# Project Assessment: Write and document script

## Criteria

### Unit code, name and release number

ICTPRG405 - Automate processes (1)

### Qualification/Course code, name and release number

ICT40118 Certificate IV in Information Technology

## Student details

### Student number

### Student name

## Assessment Declaration

* This assessment is my original work and no part of it has been copied from any other source except where due acknowledgement is made.
* No part of this assessment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.
* I understand that plagiarism is the presentation of the work, idea or creation of another person as though it is your own. Plagiarism occurs when the origin of the material used is not appropriately cited. No part of this assessment is plagiarised.

### Student signature and Date

Version: 1.0

Date created: 19 October 2018

Date modified: 15 June 2020

For queries, please contact:

Technology and Business Services SkillPoint

Ultimo College

© 2018 TAFE NSW, Sydney  
RTO Provider Number 90003 | CRICOS Provider Code: 00591E

This document can be found in the [Learning Bank](https://share.tafensw.edu.au/share/access/searching.do?doc=%3Cxml%2F%3E&in=P7ac4831b-430a-4b8d-8b56-f7b32ed5b9cf&q=&type=standard&sort=rank&dr=AFTER)

The contents in this document is copyright © TAFE NSW 2018, and should not be reproduced without the permission of the TAFE NSW. Information contained in this document is correct at time of printing: 15 June 2020. For current information please refer to our website or your teacher as appropriate.

## Assessment instructions

Table 1 Assessment instructions

| Assessment details | Instructions |
| --- | --- |
| **Assessment overview** | The objective of this assessment is to assess your knowledge and performance as required to design and document code to automate processes. |
| **Assessment Event number** | 2 of 3 |
| **Instructions for this assessment** | This is a project based assessment and will be assessing you on your knowledge and performance of the unit.  This assessment is in two parts and includes an Assessment Checklist and Assessment Feedback form:   1. Write and test Python script 2. Create user documentation   **Check the Assessment checklist to ensure that you’ve covered all the required tasks.** |
| **Submission instructions** | On completion of this assessment, you are required to upload it to the Learning Management System or hand it to your assessor for marking.   * Complete each of your assessment events and save your files with the event name and your own name – for example: Event1\_john\_smith. * Upload your assessment event files and any other additional attachments to the space provided in the online learning platform.   Note: If your assessment event requires you to use a different file naming convention (i.e. an organisation’s file naming convention), place this file name in the footer of the document then upload the file using the naming convention stated above.  Ensure you have written your name at the bottom of each page of this assessment.  Submit the following documents for each part:   * Part 1: Write and test Python script   + Python script file   + This document * Part 2: Create user documentation   + This document.   It is important that you keep a copy of all electronic and hardcopy assessments submitted to TAFE and complete the assessment declaration when submitting the assessment. |
| **What do I need to do to achieve a satisfactory result?** | To achieve a satisfactory result for this assessment, all questions must be answered correctly and all items in the Assessment Checklist must be marked Satisfactory. |
| **What do I need to provide?** | * A personal computer with internet access * Python 3.x installed * A word processor e.g. Microsoft Word * USB drive or other storage method to save work to, with at least 500KB free space. |
| **What will the assessor provide?** | * Access to the Learning Management System * Access to code debugger, if required ([*Visual Studio code*](https://code.visualstudio.com/)) * Scenario documents as outlined in assessment:   + [*Template Python scripts*](https://share.tafensw.edu.au/share/file/dd843286-ef00-4ec4-9688-7b25800e3fa7/1/ICTPRG405_Python%20Templates.zip) (ICTPRG405\_python templates.zip). |
| **Due date and time allowed** | Indicative time to complete assessment:   * Two hours. |
| **Assessment location** | This assessment may be completed outside of the classroom. |
| **Supervision** | This is an unsupervised, take-home assessment. Your assessor may ask for additional evidence to verify the authenticity of your submission and confirm that the assessment task was completed by you. |
| **Assessment feedback, review or appeals** | Appeals are addressed in accordance with [Every Student’s Guide to Assessment](https://www.tafensw.edu.au/documents/60140/76288/Every+Students+Guide+to+Assessment+in+TAFE+NSW.pdf/cc2b5417-89a6-08f7-9a67-a0c2ff1e26ee). |

## Specific task instructions

This assessment continues on from Assessment 1, using the same scenario.

You can refer to the [*template Python scripts*](https://share.tafensw.edu.au/share/file/dd843286-ef00-4ec4-9688-7b25800e3fa7/1/ICTPRG405_Python%20Templates.zip) (ICTPRG405\_python templates.zip) to help you in creating your script. You can also use a code debugger such as [*Visual Studio code*](https://code.visualstudio.com/)*.*

## Part 1: Write and test Python script

1. Write the script of the solution based on your pseudocode and flowchart from Assessment 1. Your script must:
   * use Python programming language (version 3.x)
   * include useful internal comments, following good commenting practice
     + A prologue with the purpose of the program, a description of what it does, version, date created and author name
     + Variable definitions
     + Explanations of complex code or structures
   * include proper error-handling code for checking:
     + if a folder exists before creating a folder
     + that the text file exists before reading it
   * include appropriate error-handling messages
   * run without errors
   * produce the required output.

**Submit a Python script file (.py).**

1. Ensure that you check your script for syntax and semantic errors. Describe any errors in the script that you corrected – this may include areas that were omitted or coded incorrectly. Include relevant screenshots of your testing.

**Provide your testing documentation in the box below.**

## Part 2: Create user documentation

Prepare user documentation for your script (minimum 35 maximum 75 words). Type or write your instructions into the boxes below.

This must include the following:

1. Read-me instructions, which include:
   * usage statement (authorised use, licence etc.)
   * version and date information
   * installation instructions
   * any special instructions
2. How-to instructions, which include:
   * a description of the program’s purpose and what it does
   * operating instructions
   * required inputs
   * expected outputs
   * relevant screenshots.

## Appendix 1: Assessment Checklist

The following checklist will be used by your assessor to mark your performance against the assessment criteria of your submitted project. Use this checklist to understand what skills and/or knowledge you need to demonstrate in your submission. All the criteria described in the Assessment Checklist must be met. The assessor may ask questions while the submission is taking place or if appropriate directly after the task has been submitted.

Table 2: Assessment Checklist

| TASK/STEP # | Instructions | S | U/S | Assessor Comments |
| --- | --- | --- | --- | --- |
| **Part 1.1** | Script is based on pseudocode and flowchart from Assessment 1 |  |  | *Assessors are to record their comments in sufficient detail to demonstrate their judgement of the student’s performance against the criteria.* |
| **Part 1.1** | Script uses Python version 3.x |  |  |  |
| **Part 1.1** | Internal comments are useful and follow good commenting practice |  |  |  |
| **Part 1.1** | Comments include a prologue, variable definitions and explanations as required |  |  |  |
| **Part 1.1** | Script includes proper error-handling code and messages as required |  |  |  |
| **Part 1.1** | Script produces the required output according to the scenario, without run-time errors |  |  |  |
| **Part 1.2** | Outlines any errors, including relevant screenshots |  |  |  |
| **Part 2.1** | Read-me instructions includes all items as outlined, minimum 35 maximum 75 words |  |  |  |
| **Part 2.2** | How-to instructions includes all items as outlined, minimum 35 maximum 75 words |  |  |  |

## Assessment Feedback

*NOTE: This section* ***must*** *have the assessor signature and student signature to complete the feedback.*

### Assessment outcome

Satisfactory

Unsatisfactory

### Assessor feedback

Has the Assessment Declaration been signed and dated by the student?

Are you assured that the evidence presented for assessment is the student’s own work?

Was the assessment event successfully completed?

If no, was the resubmission/re-assessment successfully completed?

Was reasonable adjustment in place for this assessment event?  
*If yes, ensure it is detailed on the assessment document.*

Comments:

### Assessor name, signature and date:

### Student acknowledgement of assessment outcome

Would you like to make any comments about this assessment?

### Student name, signature and date

***NOTE: Make sure you have written your name at the bottom of each page of your submission before attaching the cover sheet and submitting to your assessor for marking.***