

Object-Oriented Programming in C++

Prof. Caterina Doglioni and Dr. Charanjit Kaur

University of Manchester



The University of Manchester

PHYS3072 Assignment 2

Semester 2, 2023-24

Assignment 2: Physics Course Database

- For the provided course database (“**courselist.dat**” available on blackboard), write a C++ program to store and print physics courses, plus some statistical information.
- This assignment is based on material covered in the first three weeks. It has a separate goal for this week, which will be further developed in the second week.
- Please note that the total expected coding time is approx 6 hours.
- You should submit the final code. The submission deadline is **7 pm, 23rd February 2024**.

Part 1

- Write a program to read the data and compute the mean, standard deviation and standard error of the mean for the coursework marks.
- The code should determine the number of data entries in the file.
- You must either use **new** to allocate an array or if you know how to use to, use vectors as required for part 2.
- To remind you, the mean and standard deviation is defined as

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i, \quad \sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2}$$

and the standard error in the mean is $\sigma_{\mu} = \frac{\sigma}{\sqrt{N}}$.

Part 2

- Add the following functionality to the code:
 - Be efficient in storage. You now have to use vectors for storage.
 - Use a string stream to create a string containing the full course title e.g.

PHYS 30762 Object-Oriented Programming in C++

- Each of these strings should be stored in a single vector.
- Print out the full course list using an iterator.
- Your code should be able to print out a list of courses for a particular year (as identified by the first digit of the course code) including the mean and standard deviation for those courses.
 - For this, ask the user whether to print all courses or for a specific year and provide output accordingly.
 - Also, ask the user how courses should be sorted by title or by course code.

Marking Criteria

- There should be evidence of
 - A method to determine the number of records in the file (total and for the selection) (0.5 mark).
 - Read in data as an integer (for the course code) and a string (for the name) (0.5 mark).
 - Functions to calculate the mean and standard deviation of an array and/or vector of real numbers (1 mark).
 - A vector of strings to store course data (1 mark).
 - Use of string stream to combine the integer course code and title (1 mark).
 - An iterator to print out the information for each course (1 mark).
- The code should be able to print out a list of courses for a particular year, as identified by the first digit of the course code (1 mark).
- The code prints out a mean and standard deviation for the selection (1 mark).
- It should be possible to sort the list of courses by title or course code (1 mark).
- The code should be modular and should guide the user on what should be entered for different choices. (1 mark).
- Addition mark for the GitHub submission (1 mark).

Negative Marking

You will lose marks

- If the program does not compile without errors (-2 mark).
 - That includes warning messages.
- Does not adhere to house style (-1 mark).
- For not submitting the .cpp file(s) (and .h files if used) (-all marks).
- A further penalty for late submission (after 7 pm, 23rd February 2024).

	Not at all	Partially	Completely
Method to detemine records in file	0 (0.00%)	0.25 (2.50%)	0.5 (5.00%)
read data as integer and string	0 (0.00%)	0.25 (2.50%)	0.5 (5.00%)
Functions to calculate mean and standard deviation	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
Program contains and correctly implements vector to store course data	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
Program correctly implements a string stream (e.g.,to combine course code and title, or to read from string)	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
Program correctly implements an iterator to print information for all entered courses	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
program prints out list of courses for a particular year (identified by first digit of course code)	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
The code prints out a mean and standard deviation for the selection	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
Program sorts code by title and code	0 (0.00%)	0.5 (5.00%)	1 (10.00%)
No errors on compilation	-2 (-20.00%)	0 (0.00%)	0 (0.00%)
Correct use of house-style	-1 (-10.00%)	-0.5 (-5.00%)	0 (0.00%)
GitHub Submission	0 (0.00%)	0 (0.00%)	1 (10.00%)
Modular Code	0 (0.00%)	0.5 (5.00%)	1 (10.00%)