**Purpose**

The Unit Outline provides you with information on how the training and assessment for this unit will be conducted.

**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. |
| **Attendance Details** | Attendance is recommended, participation may also include work experience and/or industry participation. |
| **Pre-requisite** | Nil |
| **Co-requisite** | Nil |
| **Work Health and Safety Instructions** | It is a requirement when on campus that you follow the WH&S guidelines of the University found in the [VET Student Guide.](http://www.cdu.edu.au/sites/default/files/mace/docs/VET-student-guide.pdf)  (http://www.cdu.edu.au/sites/default/files/mace/docs/VET-student-guide.pdf)  It is expected that you will adhere to the Workplace Work Health and Safety policies and when working in the office environment ergonomic principles must be observed. |

**2. Student Information**

|  |  |
| --- | --- |
| **Student Support** | Student Administration and Equity Services provide general counselling; complaint resolution; equity information, assistance and support; disability support; indigenous academic support; international student support; language, literacy and numeracy support; student accommodation.  More information is available at:  [Student Services](http://www.cdu.edu.au/studentservices/) (http://www.cdu.edu.au/studentservices/) or  [VET Student Guide.](http://www.cdu.edu.au/sites/default/files/mace/docs/VET-student-guide.pdf)  (http://www.cdu.edu.au/sites/default/files/mace/docs/VET-student-guide.pdf) |
| **Recognition of Prior Learning (RPL)** | If you believe you already have the knowledge and skills to be able to demonstrate competence in this unit speak with your VET Lecturer as you may be able to apply for [Recognition of Prior Learning](http://www.cdu.edu.au/prospectivestudents/studyingatcdu/pathwaystostudy-rpl) (RPL).  (http://www.cdu.edu.au/prospectivestudents/studyingatcdu/pathwaystostudy-rpl) |
| **Reasonable Adjustments** | In the event that you have difficulty understanding or completing the training or assessment due to a disability, language barrier or other difficulties, notify your lecturer as soon as possible. You will be able to discuss with your VET lecturer ways to make reasonable adjustments to the training and assessment process. For example, it may be possible to complete a written assessment verbally, use assistive technologies or have the environment and resources adapted. |
| **Academic Appeals and Complaints Resolution** | If you require an extension of time, special consideration, or appeal against a final result in a unit, you should speak directly to your VET Lecturer. If you are unable to satisfactorily resolve your concern you should refer to the CDU Student Handbook for the process and/or contact:  [Student Administration and Equity Services](http://www.cdu.edu.au/saes) (http://www.cdu.edu.au/saes) or  [Complaints Management Unit.](http://www.cdu.edu.au/strategicservices-governance/complaints) (http://www.cdu.edu.au/strategicservices-governance/complaints). |

**3. Unit Outcomes**

On completion of this unit you will be able to:

| **ELEMENT** | **PERFORMANCE CRITERIA** |
| --- | --- |
| *Elements describe the essential outcomes.* | *Performance criteria describe the performance needed to demonstrate achievement of the element.* |
| 1. Apply language syntax and layout | 1.1 Apply basic language syntax rules  1.2 Use language data types, operators, and expressions to create a clear and concise code  1.3 Apply the variables and variable scope  1.4 Use the library functions in a program  1.5 Use commenting to create a clear meaning to the code |
| 2. Apply control structures | 2.1 Apply the language syntax for sequence, selection and iteration constructs  2.2 Use logical operators to create expressions for use in selection and iteration constructs |
| 3. Code using standard algorithms | 3.1 Develop algorithms that use the sequence, selection and iteration constructs  3.2 Create and use arrays  3.3 Code the standard sequential access algorithms, for reading and writing text files, including end-of-file detection loops  3.4 Apply string manipulation |
| 4. Test the code | 4.1 Use debugging techniques to trace code execution and examine the variable contents to detect, and correct, errors  4.2 Create and conduct simple tests, to confirm that the code meets the design specification  4.3 Document the tests performed and results achieved |
| 5. Create an application or game | 5.1 Design an algorithm in response to basic program specifications  5.2 Develop the application or game to meet the program specification  5.3 Test and confirm that the application, or game, meets the initial specifications |

You will demonstrate this by showing that you can:

**Performance Evidence**

Evidence of the ability to:

* apply programming language syntax, sequence, selection and iteration control structures to the development of an application, or game
* produce an application, or game, that is designed and built from the program specifications
* confirm that the created application, or game, meets the original program specifications, and obtain user sign-off for the completed program.

Note: If a specific volume or frequency is not stated, then evidence must be provided at least once.

**Knowledge Evidence**

To complete the unit requirements safely and effectively, the individual must:

* identify and describe common games programming languages, their syntax, and command structure
* describe the development of small-sized applications or games

**4. Unit Delivery Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Session name, date and time or duration** | **Learning topic/activity** | **Resources required by students** | **Assessment task** |
| 2 Hour Theory | Introduce the following concepts with examples  Print to console statements 1.1  Basic Syntax, functions, error messages 1.1  Input from keyboard to console 1.1  Naming conventions, variable types int, float, String 1.1, 1.2, 1.3  Logic statements if 2.1, 2.2 | Raspberry Pi with IDE installed  All training materials and support materials can be found at <https://roboticpython.brambling.cdu.edu.au/> |  |
| 2 Hour Practical | Introduce the following concepts with examples  Agile  Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed | Sprint Documents  Pair |
| 2 Hour Theory | Introduce the following concepts with examples  Library import 1.4  Function parameter  arguments, general functions 1.1, 1.2, 3.1  Loops while, for 2.1, 2.2, 3.1  Creating own libraries 1.4  Refactoring 5.1  Comments 1.5  Tests 4.2, 4.3 | Raspberry Pi with IDE installed  Robotic controller with an LED light attached |  |
| 2 Hour Practical | Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with an LED light attached | Sprint Documents |
| 2 Hour Theory | Introduce the following concepts with examples  List, Modulo 2,1, 2,2, 3.1, 3.2  DRY principle 1.2  Light sequence, class activity1, 2, 3, 4, 5  How to write a program 1, 2, 3, 4, 5 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached |  |
| 2 Hour Practical | Scum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached | Sprint Documents |
| 2 Hour Theory | Introduce the following concepts with examples  Vehicle movement 1, 2, 3, 4, 5  Refactoring 5.1  Documentation 1,5, 4.3, 5.1, 5.3  *Figure of 8 car movement* | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement |  |
| 2 Hour Practical | Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement | Sprint Documents |
| 2 Hour Theory | Introduce the following concepts with examples  Reading and writing to text files3.1, 3.2, 3.3, 3.4  Built in functions on data types 1.1, 1.2, 1.3, 1.4, 2.2  Designing algorithms 1,5, 3.1, 4.2  Testing 4.3, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement  Vehicle controller has motion sensor attached |  |
| 2 Hour Practical | Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement | Sprint Documents |
| 2 Hour Theory | Introduce the following concepts with examples  Reading a sensor 1, 2, 3, 4, 5  Debugging and testing 4.1, 4.3, 5.3  Designing algorithms 1.5, 3.1, 4.2, 5.2 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement | Part 1 A  Questions |
| 2 Hour Practical | Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement  Vehicle controller has motion sensor attached | Sprint Documents |
| 4X2 Hour Practical | Scrum  Sprint 5.1, 5.2, 5.3 | Raspberry Pi with IDE installed  Robotic controller with multiple LED lights attached  Controller mounted on a vehicle capable of controlled movement  Vehicle controller has motion sensor attached | Sprint Documents  Part 1B Capstone Project |

You will be provided on enrolment with a timetable/ schedule or training plan which contains specific dates, times and locations of the delivery for this unit as well as information about how changes to the timetable will be communicated.

**5. Assessment Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment task number** | **Assessment task name** | **Resources required by students** | **Due date** | **Number of assessment attempts allowed** |
| 1 Part A | Short Answer Questions | USB Computer MS Office Internet Access Printer | Two weeks after final session | 2 |
| 1 Part B | Project | Zumo robot Raspberry Pi USB Computer MS Office Internet Access Printer | Two weeks after final session | 2 |
| 1 Part C | Assessors Checklist  Completed during practicals |  |  |  |

Additional information about the assessment tasks will be provided to you by you lecturer in a Student Assessment Guide for this unit. The assessment tasks have been mapped to the Training Package units of competency and meet all the elements, performance and knowledge evidence and assessment conditions. More information on this unit can be found at [Training.gov.au.](http://training.gov.au/Home/Tga) (http://training.gov.au/Home/Tga).

If you cannot complete an assessment task by the due date you must make alternative arrangements with your VET Lecturer before the due date.

Feedback will be provided by your VET lecturer on each assessment task. The final result for this unit will be recorded as Competency Achieved (**CA**), Not Yet Competent (**NYC**) or Insufficient Participation (**IP**). The results for individual assessment tasks will be recorded as Successful (**S)** and Unsuccessful (**U**). If you are deemed Unsuccessful for a task you will be advised by your VET lecturer and given the opportunity to resubmit.

Remember that your VET lecturer is your most important contact for information about assessment. Contact details are listed on the first page.