Assessment Submission Coversheet:  
Introduction to C#

|  |  |
| --- | --- |
| **Student Name:** | Justin Katic |
| **Student Number:** | AOI00008ZB |
| **Student Email** | S191284@students.aie.edu.au |
| **Course Stream:** | ICT50215 - Diploma of Digital and Interactive Games |
| **Assessment Name:** | Introduction to C# |
| **Units Covered:** | ICTPRG430 – Apply introductory object-oriented language skills |
| **Teacher/s:** | Jay Yabsley |
| **Due Date:** | 13/03/2020 |
| **Date of Submission:** | *Will be automatically recorded on Canvas* |
| **Assessment Work Location** | Canvas |

**Declaration**

By submitting this work under my name, I declare that my submission is my own work with respect to plagiarism and does not violate any copyright laws. I have retained a copy of this assessment material that I can produce if requested.

Tick to acknowledge you have read and agree with this declaration.

Name: Justin Katic Date: 13/03/2020

Assessment Submission Coversheet:  
Introduction to C#

**Work Submitted:***Tick to acknowledge you have submitted each required piece of assessment work.*

1. **Application Design**: You are to complete your application based on the brief provided in *Appendix 1* of the *Subject and Assessment Guide*.  
   - A program that stores an array of items  
   - A program that adds text commands that allows for buying, selling from an array  
   -A program that reads in an array of items from a text file  
   -Added classes and sub classes to program  
   -ability to update items in array via text file
2. Implementation: Complete the application development to meet the supplied brief (As explained above). Ensure that it is well written, clearly coded and documented.

Each Function is commented and clearly written and documented

1. **Application Testing**: Submit evidence of the project files for at least two stand-alone subsystems and have working unit test cases for each system with clear documentation.

Submitted 2 sub systems that I had to test throughout my project the testing was mainly done through debugging and trial and error within the exe.

1. **Application Handover**: You must submit the source files and a release build of the RPG Store Simulation, removing all temporary files in the obj and bin folders. All assessment items must be submitted as a single compressed file (.zip or .7z).  
   Submitted the final release of the program in compressed folder to instructure submit section.

*For more information on these parts, please click on the* ***Subject and Assessment Guide*** *link in the course* ***Game Programming Year 1*** *under the subject* ***Introduction to C#*** *on* [*https://aie.instructure.com*](https://aie.instructure.com) *and read the* ***2020 Subject & Assessment Guide – Introduction to C#***

Assessment Submission Coversheet:  
Introduction to C#

**Submission Checklist:**

|  |  |
| --- | --- |
| **General** **Description** | **Y/N** |
| All submitted projects compile without errors. Programs that don’t compile cannot be assessed | Y |
| The program includes a “readme” or document explaining how to compile, execute and operate the program | Y |
| The program performs as described in the general description | Y |
| The program contains no logical errors | Y |
| The code is sufficiently commented and clean | Y |
| An attempt has been made to increase the program’s efficiency | Y |
| Code compiles without no warnings | Y |
| Program executes without crashing | Y |
| Program has no memory leaks, and closes all files after use | Y |
| A release executable has been made and included in the submission | Y |
| Project files and source code are included in the submission | Y |
| All files are packaged in a single compressed archive | Y |

To ensure you have submitted the right components, please fill out these checklists.

|  |  |
| --- | --- |
| Estimate the number of hours taken to complete this assessment | 30 |
| How many times have you submitted this assessment (including this time)? | 1 |

**Required Features**  
**Complete the following table by providing the class name or file name, along with the line number, to show where you have implemented each feature.**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Class/File** | **Line Number** |
| The program implements a base *Item* class, and two or more sub-classes | Y | Items.cs line  9  Potion.cs line  9  Armour.cs line  9  Weapon.cs line 9 |
| The program stores items in a *player* inventory array and a *store* inventory array | Y | Inventory.cs  Line 19  Program.cs line 57  Program.cs line 169 |
| The program loads both inventory arrays from a text file upon launch | Y | Program.cs line 76  Program.cs line 99 |
| The program saves both inventory arrays to a text file before exit | Y | Program.cs line 288  Program.cs line 306 |
| The program contains at least 2 classes containing four or more variables | Y | Potion.cs line  9  Armour.cs line  9  Weapon.cs line 9 |
| The program overrides an object constructor at least once | Y | Potion.cs line  19  Armour.cs line 19  Weapon.cs line 19 |
| The program implements text commands to buy and sell items, and view item descriptions | Y | Program.cs line  152  Program.cs line  185  Program.cs line  237 |
| Unit testing has been conducted on at least two sub-systems. These projects are included in the submission | Y | Program.cs line  306  Program.cs line  244 |
| Code is well commented (i.e., each function and class is commented) | Y | everywhere |

|  |  |
| --- | --- |
| **Feature** | **Y/N** |
| Unit testing has been conducted on at least two sub-systems. These projects are included in the submission | Y |
| Code is well commented (i.e., each function and class is commented) | Y |