SUNI**TAFE STUDENT**

ASSESSMENT GUIDE



ICTPRG302 Apply introductory programming techniques

Student Name

Place student ID barcode sticker here

Commencement Date



This page has been left intentionally blank

**Intellectual Property Notice**

**© Sunraysia Institute of TAFE [2023]** All rights reserved.

This resource was produced by Sunraysia Institute of TAFE (SuniTAFE) for use by its employees, and as such, they have the authority to print this material. Any further copying or communication of the material in any format or via any means may only be done with the prior documented permission of SuniTAFE. SuniTAFE does not have the authority to give permission for third-party materials that may be included in this resource.

Any enquiries regarding reproduction, use or licensing of this publication should be directed to:

The Copyright Officer

Sunraysia Institute of TAFE

PO Box 1904

MILDURA VIC 3502

Email: [copyright@sunitafe.edu.au](mailto:copyright@sunitafe.edu.au)

General course enquiries should be directed to the SuniTAFE Information Centre (03) 5022 3666.

**Acknowledgements**

SuniTAFE wishes to acknowledge the contribution from the following persons/organisations in the development of this resource:

Mike Hall ICT teacher

David Cleary ICT teacher

Chris Zhong ICT teacher

Education Development Services

SuniTAFE wishes to acknowledge the following additional information sources in the development of this resource:

Commonwealth of Australia. (2013). Home Page, 3.22.1.18. Retrieved October 19, 2022, from training.gov.au: https://training.gov.au

For the full definition of this unit and its assessment requirements, please consult:

https://training.gov.au/Training/Details/ICTPRG302

**Images**

All images individually attributed.

**Suggestions for resource improvements should be sent to edshelp@sunitafe.edu.au**



# Assessment Information

|  |  |
| --- | --- |
| **Conducting the Assessment** | The assessment methods and processes are described in this Student Assessment Guide.  Information about each assessment method is discussed and the assessment agreement is completed.  The assessor is to advise students the time and date of the assessments.  All assessments must be conducted in a safe environment. |
| **Applying Reasonable Adjustments** | Reasonable adjustments to assessment methods and processes may be required to accommodate the student needs and enable them to demonstrate their competencies.  Any adjustments made to the assessment must be documented in the assessment agreement in this Student Assessment Guide. |
| **Feedback to Students** | Feedback will be provided to the student after the completion of assessment marking. |
| **Assessments and Competency** | For each assessment task you complete, the assessor will determine it as either ‘Satisfactory’ or ‘Not Satisfactory’. You will be given the opportunity to resubmit an assessment task if it is determined ‘Not Satisfactory’.  On completion of all assessment tasks, you will be issued a result of either ‘Competent’ or ‘Not Yet Competent’ for the unit. All assessment tasks must be satisfactorily completed for a unit for you to be deemed competent.  A student may appeal a “Not Yet Competent” decision by following the Student Reviews and Appeals Procedure found on the Institute’s website: [www.sunitafe.edu.au/policies-procedures-forms](http://www.sunitafe.edu.au/policies-procedures-forms) |
| **Assessment Submission** | Assessments may be submitted either in hard copy or online through [SuniCONNECT](https://suniconnect.sunitafe.edu.au/login/index.php). Ask the assessor for the preferred method.  If submitting as hard copy, sign and submit the Assessment Task/s Submission Statement with each or all assessment/s.  If submitting online, tick the Submission Statement in the Assessment Task drop box and upload all the relevant documents for assessment. |

**What you need to complete:**

The assessment tasks required for completion of the unit/s are:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Part A** | | | | | | | | | |
| Preparation for programming | | | | | Due Date | | | | |
| **Assessment method/s** | | | | | | | | | |
| A Observation/ oral Questions | B  Project | C  Practical Task | D  Portfolio | E  Role play/Simulation | | F Knowledge based test | H  Written Task | I  Third Party Report | J other (specify)  \_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |  | |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Part B** | | | | | | | | | |
| Programming | | | | | Due Date | | | | |
| **Assessment method/s** | | | | | | | | | |
| A Observation/ oral Questions | B  Project | C  Practical Task | D  Portfolio | E  Role play/Simulation | | F Knowledge based test | H  Written Task | I  Third Party Report | J other (specify)  \_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |  | |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Part C** | | | | | | | | | |
| Testing | | | | | Due Date | | | | |
| **Assessment method/s** | | | | | | | | | |
| A Observation/ oral Questions | B  Project | C  Practical Task | D  Portfolio | E  Role play/Simulation | | F Knowledge based test | H  Written Task | I  Third Party Report | J other (specify)  \_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |  | |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Part D** | | | | | | | | | |
| Presentation | | | | | Due Date | | | | |
| **Assessment method/s** | | | | | | | | | |
| A Observation/ oral Questions | B  Project | C  Practical Task | D  Portfolio | E  Role play/Simulation | | F Knowledge based test | H  Written Task | I  Third Party Report | J other (specify)  \_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |  | |  |  |  |  |

You will be allowed a maximum of 3 attempt/s at each assessment task.

# Assessment Agreement

An assessment agreement is required to ensure that all students are aware of the process and purpose of an assessment and the requirements to achieve competence in this unit/s.

Document any adjustments to assessment discussed with the student.

|  |
| --- |
|  |
|  |
|  |

I declare that I have read and understood the assessment methods and assessment process and have discussed any needs with my assessor.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name: |  | | |
| Student Signature: |  | Agreement Date: | / / |

# Assessment Task Part A

|  |  |
| --- | --- |
| **Assessment Title:** | Preparation for programming |
| **Assessment Instructions:** | In this assessment students are required to plan a simple application using introductory programming techniques. This assessment has 3 parts:  Task A1 – Identify design requirements  Task A2 – Document Program specifications  Task A3 – Document Algorithm  The student must collate a portfolio of evidence and upload to SuniConnect.  This assessment task:   * is open book. * may be conducted in class, at a workplace or online depending on the student group.   Students must:   * contribute to a planning/design discussion. * produce a written program requirements checklist and a design diagram. * complete all tasks to a satisfactory level to receive a satisfactory result. |
| **Duration of the Assessment:** | 2 hrs |
| **Resources required for this Assessment:** | |
| **Required Knowledge** | * programming language features and structures. * how programming language features and structures can be applied to solve practical problems. |
| **Supplied by Institute/workplace** | * computer with required programming language and an IDE installed (eg. [Cloud9](https://aws.amazon.com/cloud9/)) * word processing software * diagram drawing software (eg. [app.diagrams.net](https://app.diagrams.net/)) * internet access * Design Requirements located in the Resource section * Coding Standards located in the Resource section |
| **Supplied by student** | For remote/online students:   * computer * word processing software * internet access |

# Task A1: Discussion

Students will be observed contributing to an instructor-led group reading and discussion of the **Design Requirements** and **Coding Standards** documents provided in the **Resources** section. The purpose of the discussion is to:

* clarify program requirements
* make design suggestions
* clarify coding standards expectations
* identify options for planning, sequencing and prioritising the development process from initial design brief through to presentation of finished product, within the available timeframe

Student contributions could be (but are not limited to):

* an interpretation of a specific requirement
* a suggested programming solution to a specific requirement (eg. use of a specific data type or flow control structure)
* a more global or "architectural" design suggestion as to how the program as a whole could work
* questions to clarify specifications, design or task requirements

The instructor will be looking for evidence that students are able to:

* apply knowledge of programming language features (eg. specific data types and flow control structures) to the development of computer program design solutions.
* plan the sequence and prioritisation of tasks involved

**Observation:** The instructor will observe the student. Details of student participation in the discussion will be recorded by the instructor in the Observation Checklist below.

# Task A2: Project Checklist

Students are to create a written checklist which summarises:

* the planned sequence of steps involved in the development process from initial design brief through to presentation of finished product, within the available timeframe
* the project requirements as specified in the **Design Requirements** and **Coding Standards** documents provided in the **Resources** section. The checklist must contain at least 10 items, which can be presented either in a table or as a series of dot points. Each item must relate to one specific requirement, multiple requirements must not be combined in a single item.

**Product:** The written checklist, which must be uploaded to SuniConnect when complete.

# Task A3: Algorithm Diagram

Students are to create a diagram detailing an algorithm which implements the design requirements of the program. The diagram must illustrate the flow of the program from beginning to end. A suitable online tool for this task is [app.diagrams.net](https://app.diagrams.net/).

An initial draft diagram must be developed prior to commencing Task B, but the draft can be updated during completion of Task B as necessary if the algorithm changes. The final diagram must accurately reflect the content of the program implemented in Task B.

**Product:** A diagram in PNG image file format (almost all diagram tools can export to PNG format), which must be uploaded to SuniConnect when complete.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task A1: Observation Checklist** | | | | | |
| ICTPRG301 Apply introductory programming techniques | | | | | |
| **Student’s Name:** |  | | | **Student ID:** |  |
| **Student Instructions:** | You will be observed by an assessor completing the following task/s. During the task/s you may be asked oral questions by the assessor to confirm your understanding.  Observations 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 records the outcome of the student’s participation in planning discussion using appropriate communication techniques | | | | |
| **Tasks to be observed** | | **1** | **Comments on performance and/or oral question responses (Optional):** | | |
| Listens to questions and answers appropriately using industry terminology. | | S  NS |  | | |
| Asks relevant questions to clarify requirements. | | S  NS |  | | |
| Uses appropriate technical language in discussing requirements. | | S  NS |  | | |
| Articulates complex concepts. | | 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:** |  | | **Date:**  / / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Submission Statement** | | | | | | |
| **Student Name:** | |  | | | **Student Id:** |  |
| **Assessment Task:** | | **Part A** | | | | |
| First submission | | | Subsequent submission | | | |
| **Student Declaration:**  I certify that the attached assessment is my original work. No other person’s work has been used without due acknowledgment in the text of the document.  Except where reference is made in the text, this document contains no material presented elsewhere or extracted in whole or in part from a document presented by me for another qualification at this or another Institution.  I understand the nature of plagiarism to include the reproduction of someone else’s words, ideas or findings and presenting them as my own without proper acknowledgement. Further, I understand that there are many forms of plagiarism which include direct copying or paraphrasing from someone else’s published work (either electronic or hard copy) without acknowledging the source; using facts, information and ideas from a source without acknowledgement; producing assignments (required to be independent) in collaboration with and/or using the work of other people; and assisting another person to commit an act of plagiarism.  I understand that the work submitted may be reproduced and/or communicated by the institution or a third party authorised by the institution for the purpose of detecting plagiarism.  I understand that Sunraysia Institute of TAFE is required to retain evidence of all completed student assessment items for a period of 3 years for auditing purposes, after which time evidence will be securely destroyed. | | | | | | |
| **Student Signature:** |  | | | **Date of Submission: / /** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessor Report** | **Assessor Name:** | | | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
| **Assessment Outcome:** | | **SATISFACTORY** | | |
| **NOT SATISFACTORY** | Is resubmission required?Yes  No | |
| **Resubmission:** | | Competency development strategies discussed with student? | | |
|  | | Agreed due date for resubmission: / / | | |
| **Assessor Signature:** | |  | | **Date:**  / / |

# 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:** |  | | **Date:**  / / |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assessment Task Submission Statement** | | | | | | |
| **Student Name:** | |  | | | **Student Id:** |  |
| **Assessment Task:** | | **Part B** | | | | |
| First submission | | | Subsequent submission | | | |
| **Student Declaration:**  I certify that the attached assessment is my original work. No other person’s work has been used without due acknowledgment in the text of the document.  Except where reference is made in the text, this document contains no material presented elsewhere or extracted in whole or in part from a document presented by me for another qualification at this or another Institution.  I understand the nature of plagiarism to include the reproduction of someone else’s words, ideas or findings and presenting them as my own without proper acknowledgement. Further, I understand that there are many forms of plagiarism which include direct copying or paraphrasing from someone else’s published work (either electronic or hard copy) without acknowledging the source; using facts, information and ideas from a source without acknowledgement; producing assignments (required to be independent) in collaboration with and/or using the work of other people; and assisting another person to commit an act of plagiarism.  I understand that the work submitted may be reproduced and/or communicated by the institution or a third party authorised by the institution for the purpose of detecting plagiarism.  I understand that Sunraysia Institute of TAFE is required to retain evidence of all completed student assessment items for a period of 3 years for auditing purposes, after which time evidence will be securely destroyed. | | | | | | |
| **Student Signature:** |  | | | **Date of Submission: / /** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessor Report** | **Assessor Name:** | | | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
| **Assessment Outcome:** | | **SATISFACTORY** | | |
| **NOT SATISFACTORY** | Is resubmission required?Yes  No | |
| **Resubmission:** | | Competency development strategies discussed with student? | | |
|  | | Agreed due date for resubmission: / / | | |
| **Assessor Signature:** | |  | | **Date:**  / / |

# Assessment Task Part C

|  |  |
| --- | --- |
| **Assessment Title:** | Testing |
| **Assessment Instructions:** | In this assessment students must confirm the application meets initial specifications.  This assessment task:   * involves two sub-tasks (Parts **C1 - Test procedures** and **C2 - Test outcomes**). * is a written and practical task in which the student must demonstrate that the completed program meets design requirements. * is open book * may be conducted in class, at a workplace or online depending on the student group   Students must:   * produce a written testing procedure * use referencing when using information from other sources * complete all tasks to a satisfactory level to receive a satisfactory result |
| **Duration of the Assessment:** | 2 hrs |
| **Required Knowledge** | * introductory level knowledge of programming language * debugging techniques * application testing methods |
| **Resources required for this Assessment:** | |
| **Supplied by Institute/workplace** | For class-based or workplace-based students:   * computer * word processing software * internet access |
| **Supplied by student** | For remote/online students:   * computer * word processing software * internet access |

# Task C1: Test Procedure

This task requires the student to document a test procedure which confirms the correct functioning of the completed computer program. The test procedure must confirm that the program meets the **Design Requirements** specified in the **Resources** section of this document (and also in your summary of those requirements in the Project Brief Checklist in Task A2 above). The purpose of testing in this case is to demonstrate to all stakeholders that the program as a whole meets requirements and is fit for purpose, above and beyond "breaking" errors.

The procedure must consist of:

* a written description of test conditions (eg. the type and structure of test data that must be available for testing).
* a written series of steps to follow to complete the test and confirm the correct functioning of the program.

The test procedure must operate on the model of "If I run the program with these inputs, I expect to see these outputs." It must cover both "success" cases (valid input) and "failure" cases (invalid or incomplete input).

*It may be possible in some circumstances for the student to write and run a formal code test using a tool such as Python's unittest module, but this is not required (as the work involved may be greater and more challenging than that of the original program being tested).*

**Product:** Written test procedure, which must be uploaded to SuniConnect when complete.

# Task C2: Test Outcomes

When the test procedure has been documented, the student must run the test procedure and record:

* the outcome of the test
* any problems identified
* any code updates or changes made as a result of the test

Repeat this cycle of testing and updating until the program satisfies all design requirements.

This process will be observed by the assessor and recorded using the Observation Checklist below.

**Product:** Written log of testing outcomes and any required code changes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task C2: Observation Checklist** | | | | | |
| ICTPRG301 Apply introductory programming techniques | | | | | |
| **Student’s Name:** |  | | | **Student ID:** |  |
| **Student Instructions:** | You will be observed by an assessor completing the following task/s. During the task/s you may be asked oral questions by the assessor to confirm your understanding.  Observations 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 records the outcome of the student’s tests to confirm that the program meets Design Requirements. | | | | |
| **Tasks to be observed** | | **1** | **Comments on performance and/or oral question responses (Optional):** | | |
| 1: Perform test(s) devised in Task C1. | | S  NS |  | | |
| 2: Record the outcome of the test. | | S  NS |  | | |
| 3: Record any problems identified. | | S  NS |  | | |
| 4: Record any code updates or changes made as a result of the test. | | 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:** |  | | **Date:**  / / |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessment Task Submission Statement** | | | | | |
| **Student Name:** |  | | | **Student Id:** |  |
| **Assessment Task:** | **Part C** | | | | |
| First submission | | Subsequent submission | | | |
| **Student Declaration:**  I certify that the attached assessment is my original work. No other person’s work has been used without due acknowledgment in the text of the document.  Except where reference is made in the text, this document contains no material presented elsewhere or extracted in whole or in part from a document presented by me for another qualification at this or another Institution.  I understand the nature of plagiarism to include the reproduction of someone else’s words, ideas or findings and presenting them as my own without proper acknowledgement. Further, I understand that there are many forms of plagiarism which include direct copying or paraphrasing from someone else’s published work (either electronic or hard copy) without acknowledging the source; using facts, information and ideas from a source without acknowledgement; producing assignments (required to be independent) in collaboration with and/or using the work of other people; and assisting another person to commit an act of plagiarism.  I understand that the work submitted may be reproduced and/or communicated by the institution or a third party authorised by the institution for the purpose of detecting plagiarism.  I understand that Sunraysia Institute of TAFE is required to retain evidence of all completed student assessment items for a period of 3 years for auditing purposes, after which time evidence will be securely destroyed. | | | | | |
| **Student Signature:** |  | | **Date of Submission: / /** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessor Report** | **Assessor Name:** | | | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
| **Assessment Outcome:** | | **SATISFACTORY** | | |
| **NOT SATISFACTORY** | Is resubmission required?Yes  No | |
| **Resubmission:** | | Competency development strategies discussed with student? | | |
|  | | Agreed due date for resubmission: / / | | |
| **Assessor Signature:** | |  | | **Date:**  / / |

# Assessment Task Part D

|  |  |
| --- | --- |
| **Assessment Title:** | Presentation |
| **Assessment Instructions:** | This assessment:   * primarily involves the delivery of a verbal presentation to showcase the application developed and the answering of questions. * obtaining feedback and sign-off. * is open book * may be conducted in class, at a workplace   The presentation will be observed by the assessor. The assessor will use an observation checklist to verify your skills. The assessor will provide you with feedback at the conclusion of the observation.  Students must:   * complete all tasks to a satisfactory level to receive a satisfactory result |
| **Duration of the Assessment:** | 2 hrs |
| **Required Knowledge** | * communication skills * stakeholder engagement and sign-off process |
| **Resources required for this Assessment:** | |
| **Supplied by Institute/workplace** | For class-based or workplace-based students:   * computer for demonstrating application functioning * the application developed * word processing software * internet access * presentation venue * stakeholders |
| **Supplied by student** | For online students:   * computer for demonstrating application functioning * word processing software * programming software * internet access * the application developed |

# Task D1: Presentation

When the program is complete and has been tested, students must conduct a formal "stand up" presentation of their application to stakeholders (who could include clients, instructors and/or fellow programmers/students).

The presentation must include the following:

* Verbal summary of design requirements.
* Verbal summary of the technologies/programming language(s) used in the solution, and why they were used.
* Verbal walkthrough of the program/application's use (a demonstration on a computer).
* Evaluation of the program's readiness for use, and suggestions for further improvements or extensions.
* Seeking feedback and obtaining sign off

The student should be prepared to answer questions from the instructor and/or the audience.

The assessor will observe the student and use the Observation Checklist below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task D1: Observation Checklist** | | | | | |
| ICTPRG301 Apply introductory programming techniques | | | | | |
| **Student’s Name:** |  | | | **Student ID:** |  |
| **Student Instructions:** | You will be observed by an assessor completing the following task/s. During the task/s you may be asked oral questions by the assessor to confirm your understanding.  Observations 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 records the outcome of the student’s presentation in which they obtain feedback and sign-off on the completed program. | | | | |
| **Tasks to be observed** | | **1** | **Comments on performance and/or oral question responses (Optional):** | | |
| 1: Demonstrates application referring to specifications and design requirements. | | S  NS |  | | |
| 2: Answers questions using appropriate industry terminology. | | S  NS |  | | |
| 3: Seeks feedback. | | S  NS |  | | |
| 4: Confirms application meets requirements | | S  NS |  | | |
| 5: Asks for final sign off | | 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:** |  | | **Date:**  / / |

# Resources

**Design Requirements**

The client requires a backup program (named **backup.py**) that can potentially be run on any common operating system - Python has been identified as the best programming language candidate to meet this requirement. The type of backup required is a ***daily full backup*** in which all data is backed up each time the program is run, rather than "mirrored" or "synced". In other words, the backup regime will create a series of complete backups each with a unique identifying date, making it possible for the client to retrieve any given document in the form it existed on any given day.

Further specific program requirements include:

* The program must be able to handle both individual files and directories of files.
* The program must be able to be run both manually and as a scheduled task.
* The program must accept a CLI argument enabling it to be used for multiple named backup tasks (eg "job1", "job2", "job3"), eg: python backup.py job1
* Backup job details (source and destination filepaths) that correspond to the named backup tasks must be defined in a separate configuration file (named **backupcfg.py**).
* The program must gracefully handle incorrect or missing input. Such errors must be handled in such a way that they do not cause the program to crash, but instead generate a useful error message indicating what went wrong for logging purposes and for alerting an administrator.
* The program must log the outcome of all jobs to a log file (**backup.log**) whose path is defined in **backupcfg.py**. Each log entry must contain the keyword SUCCESS or FAIL in capital letters to facilitate automated log file monitoring and searching.
* The program must send an alert of some kind (email, SMS) if a backup operation fails for some reason. The message sent must contain the details of the failed operation (job name and any error message) along with date and time.
* The program must comply with the Coding Standards provided above. Note in particular item 6 (internal documentation)
* Source files and directories (the ones being backed up) must be copied to the destination directory (the backup directory) with a timestamp appended to them in the form "YYYYMMDD-HHMMSS". This not only indicates when the backup was made, but enables multiple versions of the same backup job to exist in the same directory without overwriting data or generating errors.

**Coding Standards**

**General**

1: Code implements the requirements specified.

2: Code is technically correct and functions correctly.

3: Code is correctly and tidily laid out in terms of overall structure - eg. header information followed by variable configuration followed by function definitions followed by main script logic. Sections are spaced apart and clearly labelled.

4: Header section contains script title, author, author email and version control information, and copyright and license information if appropriate. This is followed by a brief description of the purpose of the script and how it should be used.

5: Clear and appropriate variable, function and class names are used.

6: Explanatory comments are used throughout as appropriate. This is to help other users (or even the original author!) understand why the code was written the way it was, sometimes months or years later.

7: Consistent indentation is used throughout (one indentation level = 4 spaces).

8: Code implements error handling and fails gracefully as appropriate.

9: Program provides a return value where appropriate.

10: Program has a shebang (eg #!/bin/sh or #!/usr/bin/python3) where appropriate (Unix systems only).

11. Code files are correctly named with appropriate file extensions.

12. Code is managed and version-controlled using Git.

**Python-specific requirements**

1: Class names follow the CapWords convention (aka "Camel Case").

2: Function names should be lowercase, with words separated by underscores as necessary to improve readability.

3: Variable names follow the same convention as function names.

4: Code is documented using the docstring conventions (see below):

* [PEP 8: Style Guide for Python Code](https://www.python.org/dev/peps/pep-0008/)
* [PEP 257: Docstring Conventions](https://www.python.org/dev/peps/pep-0257/)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Summary Report** | | | | | | | | | | | | |
| The Assessment Summary Report is to be completed and signed by the assessor after the completion of all assessments. If competency is not achieved by the student in the first instance, strategies to address competency requirements should be identified and a time for reassessment arranged. | | | | | | | | | | | | |
| **Student Name:** | |  | | | | | | | | | | |
| **Student Id:** | |  | | | | | | | | | | |
| **Unit Code & Title:** | | ICTPRG302 Apply introductory programming techniques (Release 1) | | | | | | | | | | |
| **Course Code & Title:** | | ICT30120 Certificate III in Information technology (Release 3)  ICT40120 Certificate IV in Information technology (Release 4) | | | | | | | | | | |
| **Assessment Tasks:** | | Assessment Task 1 | | | | Satisfactory | | | | Not Satisfactory | | |
| Assessment Task 2 | | | | Satisfactory | | | | Not Satisfactory | | |
| Assessment Task 3 | | | | Satisfactory | | | | Not Satisfactory | | |
| Assessment Task 4 | | | | Satisfactory | | | | Not Satisfactory | | |
| Other if required: | | | | | | | | | | |
| **Assessor Feedback:** | |  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| **Assessment Result for this Unit of Competency:** | | **Student is deemed COMPETENT** | | | | | | | | | | |
| **Student is deemed NOT YET COMPETENT** | | | | | | | | | | |
| **Assessor Name:** | |  | | | | | | | | | | |
| **Assessor Signature:** | |  | | | | | | | **Date:**  / / | | | |
|  | | | | | | | | | | | | |
| **OFFICE USE ONLY** | | | | | | | | | | | | |
| Result entered in SMS: | | |  | | SMS end date reflects assessment date: | | | | | |  | |
| *Date received:* / / | | | | *Date marked:* / / | | | | *Date returned:* / / | | | | |
| ***Student’s post-assessment ACSF summary (Foundation Skills courses only)*** | | | | | | | | | | | | |
| ***Core Skills Level*** | ***Learning*** | | | ***Reading*** | | ***Writing*** | ***Oral Communication*** | | | | | ***Numeracy*** |
|  |  | | |  | |  |  | | | | |  |

CRICOS Provider Code: 01985A RTO Code: 4693

**Call:** 1300 478 648 **Visit:** sunitafe.edu.au

