# Assignment brief and mapping

**Submission Dates: 15 - 02 -2019**

|  |  |
| --- | --- |
| All Tasks |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** |  | **Evidence** |  | **Unit coverage**  **(LO & AC**  **references)** |  | **Grading ref** |
| 1. Implement a software design using procedural programming |  | Written identification and code sample as per candidate brief |  | 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 17 |  | 0/1 |
| 2. Refine a procedural program to improve quality |  | Code developed as per student brief |  | 2.1, 2.2, 2.3, 2.4 |  | 0/1 |
| 3. Test the operation of a procedural program |  | Code developed as per student brief |  | 3.1, 3.2, 3.3, 3.4m 3.5, 3.6 |  | 0/1 |
| 4. Document a computer program |  | Written identification and code developed as per student brief |  |  |  | 0/1 |

**Assignment mark sheet**

### Candidate name Candidate number

Centre name: Donegal ETB Centre number 079 065

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Evidence** | | | **Achieved** |
| 1 | Written identification and code sample as per candidate brief | | |  |
| 2 | Code developed as per student brief | | |  |
| 3 | Code developed as per student brief | | |  |
| 4 | Written identification and code developed as per student brief | | |  |
|  | |  | Overall grade  (Pass/Fail) |  |

All parts of the tasks must be passed to allow the unit to be claimed.

|  |  |  |
| --- | --- | --- |
| **I can confirm that the evidence listed for this unit is my own work and was carried out**  **under the conditions and context specified in the assessment specification.** | | |
| Candidate signature |  | Date |
| I confirm that the candidate has achieved all the requirements of this unit with the evidence listed and the assessment was conducted under the specified conditions and context, and is valid, authentic, reliable, current and sufficient. | | |
| Tutor/assessor signature |  | Date |
| Quality assurance co-ordinator’s signature (where applicable) |  | Date |
| Qualifications consultant signature (where applicable) |  | Date |

### Where tasks involve inclusion of evidence, for example written descriptions or screenshots of code, these should be referred to and explained as appropriate in assessment reports.

**Evidence to be handed in:**

* Written reports for all tasks
* Reports should include screenshot evidence throughout

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Objective 1 : Implement a software design using procedural programming**  Assessment criteria | | | |
| The learner can: | Achieved 1 Yes  O No | Evidence Ref  PO Portfilio  PL Program Listing  A Appendix | Comment |
| 1.1 Identify the program modules and data and file structures required to implement a given design |  |  |  |
| 1.2 Select, declare and initialize variable and data structure types and sizes to implement design requirements |  |  |  |
| 1.3 Select and implement control structures to meet the design algorithms |  |  |  |
| 1.4 Select and declare file structures to meet design file storage requirements |  |  |  |
| 1.5 Select and use standard input/output commands to implement design requirements |  |  |  |
| 1.6 Make effective use of operators and predefined functions |  |  |  |
| 1.7 Correctly use parameter passing **mechanisms**. |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Learning Objective 2: Refine a procedural program to improve quality** | | | | |
|  |  |  | |  |
| Assessment criteria |  |  |  | |
| The learner can: | Achieved 1 Yes  O No | Evidence Ref  PO Portfilio  PL Program Listing  A Appendix | Comment | |
| 2.1 Use an agreed standard for naming, comments and code layout |  |  |  | |
| 2.2 Define user functions to replace repeating code sequences |  |  |  | |
| 2.3 Implement data validation for inputs |  |  |  | |
| 2.4 Identify and implement opportunities for error handling and reporting. |  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Objective 3: Test the operation of a procedural program** | | | |
| Assessment criteria:  The learner can: | Achieved 1 Yes  O No | Evidence Ref  PO Portfilio  PL Program Listing  A Appendix | Comment |
| 3.1 Make effective use of available debugging tools |  |  |  |
| 3.2 Prepare a test strategy |  |  |  |
| 3.3 Select suitable test data and determine expected test results |  |  |  |
| 3.4 Record actual test results to enable comparison with expected results |  |  |  |
| 3.5 Analyze actual test results against expected results to identify discrepancies |  |  |  |
| 3.6 Investigate test discrepancies to identify and rectify their causes. |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learing Outcome 4: Document a Computer Program**  Assessment criteria | | | |
| The learner can | Achieved 1 Yes  O No | Evidence Ref  PO Portfilio  PL Program Listing  A Appendix | Comment |
| 4.1 Create documentation to assist the users of a computer program |  |  |  |
| 4.2 Create documentation for the support and maintenance of a computer program. |  |  |  |

**Grade Summary**

|  |  |  |
| --- | --- | --- |
|  | Target Score for pass | Actual Score |
| Learning Objective 1: Implement a software design using procedural programming | 7/7 |  |
| Learning Objective 2: Refine a procedural program to improve quality | 4/4 |  |
| Learning Objective 3: Test the operation of a procedural program | 6/6 |  |
| Learning Outcome 4: Document a Computer Program | 2/2 |  |
| Total | Out of 19 |  |
| Grade (Pass/Fail) |  |  |

## Candidate feedback sheet

Candidate name

|  |  |
| --- | --- |
| **Task & date** | **Assessor comments/action plan**  (The assessor should sign after each feedback session) |
|  |  |

Candidate signature Date

Tutor/assessor signature Date

**1 Implement a Software Design Using Procedural Programming**

* 1. Identify the program modules, data and file structures required to implement a given design

1.2 Select, declare and initialize variable and data structure types and sizes to implement design requirements

1.3 Select and implement control structures to meet the design algorithms

1.4 Select and declare file structures to meet design file storage requirements

1.5 Select and use standard input/output commands to implement design requirements

1.6 Make effective use of operators and predefined functions

1.7 Correctly use parameter passing mechanisms.

1. **Refine a procedural program to improve quality**

2.1 Use an agreed standard for naming, comments and code layout

2.2 Define user functions to replace repeating code sequences

2.3 Implement data validation for inputs

2.4 Identify and implement opportunities for error handling and reporting.

**3. Test the operation of a procedural program**

3.1 Make effective use of available debugging tools

3.2 Prepare a test strategy

3.3 Select suitable test data and determine expected test results

3.4 Record actual test results to enable comparison with expected results

* 1. Analyze actual test results against expected results to identify discrepancies
  2. Investigate test discrepancies to identify and rectify their causes.

**4. Document a computer program**

4.1 Create documentation to assist the users of a computer program – user manual

4.2 Create documentation for the support and maintenance of a computer program – maintenance /technical document

**5. Appendix**

5.1 Program Listing

5.2 Program Flow – output

## 