|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student Name** | Ben Rachinger | | **Student Number** | | 472106095 |
| **Unit Code/s & Name/s** | ICTPRG407 Write script for software applications | | | | |
| **Assessment Type** | Portfolio | | | | |
| **Assessment Name** | Programs | **Assessment Task No.** | | | AT1 |
| **Assessment Due Date** |  | **Date submitted** | | | 10/09/2020 |
| **Assessor Name** | Asieh Ryan | | | | |
| **Student Declaration:** I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature. | | | | | |
| **Student Signature** | Ben Rachinger | | | **Date** | 10/09/2020 |
| **PRIVACY DISCLAIMER:** TAFE Queensland is collecting your personal information for assessment purposes. The information will only be accessed by authorised employees of TAFE Queensland. Some of this information may be given to the Australian Skills Quality Authority (ASQA) or its successor and/or TAFE Queensland for audit and/or reporting purposes. Your information will not be given to any other person or agency unless you have given us written permission or we are required by law. | | | | | |

|  |  |
| --- | --- |
| **Instructions to Student** | **General Instructions:**  The purpose of this assessment is for you to demonstrate your skills at developing, testing and debugging software in a scripting language to meet specified requirements. You will also use an integrated development environment (IDE) to do your script development.  Full details of your tasks are given below.  **Information / Materials provided:** You will be provided with access to a suitable IDE for software development.  **Assessment Criteria:**  To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate the following key skills/tasks/knowledge to an acceptable industry standard:   * design and implement three software scripts to meet all given requirements and criteria * write and review pseudocode as a solution for a given software requirement * write and apply tests to verify that the software meets the given requirements * debug one script to identify errors and rectify these errors * provide internal documentation for the written software   **Number of Attempts:**  You will receive up to two (2) attempts at this assessment task. Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2nd attempt. If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task. **For more information, refer to the Student Rules.** |
| **Submission details** | Insert your details on page 1 and sign the Student Declaration. Include this template with your submission.  *Assessment to be submitted via*   * *TAFE Queensland Learning Management System: Connect url:* [*https://connect.tafeqld.edu.au/d2l/login*](https://connect.tafeqld.edu.au/d2l/login) * *Username; 10 digit student number* * *For Password: Reset password go to* [*https://passwordreset.tafeqld.edu.au/default.aspx*](https://passwordreset.tafeqld.edu.au/default.aspx)*>* |
| **Instructions for the Assessor** | Students need to have access to a suitable IDE to work on the assessment tasks. However, they are allowed to work on this assessment off-campus; it is up to the student to organise off-campus access to an IDE. |
| **Note to Student** | An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide. |

My Python programming code for Programs 1, 2 and 3 has been submitted as .py files in addition to this Document. Below are my Pseudo-Code and Testing Table for Program 3.

**Pseudo-Code for Program 3**

Ask the user to Input their name (“User”)

Create a while loop

Say Hello to the “User” and ask them to input what Times Table they want to practice (From the 1 x Table to the 12 x Table).

If the user inputs a number over 12 or 0 or -1 etc, then print “This input is invalid. Please try again. 1-12”.

If the user inputs 1 to 12, store this number and then ask the “User” to input if they’re ready to see the times tables to compare their written times table answers.

If the user inputs Yes that they are ready, print the times table of their chosen number to 12.

If no, ask them again in a loop until they say yes.

If invalid input, then print “This input is invalid. Please try again.”

When the times table is printed, ask the user if their written answers are correct and match the programs answers/output.

If Yes, print “Great Job” and end the program.

If No, print “Better luck next time” and end the program. \

If the input is anything else then print “This input is invalid, Please try again” and loop until the User inputs Yes or No.

**Testing Table for Program 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Input** | **Expected Output** | **Actual Output** | **Pass/Fail?** |
| Whats your name: **Username** | The users name appears in the next input screen from the program i.e. “Hello **Username”** | “Hello **Username**” | **Pass** |
| What times table do you want to practice? **1** | Program moves on to ask the user if they’re ready, shows the number times table they chose. | “"Ok **Username** on a piece of paper, write down the **1** times table from 1 to 12. When you’re ready I’ll show you the answer so you can check your work." | **Pass** |
| What times table do you want to practice? **0** | Program returns an invalid input message. | "This input is invalid, Please try again. (1 - 12)" | **Pass** |
| What times table do you want to practice? **-1** | Program returns an invalid input message. | Program Crashes. | **Fail** |
| Are you ready? **Y** | Prints the times table e.g. 1 x 2, 2 x 2 etc | 1 x 2  2 x 2  3 x 2 | **Pass** |
| Are you ready? **N** | Asks the user “Are you ready?” again. | Program crashes. | **Fail** |
| Did you get it correct? **Y** | “Great job” | “Great job.” | **Pass** |
| Did you get it correct? **875895** | “This input is invalid. Please try again.” | Program crashes. | **Fail.** |