**Purpose**

The Student Assessment Guide provides you with information on how the assessment for this unit will be conducted and the assessment evidence you will need to provide that demonstrates your competency in the unit.

**1. Unit and VET Lecturer Details**

|  |  |
| --- | --- |
| **Unit Code** | ICTPRG301 |
| **Unit Title** | Apply introductory programming techniques |
| **VET Lecturer Name** |  |
| **Location** |  |
| **Phone** |  |
| **Email** |  |
| **Application** | This unit describes the skills and knowledge required to create simple applications or games. It applies to individuals with responsibility for creating applications or games and includes creating code, using programming standards, testing, and debugging. No licensing, legislative or certification requirements apply to this unit at the time of publication. |

**Disclaimer**

Charles Darwin University is collecting information for the purpose of assessing students. Only CDU authorised staff have access to this information. If required for audit purposes, your details may be forwarded to officers from Australian Skills Quality Authority, Australian Government’s national regulator for the vocational education and training sector or other technical experts/advisors. If you are an apprentice/trainee, your personal information, attendance details, progress and results will be disclosed to your employer. If you are under the age of 18 years your personal information, attendance details and results may be disclosed to your parent/guardian. Your personal information will not be disclosed to any other third party without your consent, unless authorised or required by law.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Short Answer Questions Assessment** | | |
| **Assessment Task Number:** 1 Part A | | | |
| **Student Name:** | | | **Student Number:** |
| **VET Lecturer/Assessor Name:** | | | |
| **Unit Code: : ICTPRG301** | | **Unit Title:** Apply introductory programming techniques | |
| **Due Date:** | | **Date Submitted:** | **Number of attempts allowed: 2** |
| **Instructions to Student** | | | |
| To successfully demonstrate competency you must::   * Conduct research and report on the research * The answers to be submitted in a word document and submitted using our generic address: [ICT.submit@cdu.edu.au](mailto:ICT.submit@cdu.edu.au) * Students will have two attempts. Feedback will be provided. Students who do not submit by the due date will be contacted by email and given one week for their second and final attempt. * Assessment 1 Part A can be submitted at any time prior to or by the due date | | | |
| **Reasonable Adjustment** | | | |
| The assessor must record any adjustment made available to the student for this assessment– e.g. written assessment given orally, time extensions, etc. | | | |
| **Student Declaration** | | | |
| I declare that no part of this assessment/assignment has been copied from any other person's work, except where due acknowledgement is made in the text, and no part of this assessment/assignment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.  \*By submitting electronically you agree to the above statement  **Student Signature: Date:** | | | |

| **Questions** | **Satisfactory** | |
| --- | --- | --- |
| **Y** | **N** |
| 1. List two games that use a programming language. (one using python and one using other language) | □ | □ |
| 1. For each game describe the following |  |  |
| * What is the syntax for that language   (the syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language) | □ | □ |
| * What is the command structure of that language   (Structured programming is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures, for and while loops) | □ | □ |

Use the template below to answer these questions. Replace the explanation is red with your answer.

**Question Template**

|  |  |
| --- | --- |
| ICTPRG301: Assessment Task 1 Part A | Student No: |
| Game using Python:  Name of game which uses Python | Other Game language:  Name of game which uses another programming language |
| **Syntax** | |
| Python  Brief general description of the syntax of Python. You might want to mention key words, how blocks are created, what punctuation and braces are used and what they mean etc | Other language  Brief general description of the syntax of Python. You might want to mention key words, how blocks are created, what punctuation and braces are used and what they mean etc |
| **Command Structure** | |
| Python  Brief general description of how the commands of Python fit together. Modern languages are usually either Procedural (working from top to bottom in order) or Object Oriented (everything is defined in a class files with one file having a starting method).  Languages usually are either compiled or interpreted or a combination of both. This refers to the way the instructions are translated to code that can run on a CPU. Describe how Python works | Other language  Brief general description of how the commands of the other language fit together. Modern languages are usually either Procedural (working from top to bottom in order) or Object Oriented (everything is defined in a class files with one file having a starting method).  Languages usually are either compiled or interpreted or a combination of both. This refers to the way the instructions are translated to code that can run on a CPU. Describe how the other language works. |
| How/Where does the program start | |
| Python | Other language |
| Describe how to run the instructions so that they will work | Describe how to run the instructions so that they will work |
| Example if statement | |
| Python  Find an example of how if statements are constructed in Python | Other language  Find an example of how if statements are constructed in the other language |
| Example function definition | |
| Python  Find an example of how functions are constructed in Python | Other language  Find an example of how functions are constructed in the other language |
| Example loop | |
| Python  Find an example of how loops are constructed in Python | Other language  Find an example of how loops are constructed in the other language |

|  |
| --- |
| **Assessor Feedback: Attempt number 1 ☐ 2 ☐**  **Successful ☐ Not Successful ☐** |
| **Assessor Signature: Date:** |
| I have received feedback on my performance:  **Student Signature: Date:** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Project Assessment** | | |
| **Assessment Task Number: 1 Part B** | | | |
| **Student Name:** | | | **Student Number:** |
| **VET Lecturer/Assessor Name:** | | | |
| **Unit Code: ICTPRG301** | | **Unit Title:** Apply introductory programming techniques | |
| **Due Date:** | | **Date Submitted:** | **Number of attempts allowed: 2** |
| **Instructions to Student** | | | |
| To successfully demonstrate competency you must:   * Submit the final working copy and all documentation for the capstone project. * Zumo robots and Raspberry Pi will be available at each practical session. Additional access to this equipment is subject to request. * The answers to be submitted in a word document and submitted using our generic address: [ICT.submit@cdu.edu.au](mailto:ICT.submit@cdu.edu.au) * Assessment 1 Part B to be submitted two week after the final session | | | |
| **Reasonable Adjustment** | | | |
| The assessor must record any adjustment made available to the student for this assessment– e.g. written assessment given orally, time extensions, etc. | | | |
| **Student Declaration** | | | |
| I declare that no part of this assessment/assignment has been copied from any other person's work, except where due acknowledgement is made in the text, and no part of this assessment/assignment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.  \*By submitting electronically you agree to the above statement.  **Student Signature: Date:** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Capstone Project** | | **Requirements** | **S** | **U** |
|  | Create a program that combines physical movement, audio and visual display and integrates at least two modules that includes test code that you have created  The program must interact with the user and the environment.  **Deliverables:**  Project Brief using Sprint document and Python code files.  Python code files also to be submitted in **.py** and **.txt** format | Python File(s)  Sprint Documents | □ | □ |

Sprint Document

|  |  |
| --- | --- |
| Portfolio Activity |  |
| Team Names |  |
| Goal  What exactly are we trying to achieve?  What does success look like?  SMART Goals |  |
| Method  What is to be produced?  How are we going to achieve our goal? |  |
| Metric  What goals have been met?  How do we know they have been met?  What still needs to be done? |  |
| Signed  Date  Student Number |  |
| Signed  Date  Student Number |  |

|  |
| --- |
| **Assessor Feedback: Attempt number 1 ☐ 2 ☐** **Successful ☐ Not Successful ☐** |
| **Assessor Signature: Date:** |
| I have received feedback on my performance:  **Student Signature: Date:** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Checklist Assessment** | | |
| **Assessment Task Number: 1 Part C** | | | |
| **Student Name:** | | | **Student Number:** |
| **VET Lecturer/Assessor Name:** | | | |
| **Unit Code: ICTPRG301** | | **Unit Title:** Apply introductory programming techniques | |
| **Due Date:** | | **Date Submitted:** | **Number of attempts allowed: 2** |
| **Instructions to Student** | | | |
| This assessment will be completed by your lecturer during Practicals. It is a checklist of behaviours which your lecturer will observe during these sessions. | | | |
| **Reasonable Adjustment** | | | |
| The assessor must record any adjustment made available to the student for this assessment– e.g. written assessment given orally, time extensions, etc. | | | |
| **Student Declaration** | | | |
| I declare that no part of this assessment/assignment has been copied from any other person's work, except where due acknowledgement is made in the text, and no part of this assessment/assignment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.  \*By submitting electronically you agree to the above statement.  **Student Signature: Date:** | | | |

|  |
| --- |
| **Assessor Feedback: Attempt number 1 ☐ 2 ☐** **Successful ☐ Not Successful ☐** |
| **Assessor Signature: Date:** |
| I have received feedback on my performance:  **Student Signature: Date:** |

**These activities are not individual assessed but the files will be useful for your assessment project.**

***Challenge Activities are not assessable, however feedback will be given. They are designed for students who are serious about developing a career in programming or related fields.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | **Activity 1 to be done in pairs:** Create a calculator program.  **Deliverables:**   1. Four functions which return correct calculations for Addition, Subtraction, Multiplication and Division 2. Program takes input from the key board and prints the correct answer on the console. 3. Program provides sufficient information to the user to be easy to use.   **Possible structure:**  Create the three initial variables, two integers and one String to make the decision.  Ask the user for the first number (assign to the first integer)  Ask the user for the second number (assign to the second integer)  Ask the user for the type of operation (assign to the String)  Use if statements to work out which command to use  Calculate and display the result  **Focus:**  Completing the task and documenting  **Internet Of Things (IOT) Application:**  Airport/shopping centre car park ticket validation  **Challenge Activity: Extend your calculator so that it can perform other calculations like powers, trigonometry etc. You will need to do some research to find out how Python handles these calculations.** | Python File(s)  Sprint Document |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2. | **Activity 2 to be done in pairs:** Create an LCD library using code from Session 2 for the first function in this activity. Finish the tests for the LCD library from Session Two. Create a function which lights the light and shows a word at the same time. This should be a second function in the LCD library.  Write a small test file that does the following:   * turns the yellow light on and off * writes a word on the LCD when the yellow light turns on * clear the LCD and turn the light off * blink the yellow light, then write another word on the LCD with the yellow light turning on * turn light off and clear the LCD * repeat this pattern   **Deliverables:**   1. LCD library has two functions 2. Function which lights a light and displays a word at the same time 3. Program running on the Raspberry Pi 4. Tests written for all functions in the LCD library   **Possible structure:**  Create two files:   * LCD library * main file to test whether the library functions work   Create LCD function and the LED blink function in the LCD library  Write a main file to import and run both functions from the LCD library  **Focus:**  Completing the task, document testing done  **Internet Of Things (IOT) Application**  Vending machines  **Challenge Activity: Create a function that takes two inputs from the console. The first input is a word that displays on the LCD the second will determine the number of blinks of the LED. You can use the green light as well. Once complete, clear the LCD and repeat.** | Python File(s)  Sprint Document |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3. | **Activity 3 to be done in pairs:** Create a display in the Zumo LCD which puts one star on the first line, two stars on the second line, three on the third up to eight stars. This should be a rolling display so the first screen should be one star on the first line, two stars on the second. The second display two stars on the first line and three on the second line etc up to eight stars and then it stops  This function accepts star as an argument, so the function should produce a similar display for any character.  **Deliverables:**   1. Convert the rolling display the function from Session 3 to work on the LCD screen and place in the LCD library 2. Create the star display and place in a single function in the LCD library with the star as a parameter.   **Possible structure:**  Students to develop their structure in the Sprint documents  **Focus:**  Completing the task, document how goals were met  **Internet Of Things (IOT) Application**  Electronic road work signs  **Challenge Activity: As far as possible centre each of the lines on the LCD for the star display. Get the star display to move up to 8 and back down to one.** | Python File(s)  Sprint Document |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4. | **Activity 4 to be done in pairs:** Move the vehicle in a figure of 8 pattern. Vehicle must return to starting position at the end. Vehicle cannot be more than 10cm from starting location and no more than 30 degrees from starting direction (to allow for vehicle error margin).  Vehicle must halt at the end of the sequence waiting for a button press before moving through the figure of 8 again.  To do this you must create 3 library functions, forward, right and left.  Forward takes an argument and moves the vehicle for a period of time.  Test for 50cm  Right and left turn the vehicle 90o in their respective directions  All function to have documentation using comments.  **Deliverables:**  Students to develop their structure in the Sprint documents  **Possible structure:**  Students to develop in Sprint documents  **Focus:**  Completing the task, document the deliverables and outcomes  **Internet Of Things (IOT) Application**  Park assist in cars  **Challenge Activity: Display the vehicle’s action on the LCD if when the vehicle is moving forward, forward id displayed on the LCD. Use the LEDs as turn indicators.** | Python File(s)  Sprint Document |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5. | **Activity 5 to be done in pairs:** Create a Vehicle Text File Parser and run the vehicle through a series of instruction from a Text File. Vehicle must continually follow the set of instruction with a 5 second delay between the end of the sequence and the commencement of the repeat. Text file to have at least 8 movement instructions. All library functions to have tests.  **Deliverables:**  Students to develop their structure in the Sprint documents  **Possible structure:**  Students to develop in Sprint documents  **Focus:**  Completing the task, document the deliverables and outcomes  **Internet Of Things (IOT) Application**  Automated luggage handling  **Challenge Activity: Develop a Parser which can play a tune developed in a text file. The first part of Baa baa black sheep is given to you to start with.** | Python File(s)  Sprint Document |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6. | **Activity 6 to be done individually:** Create a program that moves the vehicle moving around a room while avoiding obstacles.  Press a button to start and another to stop or pause the program.  Provide user instructions on the LCD  The vehicle should not crash into walls or other solid objects.  The vehicle should be able to exit from a three sided obstacle, similar to a canyon (three walled dead-end).  When testing the IR sensor, write the output to a text file.  All library functions to have tests.  **Deliverables:**  Students to develop their structure in the Sprint documents  **Possible structure:**  Students to develop in Sprint documents  **Focus:**  Completing the task, document how deliverables were developed and achieved, how goals were met, what debugging occurred during the process  **Internet Of Things (IOT) Application**  Home security, robot vacuums  **Challenge Activity: Get the vehicle to accurately follow a line while providing visual and/or audio output** | Python File(s)  Sprint Document |  |  |