped archive or the repository to SuniConnect.

# Task B2: Debugging

Debugging the errors that arise in the course of computer application development is a normal part of the programming process. Identifying and resolving these errors is a critical programming skill. This task is completed in conjunction with Task B1. The errors involved will typically be basic syntax errors that crash the program.

For this task, the student must provide a list of **five** **different errors** encountered during Task B1 and an explanation of how they were debugged and resolved. The student must demonstrate the ability to:

* Examine variable values
* Use debugging techniques
* Troubleshoot and solve errors

The student must upload to SuniConnect a word-processor document detailing the debugging errors encountered and how they were resolved (one short paragraph per error).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task B1 Checklist** | | | | | |
| ICTPRG302 Apply introductory programming techniques | | | | | |
| **Student’s Name:** |  | | | **Student ID:** |  |
| **Student Instructions:** | This product checklist is used by the assessor to assess the program code produced in Part B1.  Each item will be recorded by the assessor as **S** if the task/s has been performed to a satisfactory skill level or **NS** if the task/s have NOT been performed satisfactorily.  You must achieve a satisfactory result for the whole of the task. | | | | |
| **Description:** | This checklist contains 20 items that the assessor will be looking for in the student’s program code. | | | | |
| **Items to be observed** | | **1** | **Comments on performance and/or oral question responses (Optional):** | | |
| Specific program requirements | |  |  | | |
| 1: The program takes a CLI argument enabling it to:   * be used for multiple different named backup tasks (eg "job1", "job2", "job3") * run automatically as a scheduled task | | S  NS |  | | |
| 2: Backup tasks are defined in a separate configuration file. | | S  NS |  | | |
| 3: Input errors do not crash program. | | S  NS |  | | |
| 4: Outcome of all backup jobs is logged to file. | | S  NS |  | | |
| 5: An alert is sent when a backup job fails. | | S  NS |  | | |
| 6: Program conforms to coding standards. | | S  NS |  | | |
| 7: Timestamp appended to backed-up files. | | S  NS |  | | |
| 8: Program handles both files and directories. | | S  NS |  | | |
| General code requirements | |  |  | | |
| 9: Apply basic language syntax rules. | | S  NS |  | | |
| 10: Create code using language data types, operators and expressions. | | S  NS |  | | |
| 11: Apply variables and variable scope. | | S  NS |  | | |
| 12: Use program library functions. | | S  NS |  | | |
| 13: Clarify meaning of code using commenting techniques. | | S  NS |  | | |
| 14: Apply language syntax in sequence, selection and iteration constructs. | | S  NS |  | | |
| 15: Create expressions in selection and iteration constructs using logical operators. | | S  NS |  | | |
| 16: Develop algorithms using sequence, selection and iteration constructs. | | S  NS |  | | |
| 17: Create and use data structures. | | S  NS |  | | |
| 18: Code standard sequential access algorithms used in reading and writing text files. | | S  NS |  | | |
| 19: Apply string manipulation. | | S  NS |  | | |
| 20: Examine variable contents and use debugging techniques to detect and correct errors. | | S  NS |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessor Report** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| **Assessment Outcome:** | **SATISFACTORY** | | |
| **NOT SATISFACTORY** | Is resubmission required?Yes  No | |
| **Resubmission:** | Competency development strategies discussed with student? | | |
| Agreed due date for resubmission: / / | | |
| **Assessor Name:** |  | | |
| **Assessor Signature:** | A black background with a black square  Description automatically generated with medium confidence | | **Date:**  / / |