



Tutorial 2: C++ Basics

C++ Basics

Prof. Sangyoung Park

Module "Vehicle-2-X: Communication and Control"



- Header file (*.h)
 - Contains declaration of variables and functions
- CC file (*.cc or *.cpp)
 - Contains the actual definition and C++ code
 - If you want to make use of the variables and functions in a cc file, you need to use “#include” directive to include the header files

How would you code?

- On Windows platform, for now let's use an editor of your selection
 - For example, I used sublime editor
- Editors come with coloring schemes such as the follows
- This makes the code convenient to read, easy to spot syntax errors (typos), etc.

```
[#include <stdio.h>

class Vehicle{
public:
    Vehicle(double, double);
    double getVelocity();
    void setVelocity(double velocity);
    double m_velocity;
    void proceedTime(double time);
    double getLocation();
private:
    double m_x;
};
~
~
~
~
```

Divide the Car.cc file

- Let's divide the car.cc file into a header file and a cc file
- First, create car.h file in the editor of your selection
- Which part is the declaration? In the original car.cc file?
 - Answers given on the right
 - Declarations don't contain the actual definitions

```
[#include <stdio.h>

class Vehicle{
public:
    Vehicle(double, double);
    double getVelocity();
    void setVelocity(double velocity);
    double m_velocity;
    void proceedTime(double time);
    double getLocation();
private:
    double m_x;
};

class Pkw: public Vehicle{
public:
    Pkw(int);
    void setPassenger(int passenger);
    int getPassenger();
private:
    int m_passenger;
};
~
~
~
```

How does the .cc file recognize the declarations?

- You need to use "#include"
- Try typing #include "car.h" in the car.cc file
- Try also later, what happens if you don't have this line

```
#include <stdio.h>
#include "car.h"

Vehicle::Vehicle(double initVelocity, double initLoc){
    m_velocity = initVelocity;
    m_x = initLoc;
}

double Vehicle::getVelocity()
```

Create a separate main.cc file

- Let's create a separate main.cc file from your editor
- Let's just copy the main function to this file
- Don't forget to #include "car.h"
 - Otherwise you won't be able to use Pkw in this file

```
#include <stdio.h>
#include "car.h"

int main(void)
{
    Pkw pkw = Pkw(4);
    printf("Num of passengers: %d\n", pkw.getPassenger());
    Pkw* pkw_p = new Pkw(5);
    printf("Num of passengers: %d\n", pkw_p->getPassenger()); // -> operator is used if pointer is used
    return 0;
}
~
~
~
```

How do we generate an executable?

- Now, we want to generate an executable from multiple files, how?
- Do you remember compile -> link from the last tutorial?
- Compile
 - `>> g++ -c main.cc car.cc`
 - Above line compiles to cc files without linking
 - The command will generate two files named `main.o` and `car.o`
 - These are called “object files”
 - Machine codes, but not yet executable, because it doesn’t know where `printf()`, etc. are
- Link and generate executable
 - `>> g++ main.o car.o`
- Output executable `a.out` or `a.exe` is generated
 - You can check the same results by typing `./a.out`

But you don't have to do this manually

- The purpose of this tutorial was to help you get familiarized with the C++ header files and cc files
- Veins simulator will contain thousands of header files and cc files and you can't type g++ command every time
- It's automatically handled by the tools
- It's much easier and you'll be familiar with them soon

C++ Programming Task of your own: LKW

- Could you try to implement a class called “Lkw”, which inherits the Car class, and has a new protected variable ”double payload”?
- Implement it in two new files lkw.cc and lkw.h
- Implement a member function named “setPayload(double payload)”
- Implement a member function named “double getPayload()”
- Modify the main() function in the main file such that it generates two objects of Lkw using the “new” operator
- Set the payload value and print it using cout function
- Hints
 - #include “car.h”
 - #include “lkw.h”
 - g++ -c main.cc car.cc lkw.cc
 - g++ main.o car.o lkw.o
 - ./a.out

