Staffordshire University

Faculty of Computing, Engineering and Sciences

COSE40674
Object Oriented Programming in C++
2019

C++ Programming Assignment

Module Learning Outcomes for this assignment

DEMONSTRATE A CRITICAL UNDERSTANDING OF THE FEATURES OF THE C/C++ PROGRAMMING LANGUAGE WITHIN A WINDOWS ENVIRONMENT.
(Application, Communication, Knowledge & Understanding)

ANALYSE A GIVEN TASK SCENARIO AND APPLY THE FEATURES OF THE C/C++ PROGRAMMING LANGUAGE TO IMPLEMENT A VIABLE SOLUTION.
(Analysis, Application, Learning, Problem Solving)

CRITICALLY SELF REFLECT UPON ALTERNATIVE SOLUTIONS TO GIVEN PROBLEMS.
(Analysis & Communication, Knowledge and Understanding, Reflection)

UTILISE THE C++ PROGRAMMING LANGUAGE TO DEVELOP A LARGE-SCALE OBJECT ORIENTED SOLUTION TO A GIVEN SYSTEM TASK AND CRITICALLY APPRAISE THE SOLUTION.
(Analysis, Application, Knowledge and Understanding, Problem Solving)

Submission Deadline – 23:30 on Wednesday 15th May 2018

(Demonstrations will take place between Thursday 16th May and Friday 24th May : room and times will be advertised in advance)

Assessment Overview

A programming task in 'C++', weighted at 100% for the module

Task Specifications

A Publisher Database in C++: Implementation [20 Marks]

The aim of this assignment is to produce a simulation of a custom publisher database within a Windows™ development environment using object-oriented C++. The final code will be required to run on Visual Studio 2017 on one of the computers in the labs in the Mellor building.

You are required to produce a C++ program which represents a publisher database. The publisher wishes to access information on a variety of publications using this computer program.

You will load the database from a text file, or 'hard code' it into your program. You do **not** need to save any changes back to a text file.

The database will store the following information:

- **Paperback books**

The database is required to store the following information for each book:

a. ID (a number or a string)
b. Title (A string)
c. Number of pages (A number)
d. Book content (A large string)

The book content will be provided and is available as two files in the assessment folder on blackboard - each file will contain several hundred words - you may use one data file per book. When your program is loaded you should load either one of the two datafiles for each book. Obviously, this means some books will have the same content - but this doesn't matter for this assessment.

- **Hardback books**

The same as paperback books, but it will also have a property for a weight for each book. For example, this might be 0.9 kilograms.

- **Magazines**

This is the same as the paperback book, but it also contains an array of advertisement objects. An advertisement object has a title (string) property and an advertiser ID (number or string) property. The advertiser ID should relate to an Advertiser in the publisher database (see next section).

- **Advertisers**

An advertiser is a company or person. This object should contain the name (string) of the person or company and an ID (a string or a number).

Functionality

The program should make use of the following C++ concepts:

- Class - methods and properties
- Abstract base classes
- Inheritance
- Composition
- Operator Overloading
- Threading
- Mutexes
- C++ Strings
- New and Delete

The program will have a simple user interface in the console which will ask the user to enter a number to choose from the following menu:

1. List all magazines
2. List all paperback books
3. List all hardback books
4. List all advertisers
5. List advertisements and advertiser for a magazine
6. Remove a Book / Magazine by ID
7. Add a magazine
8. Search for a word
9. Sum of the weight of all hardback books

List all magazines

Output the ID, title, number of pages of each to the console screen

List all paperback books

Output the ID, title, number of pages of each book to the console screen

List all hardback books

Output the ID, title, number of pages, and shipping weight of each book to the console screen

List advertisements

Prompt the user to input a magazine ID. The program will then list number of advertisements, and

publishers

If there is more than advertisement for a advertiser, list advertiser only once

Remove a Book / Magazine by ID

Prompt the user to input a magazine or book ID. The program should then delete the magazine/book from memory. If you are loading magazines/books from a file at startup - you do NOT need to delete the magazine/book from this, only from memory.

Add a magazine

Ask the user to input a title and a page count. The program will then assign an ID, the content (from one of the two content files), and three advertisement objects which must have the advertiser ID from an existing advertiser.

Search for a word
This functionality must use threading. For each book and magazine, it will create a new thread and search the content for a user input word which may or may not exist. It should then output the word count of that word for each book or magazine. Each individual thread can output the word count. For example, if there 5 books and magazines in total, you will run 5 threads.

Each thread must take at least 5 seconds. You can use std::chrono to ensure this requirement - so after your search is done, you could 'wait' or 'sleep' for enough time to **before** outputting the word count and exiting the thread:

The sum of the weight of all hardback books
Simply total and output the weight of all hardback books.

Loading database at startup:

You can hard-code the database into your program if you wish, but for more points, you could import the database from a simple text file.

Documentation
You are required to include the following documents…

1. A brief user manual that will allow a member of the module teaching team to set up and run your submission without you being in attendance.

2. A brief document (max 1000 word) that provides evidence of correct operation. This could be screenshots or an extract from any log files that your processes generate.

3. A brief document (max 1000 words) explaining the design of your program. This should include which classes you have created, and a brief explanation of their methods and properties. If you have used C++ Inheritance this should be

clear in your class explanations.

Marking Scheme Overview TOTAL MARKS 20

Correct functionality of User Interface:	8 marks
Class Design	2 marks
Implementation of C++ features	8 marks
- Abstract base classes	
- Inheritance	
- Composition	
- Operator Overloading	
- Threading	
- Mutexes	
- C++ Strings	
- New and Delete	
Documentation	2 marks

Note: ***Even if your code does not function correctly you will still be awarded marks for your attempt. Non compiling code will receive zero marks for the code aspect of the assignment.***

Demonstration Requirements

You will be required to demonstrate your submission between Thursday 16th May and Friday 24th May – a booking system will be made available prior to the assessment hand in date.

All demonstrations will take place at a date to be confirmed, failure to demonstrate will result in zero marks being awarded for the whole of this task, regardless of how much has been attempted.

If you are unable to attend a demonstration, the Extenuating Circumstances process should be followed.

Submission Requirements

Submission will be carried out using Blackboard.

You are required to submit electronic versions of the following items, contained within a single .zip file.

1. 'C++' source code of the Task as simple text files ('.cpp' & '.h').

2. Documentation as described above

Please **do not** embed your code within a Word™, or similar, file – keep them as simple text files.

Failure to submit in strict accordance with the University Regulations may result in zero marks being awarded for all aspects of the assignment.