# 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:** | A black background with a black square  Description automatically generated with medium confidence | | **Date:**  / / |