COM326 Object-Oriented Programming

Week 11 – Session 1: File Input & Output

In this laboratory session you are going to explore File IO with C++.

Challenge 1: Student record system

Download the lab solution from week 9 from GitHub. In this lab we are going to add file access to our student record system so that we can write student details to a file. Naturally we would use a database such as MYSQL or SQL Server if we were developing this system for real. However, you should note that both of those DBMSs are written in C/C++ and both write data to a file. For illustrative purposes, the student record system will provide a fine scenario for you to develop technical competencies in file IO using C++.

Task 1 Details, details, details - duration 15 minutes

We will begin with a design task. Before we can start writing data to a file, we need to decide what data should be written and the format. With a classmate examine the solution for week 9 and the design of the student, module and lecturer classes. Consider the person, student and module classes, try to answer the following questions:

- What data should be written to file?
- How should the data be structured in the file?
- Do we need to include heading/titles within the file?

Task 2 Write student details - duration 30 minutes

Now that you have decided on what information you need store and its format/structure, create a function called:

```
WriteStudentDetails(std::vector<Student>& data)
```

The function should accept as a parameter a vector of student object and return a bool true/false if the details were written to file successfully.

Write the details for three student objects to a file called studentData.txt file. Ensure that each of the students has several modules associated with them.

Task 3 Read student details from file - duration 30 minutes

Write a function called:

```
std::vector<Student> ReadStudentDetails(std::string dataFile);
```

It should take the name of a data file as a parameter and read the contents back into the system constructing a vector of students complete with associated module marks. This vector should be returned by the function. In the main function you should print out the contents of the vector to confirm the details are correct.