EE402 OOP with Embedded Systems

The Task:

Your first assignment is to write a single body of code example that integrates and demonstrates the following aspects of OOP in C++:

1. **Inheritance** through the use of multiple classes. This should include multiple inheritance as one example, which does not have to make perfect sense in the real-world. (Please note Point 2.)
2. **Separate compilation** with all classes in your assignment (i.e., all classes should have .cpp and .h files).
3. **One abstract class** with an abstract method that is implemented in a child class. This should be integrated into your code in Point 1.
4. Correct and appropriate use of the **access specifiers** public, private, and protected with your classes.
5. **Passing an object** to a function by value and by reference. Demonstrate the impact of the change on the states of your object.
6. Calling a method on an object that is **const qualified** and passing the same object to a function by **constant reference**.
7. Correct use of **new and delete** for the allocation of an object/objects, with operations on the object using pointers.
8. Create an **array of pointers to objects** (minimum 3 elements) of one of your classes and pass this array to a function that displays all elements.
9. A simple **destructor** with some basic functionality in one of your classes. Demonstrate two examples of how it can be called.
10. Use of **dynamic binding** with virtual & non-virtual methods. Demonstrate the impact.
11. Correct use of three **overloaded operators** (e.g., +, = and ==) for one of your classes.
12. Implement a **non-member operato**r in one of your classes.
13. Write code to demonstrate an **object passing itself** to a function (that is not part of a class), which modifies the state of that object.
14. A class with a **modified copy constructor** and demonstrate the effect of this on pass-by-value and pass-by-reference calls.
15. Use of two different C++ **explicit style casts.**
16. Implement **static states** and **methods** in one of your classes and show example usage and impact.
17. Demonstrates the difference between a C++ class and a **C++ struct**.
18. Write a straightforward example **class template** using the examples in the notes.
19. Use of the **vector** container to contain a number of objects of one of your classes using the examples in the notes.
20. Use of an **algorithm** on your container using the examples in the notes.

There are 20 points to address and there is 10% available for the assignment, so 0.5% per point is the outline marking scheme. If you address 14 out of 20 points clearly and correctly then you should receive 7/10 (or 70% on Loop).

Implementation Points:

The topic is up to you and can be based on your work, experiences, interests, hobbies, etc., (with constraints as below). You should keep your implementation topic simple: Plan for three to five interrelated classes and make sure that each class is actually a class, not an object variant (remember the IS-A check).

You must use a single topic for all of the points. You CANNOT write an individual code example for each point. There should only be one main function and there is an upper limit of eight classes.

If one of the points does not fit into your application, just add it anyway for demonstration, i.e., the application does not have to make perfect practical sense. Note your reservations in the code comments.

You **must not** use any of the following topics:

- Bank accounts or any other accounts that use a balance, lodgement, account number type arrangements
- Personnel management structures where human name, ID, address, phone number are likely states (including university structures).
- Any example involving vehicles/cars/motors etc. (as per the notes)
- Geometric shapes or drawing examples (these are commonplace in textbooks.)
- Animal/plant/fish/insect/duck classifications.

I strongly recommend that you hard code the input of data, i.e., don't set up a user controlled menu system and do not read values in from the user using the standard input stream (cin). It is not necessary for this assignment and would lengthen the amount of work involved considerably.

Please note that you may have to comment out code to illustrate a point. That is fine but please retain the commented out code, particularly if it demonstrates a different point.

Report Structure

You must submit your assignment to Loop as a **single PDF document**. You are not required to submit code repositories/archives and you are not required to demonstrate that your code works. You are being graded on the structure of your code.

The report must be structured in the following way, with only the following text sections:

- **Student Information:** Assignment title, student name, email address, id number and programme name.
- **Introduction** (one paragraph only) - to briefly introduce your application and program structure. A diagram is permitted if required.
- **Assignment Implementation** - <u>one numbered paragraph for each of the points ordered as above</u>, which points to the location in the code where the point has been implemented – e.g., *Point 3 Abstract Classes – the class Account is an abstract class because it is missing the display()*

method implementation and this has been provided in the child CurrentAccount class. If you do not implement a point then label it and state "not implemented"

- **Source Code -** Properly Formatted and Spaced Code. The code should be pasted into the text document. Each file can be pasted in one after the other with its name listed at the top. The code should have comments referring back to the 20 numbered paragraphs e.g., in the CurrentAccount class, *// Point 3 Abstract Classes here with abstract method implemented.*

This report format is designed to be accessible by the tutors for marking purposes and numbering aligns against the marking rubric making feedback possible. **Failure to structure the document in this way will result in an automatic penalty of 5%-20%** (depending on severity) and may result in a further loss of marks due to lack of comprehension by the tutors.

Once again, you do not need to submit compilable or working code. This assignment is designed as a 'forced' study aid to encourage you to work through the notes and will be important to the final exam in January.

Submitting Your Work and Questions

This assignment is due for Tuesday the 26th of October by midnight.

Late submissions will be reduced in the following way:

- 5% (gross) reduction for 1 minute to 1 hour late.
- 10% (gross) reduction for 1 hour to 24 hours late.
- 10% (gross) additional deduction for each day late[1].
- No submission permitted 7 days after the assignment deadline.

Please note that you can submit your work multiple times before the deadline, which will help you avoid late penalties.

You should submit your assignment via Loop at the address: https://loop.dcu.ie/mod/assign/view.php?id=1661271

Once again, use the main discussion forum here: https://loop.dcu.ie/mod/forum/view.php?id=1661239 if you have any questions on the assignment. Do not paste your code! Paste in a similar example if you need to clarify a point. If you know the answer to someone else's question then please feel free to respond.

Plagiarism

You may work with others to understand the course notes and assignment instructions, but as soon as you work together (or with a third party) on the assignment itself then you are crossing the line.

You may be interviewed on your submission with or without cause.

[1] Negative marks limited to 0% -- e.g., a score of 40/100 will receive a mark of 0% after incurring a penalty for being 5 days late.

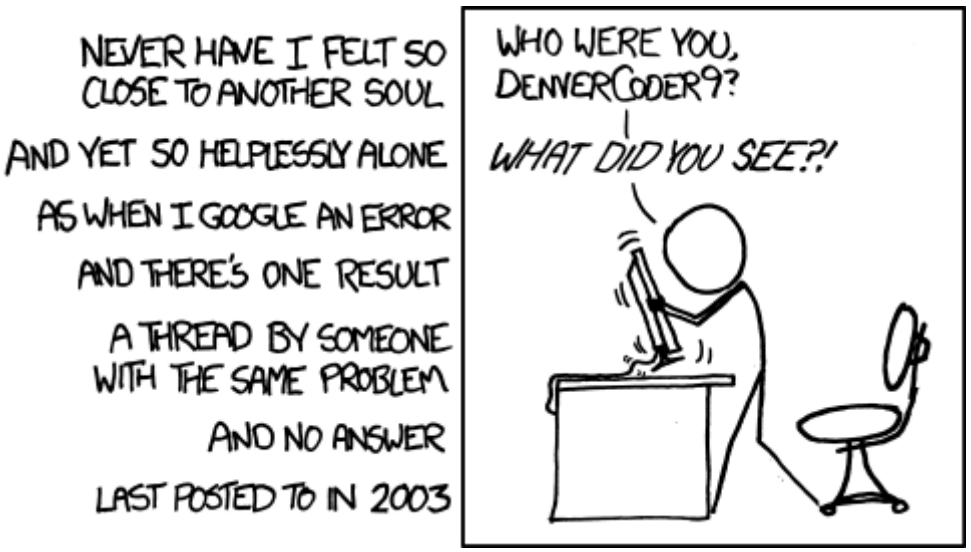
Please note that by the electronic submission of your assignment work I am assuming that you are familiar with the academic rules for assignments.

The assignment should be 100% your own work. If any unreferenced work is used from any other source (including colleagues, the internet, hired 3rd parties, or past students) it will be a serious matter and referable to the disciplinary board. Please note that all assignment work can be the subject of an interview process. See:

http://www.dcu.ie/info/regulations/plagiarism.shtml

All assignments for this module will be uploaded to plagiarism detection software where the content will be analysed and compared to other assignments from this year and previous years. I take this matter very seriously as a few years ago several students were referred to the disciplinary committee resulting from plagiarism in this module. In one case a student was expelled from the University due to the severity of the plagiarism and the fact that at fourth/Masters level, they should have known better.

Here is a XKCD comic you will understand by the end of the assignment!



Grading

If after you receive your mark you are concerned that there is an error with the correction, **please contact the module tutor that corrected your assignment in the first instance**. If the tutor checks it and you are still unhappy with their update/explanation then contact me directly and I will review the decision.

If I have failed to address any aspect of this assignment then please post your questions to the forum and not to me directly as others likely have similar questions. However, if you have a query/concern over your overall concept then send it to me directly.