|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student Name |  | | | | Student Number | |  | |
| Unit Code/s & Name/s | ICTPRG523 Apply advanced programming skills in another language | | | | | | | |
| Assessment Type | Written | | | | | | | |
| Assessment Name | Programming Assignment | | | | Assessment Task No. | | AT1 | |
| Assessment Due Date |  | | | | Date submitted | |  | |
| **Assessor Feedback:**  **Student provided with feedback** *(check box when completed)* | | | | | | | | |
| **Attempt 1** | | Satisfactory | | Unsatisfactory | | Date | | / / |
| **Attempt 2** | | Satisfactory | | Unsatisfactory | | Date | | / / |
| Assessor Name |  | | Assessor Signature | | |  | | |
| Note to assessor: Please record below any reasonable adjustment that has occurred during this assessment e.g. written assessment given orally. | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessment criteria / benchmarks  **The evidence submitted demonstrates that the student satisfactorily:** | Attempt 1 | | Attempt 2 | |
| S | U | S | U |
| 1.1 - Design dynamic data structures |  |  |  |  |
| 1.2 - Implement and use dynamic data structures, including double-linked lists and binary trees for coding |  |  |  |  |
| 2.1 - Code using hashing techniques |  |  |  |  |
| 2.2 - Consider and record the advantages and disadvantages of at least three sorting algorithms |  |  |  |  |
| 2.3 - Code at least one sorting algorithm |  |  |  |  |
| 2.4 - Code advanced searching techniques for use with complex data structures |  |  |  |  |
| 3.1 - Demonstrate use of the features of the language that enable inter-process communication through at least one mechanism |  |  |  |  |
| 3.2 - Demonstrate use of features of the language that allow for OS ‘signals’ to be captured and responded to |  |  |  |  |
| 4.1 - Use a third-party library in the construction of an application |  |  |  |  |
| 4.2 - Reference third-party documentation |  |  |  |  |
| 4.3 - Use procedural techniques to write an application to work within a graphical user interface (GUI) environment |  |  |  |  |
| 5.1 - Use integrated development environment (IDE) debugging facilities or a stand-alone debugger |  |  |  |  |
| 5.2 - Use program debugging techniques to detect and resolve errors of syntactical, logical and design origin. |  |  |  |  |
| 6.1 - Demonstrate use of source code version control |  |  |  |  |
| 6.2 - Demonstrate adherence to guidelines for developing maintainable code and to a set of provided coding standards |  |  |  |  |
| 6.3 - Apply suitable internal documentation to all code created and using documentation tools available in the target language |  |  |  |  |
| 7.1 - Design and document tests to be undertaken |  |  |  |  |
| 7.2 - Undertake limited testing of produced code to ensure it complies with program specification |  |  |  |  |
| 7.3 - Capture test results |  |  |  |  |
| 8.1 - Develop a solution from a program specification design document |  |  |  |  |
| 8.2 - Design the algorithm, and construct and test applications in response to a problem description and language |  |  |  |  |
| 8.3 – Write e-mail using correct grammar and formatting, and a range of vocabulary, appropriate to the manager. |  |  |  |  |
| **Evidence of the student having demonstrating consistent performance (under the following assessment conditions):** |  |  |  |  |
| Using an appropriate software development environment |  |  |  |  |
| With supplied technical documentation |  |  |  |  |
| **The student has demonstrated competency in:** |  |  |  |  |
| Task skills - performing every task in the assessment at an appropriate skill level |  |  |  |  |
| Task management skills - managing the various tasks in this assessment at an appropriate skill level, and within a working environment that replicates a workplace appropriate for software development. |  |  |  |  |
| Contingency management skills - managing issues that arise at an appropriate skill level, and within a working environment that replicates a workplace appropriate for software development. These might include, but not limited to: staged development with progressive unit and integrated testing, referencing and analysis of industry-standard algorithms for standard sorts and data structure management, and progressive validation of completed implementation against project specifications. |  |  |  |  |
| Job role environment skills – operating and interacting appropriately and effectively within a working environment that replicates a workplace appropriate for software development. |  |  |  |  |
| Note: These checklist items are based on additional Critical Evidence, Required Skill and Required Knowledge not explicitly stated elsewhere in the assignment requirements checklists. |  |  |  |  |