SIM UOW CSCI251

ASSIGNMENT 3

Last updated: 22 February 2024

Timothy Mah

8750634

Lecture Group L01

Contents

[Introduction 2](#_Toc159493304)

[Usage 2](#_Toc159493305)

[Program flow 2](#_Toc159493306)

[Extra features 2](#_Toc159493307)

[Input validation 2](#_Toc159493308)

[Main menu 2](#_Toc159493309)

[Sorting menu 2](#_Toc159493310)

[Limitations 2](#_Toc159493311)

[Learnings 2](#_Toc159493312)

[Compiler will omit trailing decimal point of a double if not needed 2](#_Toc159493313)

[Good diagram for reference 3](#_Toc159493314)

[Generic programming is like python? 3](#_Toc159493315)

[Function template vs function overloading 4](#_Toc159493316)

[if constexpr and is\_same\_v 4](#_Toc159493317)

[Cannot merge template functions and operator overloading together 4](#_Toc159493318)

[Any 5](#_Toc159493319)

[Thoughts and reflections 5](#_Toc159493320)

# Introduction

Hello this is my documentation for SIM UOW CSCI251 ASSIGNMENT 3.

# Usage

The final program file can only be run in Ubuntu because of the makefile. However, the main file can be compiled to support Windows by compiling as .exe.

To run,

./ csci251\_a2.app

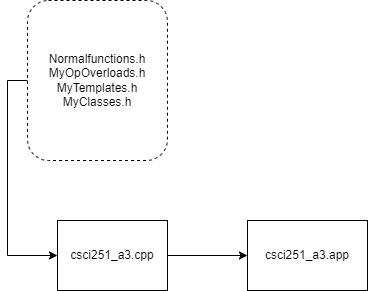
To compile,

g++ -std=c++2a ./csci251\_a3.cpp ./MyClasses.h ./MyOpOverloads.h ./MyTemplates.h ./NormalFunctions.h -o csci\_a3.app

# Program structure

## File structure

This is my program file structure.



Main() is in csci251\_a3.cpp. The program first enters here.

NormalFunctions.h is where most of the void functions that just print the menu, are.

MyOpOverloads.h is where I store my operator overloads. I did not put them in my class declarations. The logic for the operator overloads are declared as functions in the same file.

MyTemplates.h stores the other template functions

MyClasses.h is where my class declaration and implementation is. I did not split declaration and implementation.

## Notable functions

Some functions are quite straightforward especially the ones in NormalFunctions.h since they simply print the menu but there are other functions with a little more complex logic

### printObjects(const string& mode2, auto objects)

As its name suggests, this function prints the objects in the format specified in the assignment brief. This function is hwere the overloaded << are used

### sortObjects(vector<ObjectType>& objects, const string& mode3, const string& mode4)

This function is where all the sorting logic is. It is a template function that can perform sorting for any of the 4 object types. I used the if constexpr condition so that the correct sorting logic is performed for the corresponding object type. getScalarValue() was especially useful here since all types have this function with the same name

# Extra features

Please may I have more marks ☺

## Input validation

### Main menu

User can only input 1 -7. All else will be rejected and the program will not continue until valid user input is received. Program will continuously prompt for valid user input.

### Sorting menu

User can only input options show. All else will be rejected and the program will not continue until valid user input is received. Program will continuously prompt for valid user input.

# Limitations

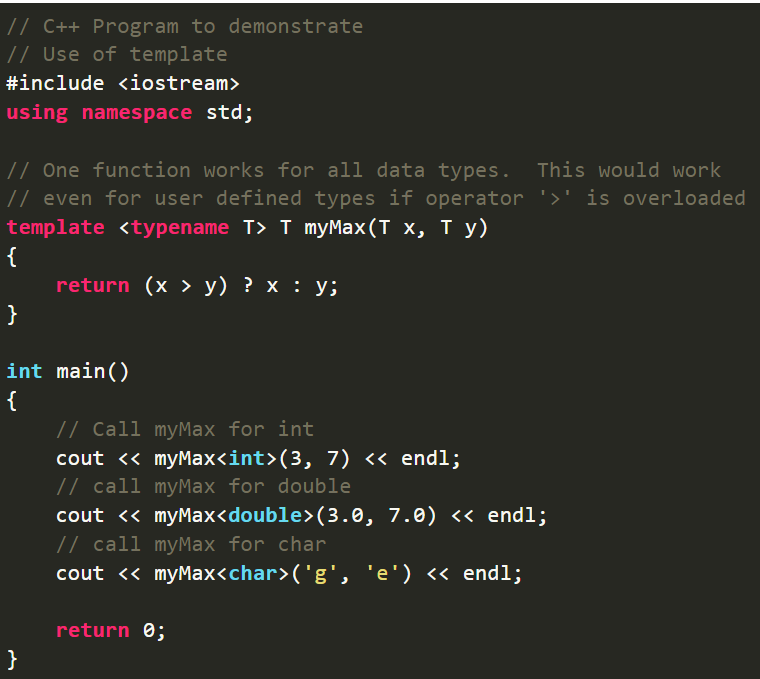
1. My program makes use of C++ std 20 features such as auto datatype in function parameters. Compiler must support latest C++ std to compile my program

# Learnings

Some cool stuff I learnt

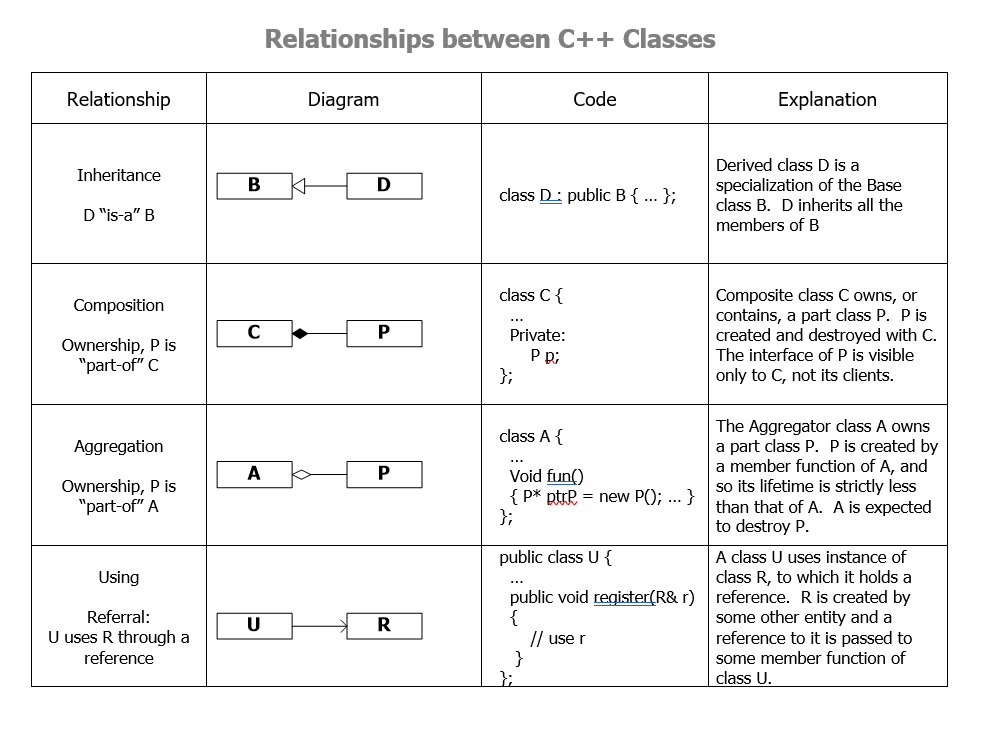
## Compiler will omit trailing decimal point of a double if not needed

I was reading up about function templates and I came across this code snippet



It’s a simple program to show the ability of function templates to pass datatype as a parameter when function is being called. I understand how it works but when I run this code snippet, the second cout output is 7. I had assumed it would be 7.0 since the return value is a double. However, upon further research, I discovered that it is because most C++ compilers, including g++ apparently, would omit trailing zeroes from a double when it is being cout. In the program the variable is stored as double yes, but the compiler will remove the trailing zeroes. I found this quite interesting behaviour

## Good diagram for reference

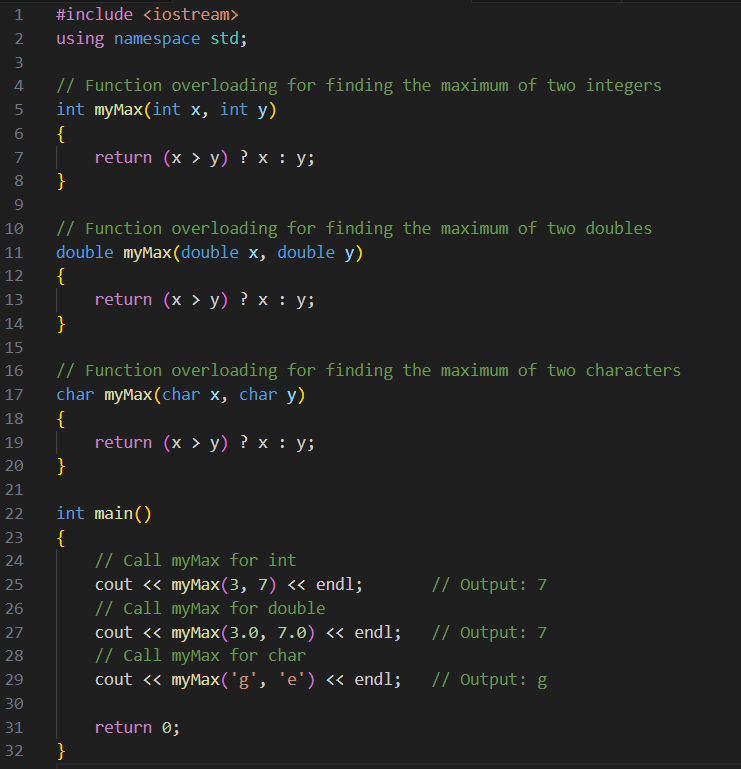


## Generic programming is like python?

Generic programming is an approach where generic data types are used as parameters in algorithms so that they work for variety of suitable data types. This is one component of the assignment. I suppose this is like python.

## Function template vs function overloading

This is how it would look like if the above code snippet about function templates, were to be re-written using function overloading

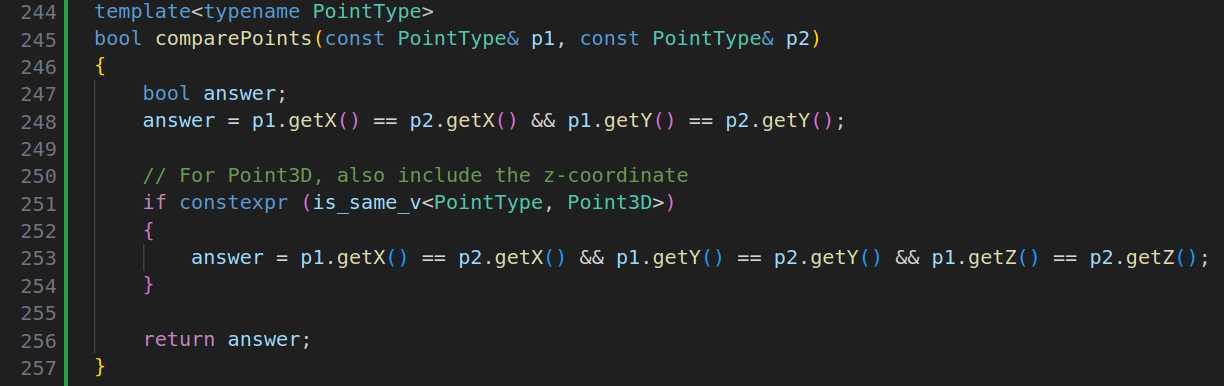


We can see that using templates is a cleaner solution since I don’t have to specify the function to accept the different data types as needed when performing function overloading.

This shows that although function templates and function overloading allows a function to behave differently based on parameter’s data type, function templates allow any data type, while function overloading requires the function declaration/ implementation to specify every scenario of data type to work.

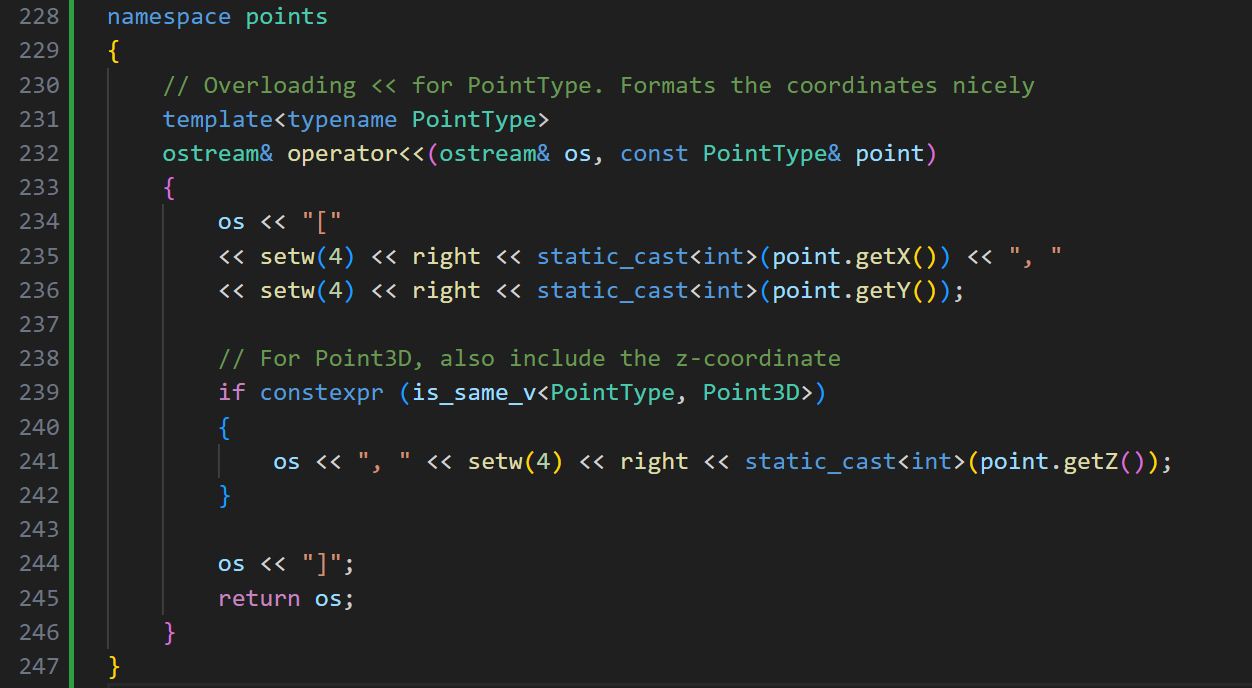
## if constexpr and is\_same\_v

These components allow me to create if statements within my template function to perform differently when a certain object type is used as template



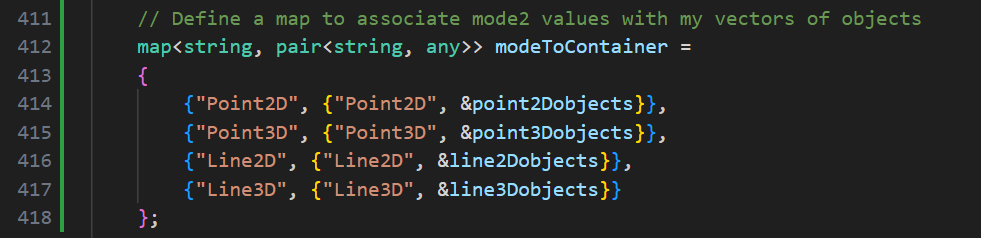
## Cannot merge template functions and operator overloading together

Initially I wanted to write my function something like this. This function is an operator overload for << and it uses template as well. After many hours of debugging I realised that the problem was not so much with the code, but my system design. I realised that if my << overload takes in a template as parameter, it basically can take it any value thus it also has the same signature as a normal <<.



## Any

Any is a vector like container that can hold any datatype. It’s part of STL.



Especially useful when I wanted to store all my objects with the different class types. It’s Python!

# Thoughts and reflections

3 assignments back to back is quite tiring ☹ but I have friends in other uni that have 4-6 modules per sem so they have twice my workload and yet I only have 2 per sem and I feel tired. Perhaps I am capable of more. Perhaps I am a slow learner. I don’t know but I learnt a lot from this module.