COMP 322 Lecture 2 - C++ Basics

Junji Duan

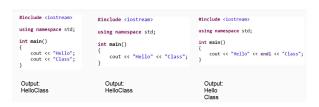
2024/1/12

Today's Outline

- Standard input & output
- Namespaces

Standard input/output

- C++ uses "streams" for reading from (input) and writing to (output) a media
 - Media can be a keyboard, screen, file, printer, etc.
- Input and output streams are provided by the iostream header file
 - include <iostream>
- cout stream object is used to print on screen
 - cout « "some message";
 - «: insertion operator
 - cout: object of ostream class
- Default standard output is the screen
- Similar to printf() in c, system.out.println() in java



- cin stream object is used to read from the keyboard
 - $\sin x;$
 - »: extraction operator
 - cin: object of istream class

- Cin can read strings but limited to one word
 - cin » stringVariable;
- Use getline function to read a full sentence
 - getline(cin, stringVariable);
- Similar to scanf() in c, scanner class in java

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

int main()
{
    string var;
    cout << "Please enter your name" << endl;
    cin >> var;
    cout << "your name is: " << var;
}</pre>
```

Namespaces

- A name can represent only one variable within the same scope
- Large projects consists of multiple modules of code provided by different programmers
 - What happens if one module has a variable name that is the same as another variable in different module?
 Name conflict (also called name collision)
- Namespaces solve the name conflict problem

```
QuebecTemp.h

main.cpp

double getTemp()
{
    foulde getTemp()
    four a30.7;
}
}

Or also: main.cpp

#include <iostream>
#include *QuebecTemp.h*

int main() {
    std::cout < "Temperature is: " << QC::getTemp() << std::endl;
    return d;
}

Or also: main.cpp

#include *QuebecTemp.h*

using namespace QC;

int main() {
    std::cout < "Temperature is: " << getTemp() << std::endl;
    return d;
}
```