**Splitting class in separated files**

The main objective of OOP to make the code clean, or clean architecture. For this reason, embedding the class with main function is not practical. In this tutorial, we will work to build clean code by split class in separated files.

Partial classes are useful in several situations. On a large project, for example, separating a class into multiple files enables more than one programmer to work on the project at same time.

Objective:

* Build C++ project with classes in separated files.

There are three types of files:

1. Headers (.h files)

Contains the declaration of the whole class members.

Contains only attributes declaration and function prototypes

1. Implementation files (.cpp files)

Contains the implementation of class methods (functions)

1. Client code

Contains the main function (.cpp)

Let’s start by the following example:

* Create your project, name it testspeed.
* Copy the following code to testspeed.cpp

#include <iostream>

using namespace std;

class car

{

private:

int maxSpeed = 0;

public:

void setspeed(int s)

{

maxSpeed = s;

}

int getspeed()

{

return maxSpeed;

}

};

int main() {

car myObj;

myObj.setspeed(200);

cout << myObj.getspeed();

return 0;

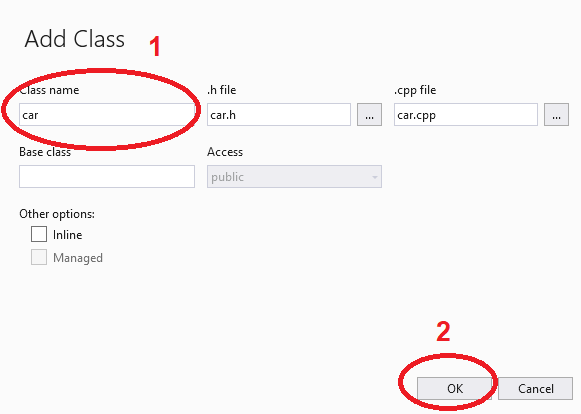
}

* Run you project

Output:

Now, we will start to split the above class into separate class:

* In your project, create new class (project – add class)



* notice that two files will be created in your project (car.h, car.cpp)
* in car.h, copy car class including attributes and function prototypes:

#include <iostream>

using namespace std;

class car

{

private:

int maxSpeed = 0;

public:

void setspeed(int s);

int getspeed();

};

* In car.cpp, copy the method implementation. Notice that you need to include car.h, and add class name (car) before each method with scope operator (::)

#include "car.h"

#include <iostream>

using namespace std;

void car::setspeed(int s)

{

maxSpeed = s;

}

int car::getspeed()

{

return maxSpeed;

}

* Last step (client code), remove car class from testspeed.cpp, and include car.h file. As follow:

#include "car.h"

#include <iostream>

using namespace std;

int main()

{

car myObj;

myObj.setspeed(200);

cout <<"the car speed is: "<< myObj.getspeed();

return 0;

}

Notice that you can define more than one class in the header file

Class one

{

----

----

};

Class two

{

----

----

};

And in the implementation file, you will need to add the clas name and scope operator with each method

one::method1() { …. }

one:method2(){ ---}

two:method1(){---}

and so on.

Also, do not forget to include header file in both implementation file and client code.

**Exercise**:

* Rebuild your code in lab2 by split your class in separated files.

That is the end of labsheet.. Good Luck