

Programming Method

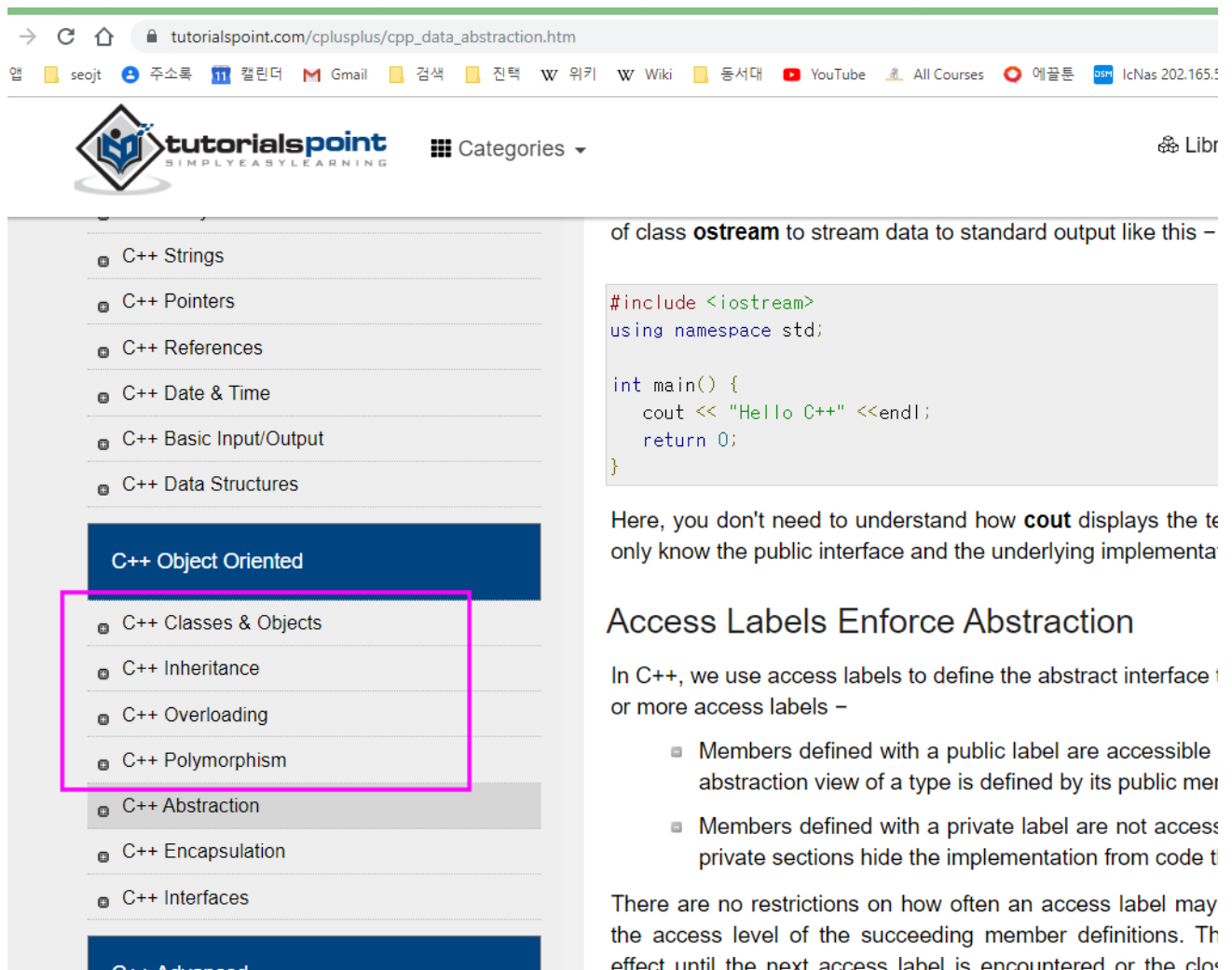
Midterm Exam Range

Date: October 22(Tue), 2019, 12:00 pm

1) Read 4 chapters of C++ tutorial at tutorials point

Link: https://www.tutorialspoint.com/cplusplus/cpp_data_abstraction.htm

- C++ Classes & Objects
- C++ Inheritance
- C++ Overloading
- C++ Polymorphism



The screenshot shows the TutorialsPoint website interface. The top navigation bar includes the site logo, a search bar, and various utility links. The main content area is divided into a left sidebar and a right main panel. The sidebar lists various C++ topics, with 'C++ Object Oriented' highlighted in blue. Under this category, four items are listed and enclosed in a pink rectangular box: 'C++ Classes & Objects', 'C++ Inheritance', 'C++ Overloading', and 'C++ Polymorphism'. The main panel on the right contains text about the `ostream` class and a code snippet for using `cout` to print 'Hello C++'. Below the code, there is a section titled 'Access Labels Enforce Abstraction' which explains how access labels (public, private) are used to define the abstraction view of a type.

of class **ostream** to stream data to standard output like this –

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello C++" << endl;
    return 0;
}
```

Here, you don't need to understand how **cout** displays the text, you only know the public interface and the underlying implementation.

Access Labels Enforce Abstraction

In C++, we use access labels to define the abstract interface and to enforce more access labels –

- Members defined with a public label are accessible to all code. The abstraction view of a type is defined by its public members.
- Members defined with a private label are not accessible to other code. Private sections hide the implementation from code that uses the type.

There are no restrictions on how often an access label may be used. The access level of the succeeding member definitions. The effect until the next access label is encountered or the closing brace of the class definition.

2) Understand source files listed below:

Link: <https://github.com/GP101/Programming/tree/master/CppProgramming>

ConsoleApplication1 inheritance.cpp

ConsoleApplication1 reference.cpp

CppOcw copy constructor01 crash.cpp

CppOcw copy constructor02 crash solution.cpp

CppOcw_operatoroverloading01.cpp

CppOcw_operatoroverloading02 ostream.cpp

CppOcw_virtual01_virtual function.cpp

CppOcw_virtual02_virtual in destructor.cpp

CppOcw_virtual03_vtable.cpp

github.com/GP101/Programming/tree/master/CppProgramming

주소록 캘린더 Gmail 검색 진택 위키 Wiki 동서대 YouTube All Courses 에플론 bsm IcNas 202.165.58.12...

Why GitHub? Enterprise Explore Marketplace Pricing Search

GP101 / Programming

<> Code Issues 0 Pull requests 0 Projects 0 Security Insights

Branch: master Programming / CppProgramming /

jintaeks Added C++ copy constructor sources. ...

..	
ConsoleApplication1 inheritance.cpp	Added inheritance source.
ConsoleApplication1 lookuptable.cpp	Added lookup table sources.
ConsoleApplication1 lookuptable2 state machine...	Added lookup table sources.
ConsoleApplication1 reference.cpp	Added inheritance source.
CppOcw copy constructor01 crash.cpp	Added C++ copy constructor sources.

@