

**Technical Specifications**

Introduction to C++

CIP Code 11.0804

Advanced Diploma of Professional Game Development

Game Programming

#### General Description

This subject teaches you to program in C++ and how to apply the C++ language to game development. You will be assessed as competent in the necessary skills if you can successfully complete the assessment criteria described in this section.

The assessment for this subject requires you to develop a class in C++ to represent text strings and then to use that class to develop a simple text-based adventure game. This assessment is designed to test the knowledge and skills that you will be learning throughout the subject, namely:

* Applying basic C++ syntax and layout
* Applying basic object-oriented principles in C++
* Code documentation and testing

The assessment is divided into four parts, which you should complete in order. Each part is dependent on the earlier parts being completed.

#### Evidence Specifications

##### Part 1 – Create a String Class

You are to create a basic string utility class that will make working with character arrays easier to manage. Your String class should contain the functions listed below. You need to include functions that implement the following features in your String class:

* The ability to query the string’s length, returning an integer
* The ability to access a character at a certain index within the string class
* The ability to compare if the string is the same as another string class
* The ability to append one string to another
* The ability to prepend one string to another
* The ability to return the string as a basic constant C-style string (const char\*)
* The ability to convert the string to a duplicate containing all lowercase letters
* The ability to convert the string to a duplicate containing all uppercase letters
* The ability to find a sub-string within the string class
* The ability to find a sub-string within the string class, starting from a certain index within the string
* The ability to replace a sub-string found within the string with a different sub-string
* The ability to set the string to an input C-style string

##### Part 2 –Unit test String Class

As you add functionality to your string utility class you will need to test each function to ensure its correctness. You are required to create simple test functions to validate every function specified in part 1 and include them within an application that you run often to ensure the validity of your String class.

Your unit-test application must also write results to a text-based log file. If a test log file already exists, append the test results to the end of the file. If the file does not exist, one should be created automatically.

After each run of tests, output the name of the function being tested and its pass or fail result.

If each item in the document is separated by a tab character than this will become nicely formatted within MS Excel, Open Office Calc or Google Spreadsheets. The output should look something like this:

*Test 0 Length Successful Test 1 CharAt Successful*

*Test 2 EqualTo Failed*

*Test 3 Append Failed*

*Successful 50.00%*

*Test 0 Length Successful*

*Test 1 CharAt Successful*

*Test 2 EqualTo Successful*

*Test 3 Append Failed*

*Successful 75.00%*

##### Part 3 –Text-Based Adventure Game

You are to create a basic text adventure game within the console. Text adventure games are a form of Interactive Fiction that allows the player to navigate an imaginary world by entering text commands in to the console. For an overview of text adventure games see: <https://en.wikipedia.org/wiki/Interactive_fiction>

The player should be able to navigate between locations by typing commands such as “Move North”, “Move South”, “Move East”, and “Move West”. Your string class should compare entered text against available commands to ensure that the user has entered a valid command.

Each location should have a description that is printed to the console when the location is entered describing the current surroundings.

Different types of locations should exist that allow for different types of commands to be entered. For example, a “Riddle Room” may ask a question of the user and the user must enter the correct answer before they are allowed to move into an adjacent room.

You will use your string class to process all commands within your game. Your code might look something like this:

*MyString command;*

*Std::cin >> command; // waits for input from console.*

*if ( command.ToLower().Equals("move south") == true ) {*

*// TODO: Process Move South Command }*

For this assessment, you need to demonstrate use of inheritance, with at least two levels of inheritance. As an example you could create a ‘Base Location’ class for which all other location types will inherit from. Another suggestion would be to have Monsters within your rooms that inherit from a base monster class, i.e. Monster -> Orc -> Orc Boss

#### Evidence Rubric

This is the specific evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed below. You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your teacher.

* String class
* Unit-test String class
* Text Adventure Game

|  |
| --- |
| Assessment Requirements |
| String class Evidence that includes:   * A .h and .cpp file submitted that implements the defined String class as per the specifications in Part 1 of the Assessment Description above, following industry-standard coding techniques * String class code is properly commented to describe the functionality and use of the class * Submitted code is free from faults and errors |
| Unit-test String class Evidence that includes:   * Executable application and source-code for the application submitted that performs a Unit-test of the String class and outputs results of tests as per the Assessment Description above * Application runs successfully and is free from faults and errors * Tests all pass successfully |
| Text Adventure Game Evidence that includes:   * Executable application and source-code submitted that implements a Text Adventure Game as per the specification in the Assessment Description above * Text Adventure Game utilises the String class created in Part 1 of the Assessment Description * Code uses class inheritance to implement multiple levels of inheritance, being at least 2 levels deep |

This table defines the individual requirements for each part of the assessment criteria. Please refer to provided subject assessment documentation for any additional requirements.