Scientific Programming

C++ Classes Summarized

Anand Kamble
28th November 2023

---

1 Introduction

In the following paper, we will provide a comprehensive overview of key concepts, namely classes, friendship, inheritance, and polymorphism.

---

2 Classes

2.1 Overview

One of the most important concepts in any programming language is object-oriented programming, which allows us to write dynamic code. Classes are like advanced versions of data structures. They not only hold data but also include functions, which are also referred to as 'methods of a class'. Also, the data or the functions inside a class are referred to as members of the class, and these members can be private or public depending on the developer's choice. I have included a class as an example in the file 'Sparse.cpp'.[2]

2.2 Constructors

A class has a constructor which is called when a class is created, these are the functions that initialize a class member or allocate required memory. Compilers do create a default constructor if no constructor is provided by the programmer. These constructors can also be overloaded in the same way functions are overloaded. Also, there are copy constructors that handle the copying of member data if the class is copied from one variable to another. An example of this can be found in Sparse.cpp at lines 32 and 34.

2.3 Destructor

This is a function that is called when the class object is to be destroyed, usually when it goes out of scope. It handles deallocating the memory and other clean-up tasks.

2.4 Pointer to a class

Since, classes are like blueprints of an object, when a class is initiated it creates an object which includes all the members defined in the class. These objects can be pointed to by the pointers too. And can be stored in different types of data structures like vectors or linked lists.

2.5 Operator Overloading

C++ is a typed language and by defining a new class, we are defining a new data type that can be manipulated using operators. In order to be able to use operators such as addition (+), and multiplication(*) we have to overload the operators by defining a method that will be executed when a certain operator is used on the class. An example of operator overloading is in Sparse.cpp at line 98.

2.6 Templates

We can use templates inside a class to make it dynamic-typed. This is called a template specialization. This allows us to generalize a class to be used with various types. For example, if we used templates with the 'SparseMatrix' class, we can specify it to use floats or doubles rather than integers to store the matrix elements.

2.7 Friendship and inheritance

Friends of a function are the classes or functions which have access to all the members of the class including private ones. This allows us to access or modify the members of a class from outside of the class. These can be useful when we want two class objects to interact with each other, for example overloading the operators.

Inheritance is an important concept of classes and object-oriented programming (OOP) where one class is built on top of another and it inherits all the members of the base class except the constructor, destructors, friends, assignment operator members, and private members. This helps us to keep the code organized and modular and prevents rewriting of members that are used at different parts of the program. This concept becomes very useful as the programs become more complex and there are multiple developers working on the same project.

2.8 Polymorphism

The typed nature of C++ can introduce some problems when we create new objects or modify the existing ones. To overcome this we can use polymorphism, which allows us to treat various types of objects as a common type.[1]

References

[1] Tony Delroy (https://stackoverflow.com/users/410767). Polymorphism in c++. Stack overflow. URL:https://stackoverflow.com/a/5854862/22647897 (version: 2020-02-28).

[2] Abhirav Kariya. C++ classes and objects. `https://www.geeksforgeeks.org/c-classes-and-objects/`, 2023. Accessed on November 29, 2023.