The British College

OBJECT ORIENTED PROGRAMMING IN C++

ASSESSMENT-1[25% of Assignment]

DETAILS OF THE ASSESSMENT

You are required to develop a simple console program, using the C++ programming language, to meet the specified requirements as detailed below. You will be required to demonstrate the work, you've submitted to your tutor in order to receive marks. During this demonstration your tutor will examine your solution and also ask questions about how the work was undertaken.

Assignment Overview

The aim of this assignment is to create a console-based C++ program called "Date.cpp" As per the requirements mentioned.

All students are required to make use of appropriate coding conventions in C++ making use of the things covered and learnt in the classes so far for solving the following problem.

Note – To achieve the marks, data must be entered, read and displayed as per the requirements. The assignment has been divided into three parts, each part divided into a number of requirements as described below.

Please note - Any awarded mark is for the demonstration and explanation of the work, rather than for the submitted work itself.

ASSESSMENT-1(25% of assignment marks):

Due **27th May 2022 23:59.** Demo on week beginning **1st June 2022** during your timetabled lab session.

Requirement 1 (5 marks)

Design a class called Date. The class should store a date in three integers: month, day, and year. There should be member functions to print the date in the following formats: 12/25/2022; December 25, 20122; 25 December 2022.

Input Validation: Do not accept values for the day greater than 31 or less than 1. Do not accept values for the month greater than 12 or less than 1.

Requirement 2 (10 marks)

Modify the Date class.
The new version should have the following overloaded operators:

- ++ Prefix and postfix increment operators. These operators should increment the object’s day member.
- -- Prefix and postfix decrement operators. These operators should decrement the object’s day member.
- - Subtraction operator. If one Date object is subtracted from another, the operator should give the number of days between the two dates. For example, if April 10, 2022 is subtracted from April 18, 2022, the result will be 8.

Requirement 3 (10 marks)

The class should detect the following conditions and handle them accordingly:

- When a date is set to the last day of the month and incremented, it should become the first day of the following month.
- When a date is set to December 31 and incremented, it should become January 1 of the following year.
- When a day is set to the first day of the month and decremented, it should become the last day of the previous month.
- When a date is set to January 1 and decremented, it should become December 31 of the previous year. Demonstrate the class’s capabilities in a program using C++.

ASSIGNMENT MARKING SCHEME / CRITERIA

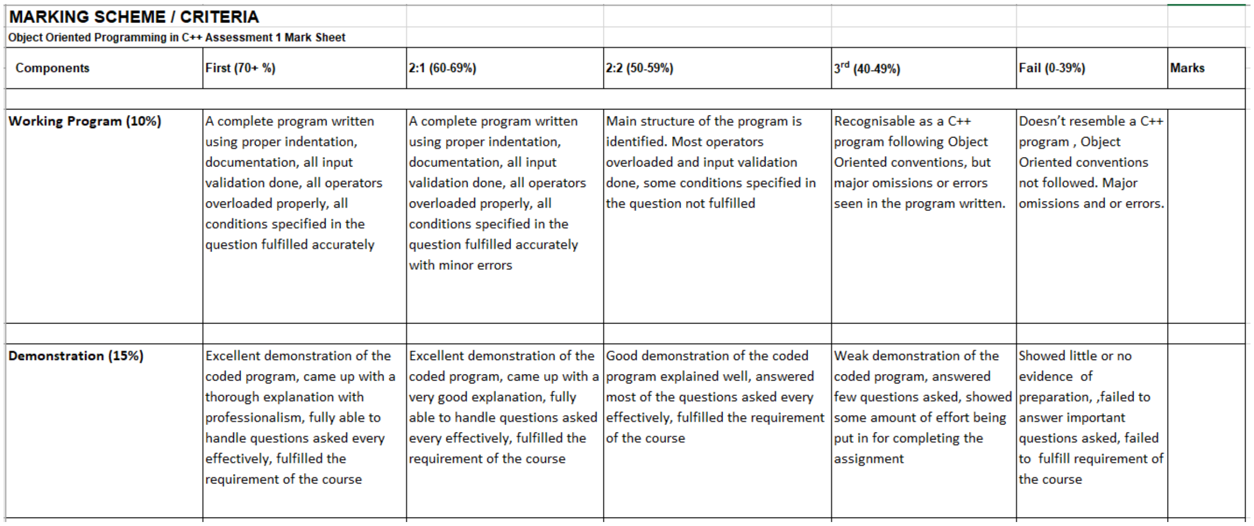
Each requirement you complete will be assessed via demonstration. An initial mark will be established during the demonstration, but this may be subject to later moderation by the module team.

Note: The mark awarded is primarily for the explanation of the implemented solution as well as the quality of the submitted code. Hence submission of a working program that satisfies the requirements does not in itself guarantee marks. Marks for a requirement will only be awarded if it fully complies with the specification, you are able to discuss your solution with the tutor in a knowledgeable way, and response to questions during the demonstration. Marks could be reduced for poorly presented code, inability to answer questions etc.

LATE SUBMISSION OPPORTUNITIES

If you fail to submit the electronic copy of your work in the Orbund by the specified date, then the usual University penalties for late submission will apply.

If you fail to demonstrate your work in the lab session following submission then you will be given one more chance to demonstrate and a late penalty of 5% will be applied. If you fail to demonstrate on your second opportunity however then will be recorded as **Non-Submission** for this this component.



MARKING SCHEME / CRITERIA

Object Oriented Programming in C++ Assessment 1 Mark Sheet

Components	First (70+ %)	2:1 (60-69%)	2:2 (50-59%)	3rd (40-49%)	Fail (0-39%)	Marks
Working Program (10%)	A complete program written using proper indentation, documentation, all input validation done, all operators overloaded properly, all conditions specified in the question fulfilled accurately	A complete program written using proper indentation, documentation, all input validation done, all operators overloaded properly, all conditions specified in the question fulfilled accurately with minor errors	Main structure of the program is identified. Most operators overloaded and input validation done, some conditions specified in the question not fulfilled	Recognisable as a C++ program following Object Oriented conventions, but major omissions or errors seen in the program written.	Doesn't resemble a C++ program , Object Oriented conventions not followed. Major omissions and or errors.	
Demonstration (15%)	Excellent demonstration of the coded program, came up with a thorough explanation with professionalism, fully able to handle questions asked every effectively, fulfilled the requirement of the course	Excellent demonstration of the coded program, came up with a very good explanation, fully able to handle questions asked every effectively, fulfilled the requirement of the course	Good demonstration of the coded program explained well, answered most of the questions asked every effectively, fulfilled the requirement of the course	Weak demonstration of the coded program, answered few questions asked, showed some amount of effort being put in for completing the assignment	Showed little or no evidence of preparation, ,failed to answer important questions asked, failed to fulfill requirement of the course	