Some Interesting Topic In Classes

Saurav Samantaray

Department of Mathematics

Indian Institute of Technology Madras

October 28, 2024

Header Files

- It doesn’t matter if we include header files such as `iostream`, `string`, etc. more than once.
- But we should be very careful not to include files such as `book.hpp` in the form that it was written more than once, as this can cause problems.
- When we are working with several different classes it is easy to inadvertently include header files more than once.
- To avoid this code being included twice, we adapt it so that the header file for a class called `BOOK` is of the form

```
#ifndef BOOKHPP
#define BOOKHPP
#include <iostream>
#include <string>
using namespace std;
class Book
...
#endif
```

Copy Constructor

- The default constructor is automatically generated by the compiler for any class.
- There is another type of constructor that is automatically generated, the *copy constructor*.
- This constructor requires as input another instance if the class, and creates a copy of this instance.
- Like with the default constructor, the copy constructor can also be overridden.
- The argument to a copy constructor has to be a reference to another instance of the class, rather than that object itself.
- It is also a good practice to declare the argument to a copy constructor as a `const`.
- More on copy constructor later!!

The `friend` Keyword

- When developing a program, we may wish to access private members of a class from outside the class.
- On way of doing this is to create a new public method that accesses the private member.
- Another way is to write a free function that is a *friend* of the class:
- such functions may access all members of the class, including private variables.
- The friend function is not a member of the class
- It is not defined using a :: operator

Pointers to Classes

We may declare a pointer to an instance of a class.

```
#include ”book.hpp”
int main ()
{
   Book* p_book_i_am_reading = new Book;
   (*p_book_i_am_reading).author = "philip pullman";
   p_book_i_am_reading->title = "Lyras Oxford";
   delete p_book_i_am_reading;
}
```

- We declare a pointer, `p_book_i_am_reading`, to an instance of the class `Book`, and
- allocate memory for this instance through the use of `new`.
- `*p_book_i_am_reading` is used to denote the contents o the memory whose address is stored by the pointer.
- By placing this in brackets we can access the class members.
- the next line the arrow notation access is a more convenient way of accessing a class member associated with a pointer to a class
- in this the forward arrow ”->” means de-reference and then access the member.