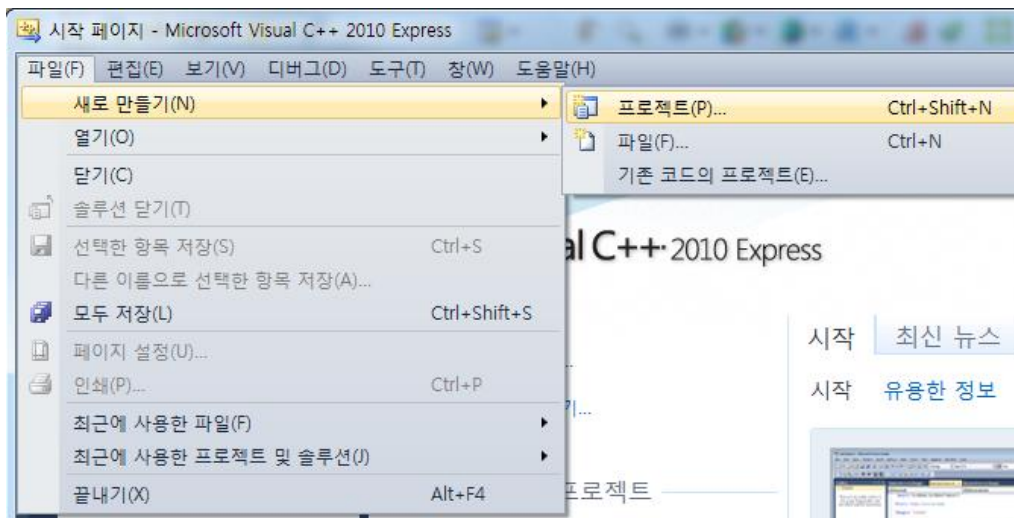


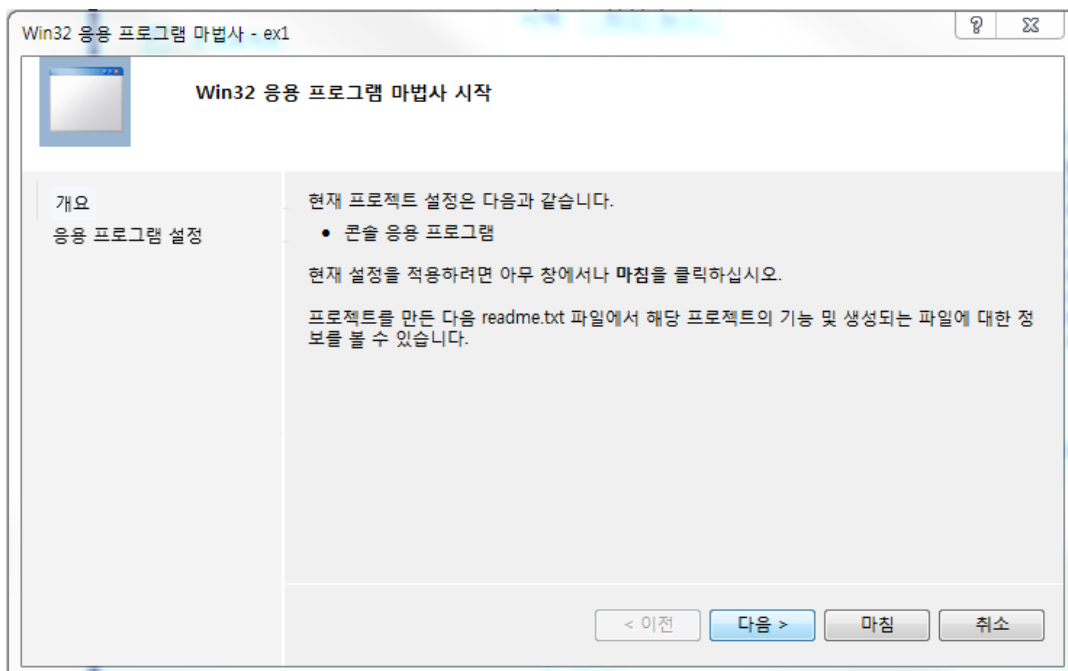
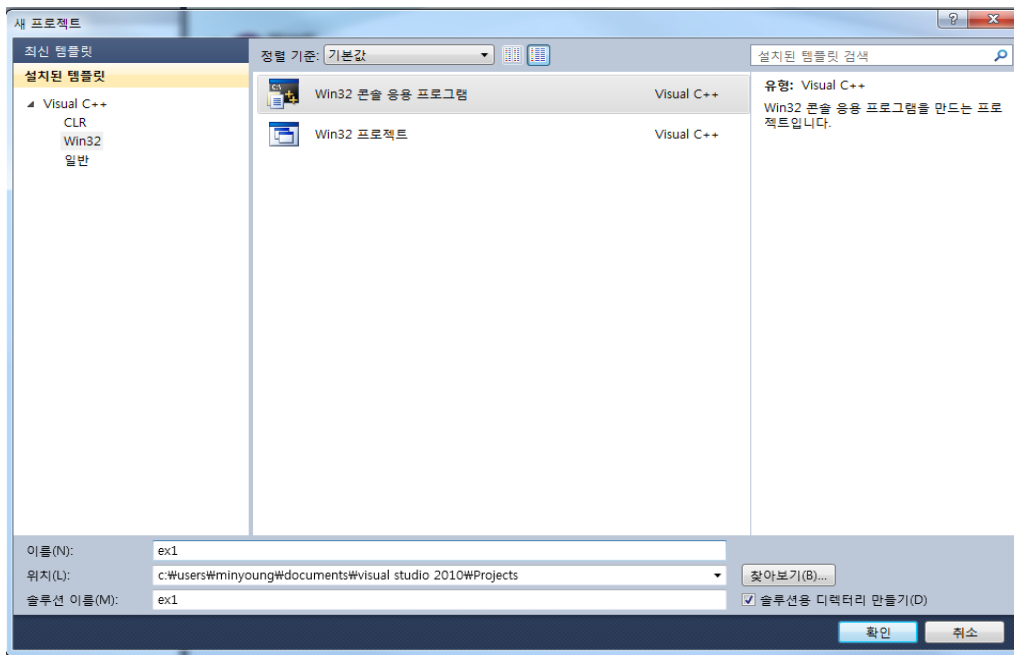
# Week 1 : Introduce Programming Environment

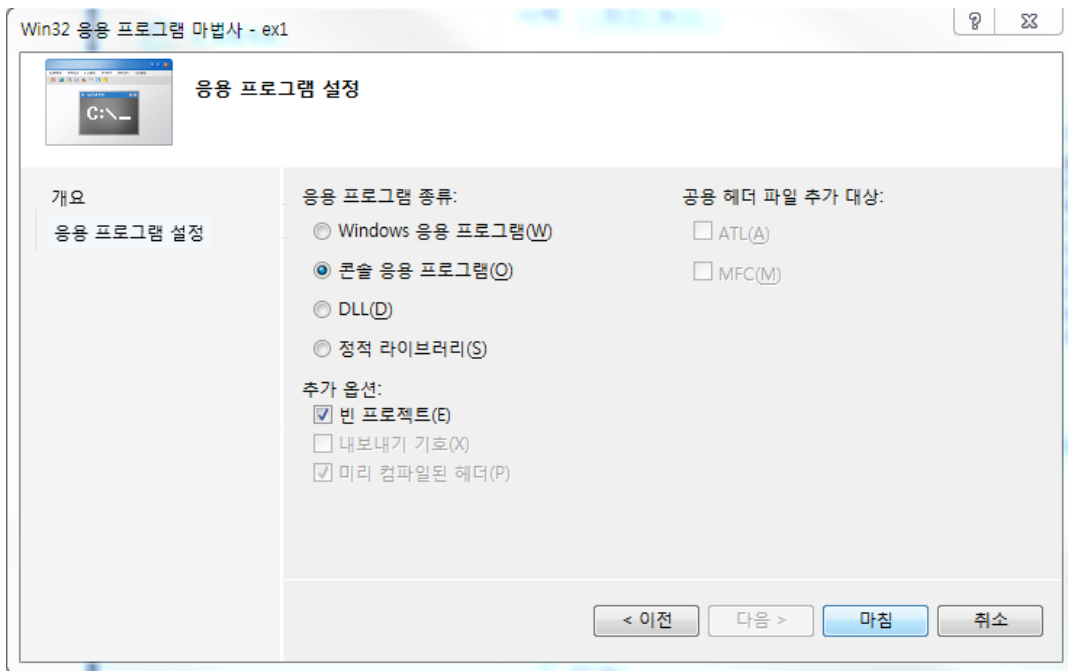
- C++ Programming Environment
- Visual Studio for C++ Programming

## 1. Generating a project

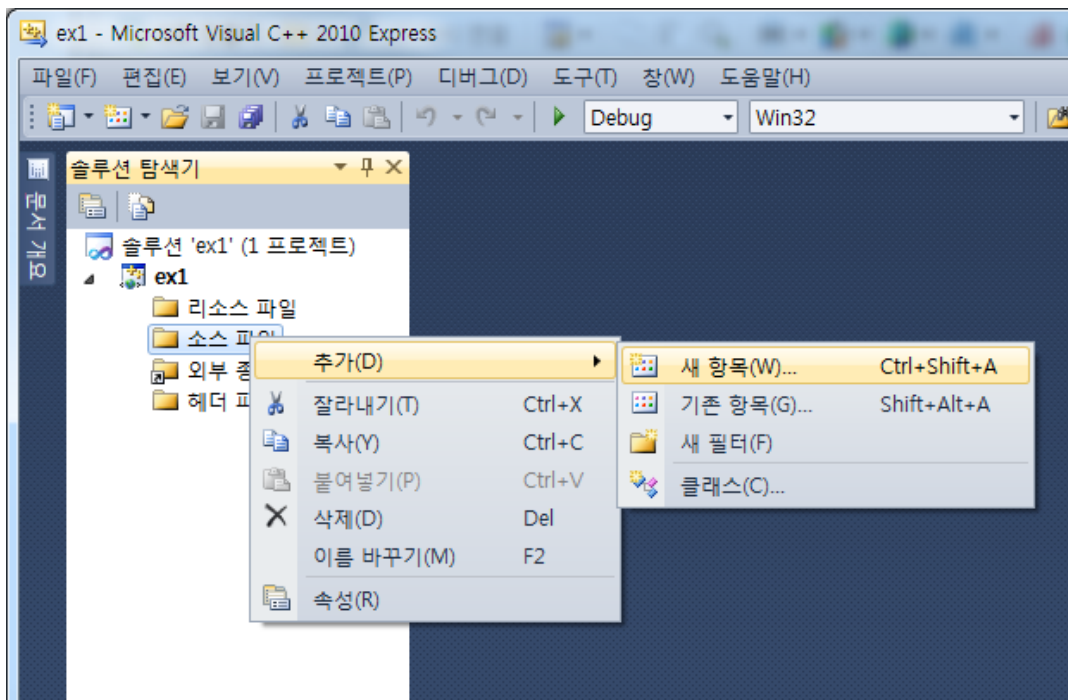


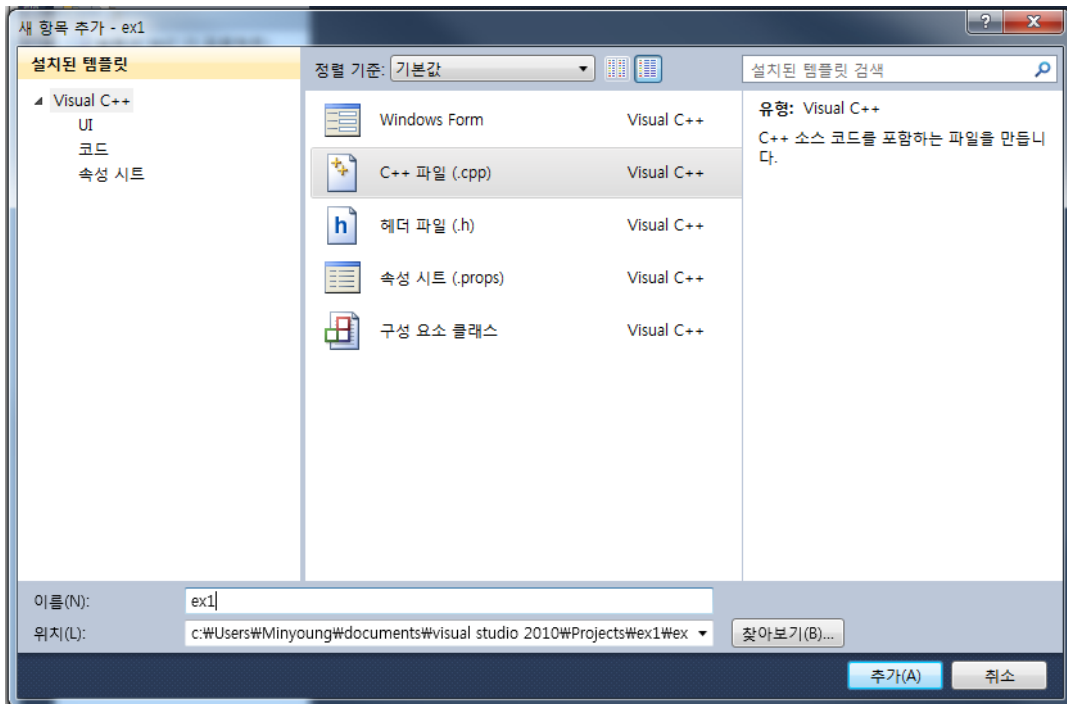
M1522.000600 Computer Programming  
(2015 Spring)





## 2. Writing a code and execute it





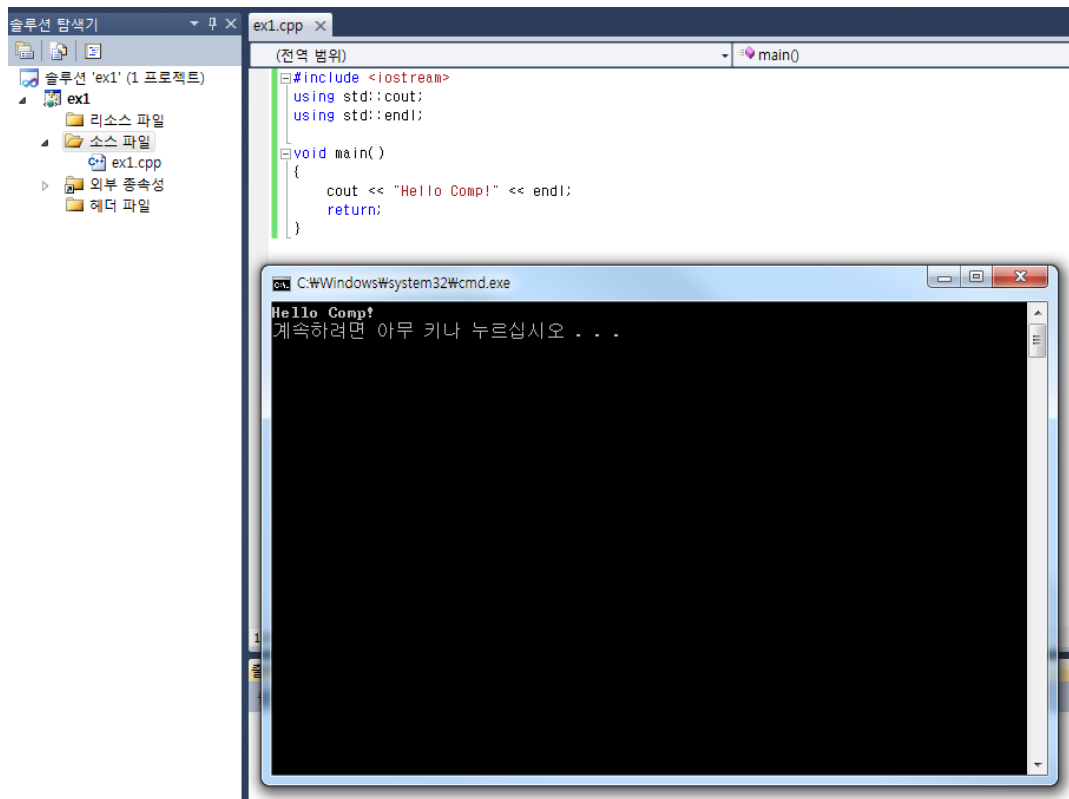
Template : C++ file / writing a file name.

Writing a code and execute it

→ Compiling : F7 / Compiling and executing: F5

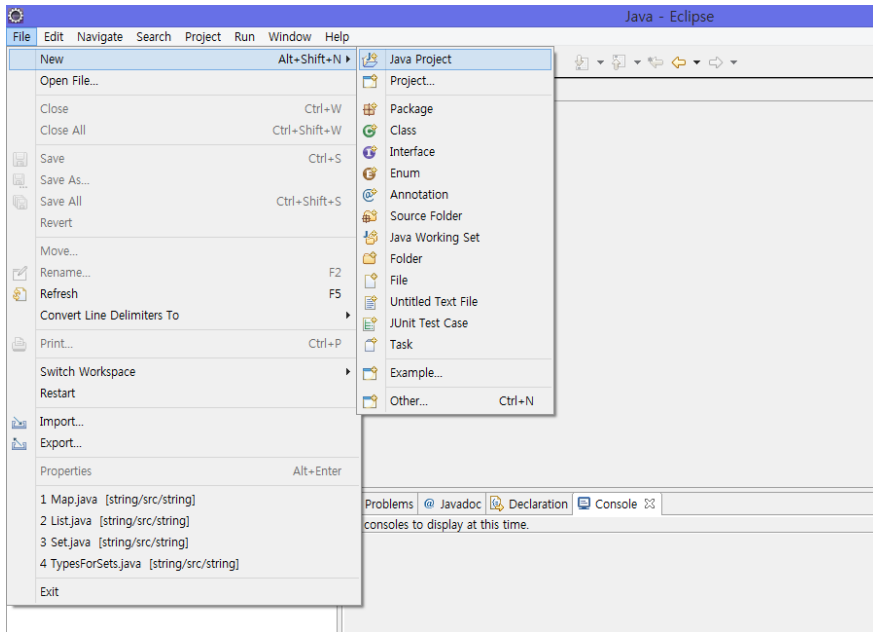
→ Non-debug : Ctrl + F5

M1522.000600 Computer Programming  
(2015 Spring)

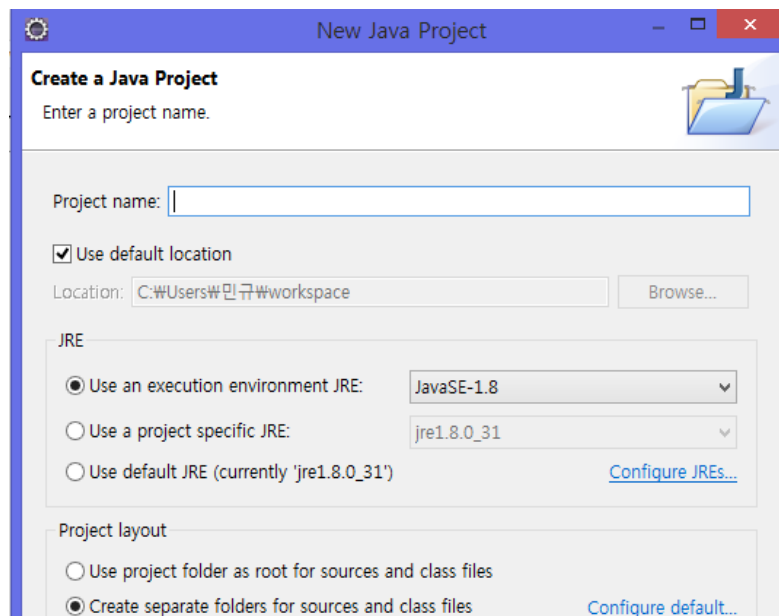


- Java Eclipse Programming Environment

### 1. Generating a project

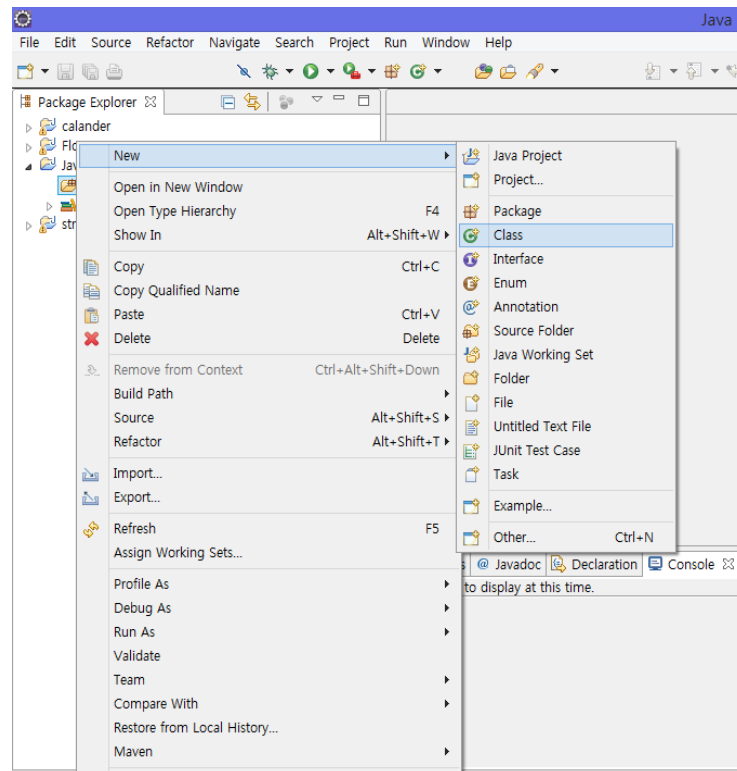


File > New > Java Project



Writing a project name > Finish

## 2. Writing a code and executing it.



Click the right mouse button at the generated project > New >

Class

If you want to bind many classes to a package, make a package first, after that make a class.

Source folder:

Package:

Enclosing type:

---

Name:

Modifiers:  public  package  private  protected  
 abstract  final  static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)  
 Constructors from superclass  
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Writing a class name > Finish

Executing : ctrl + F11

Automatic completion : ctrl + space

Automatic alignment : select a scope and ctrl + shift + f

make a comment : select a scope and ctrl + shift + c



## Example1

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

class Item { // Class definition
public:
    string title;
    double price;
    double SalePrice() { return (price*0.9);}
    bool isAvailable() { return (inStockQuantity > 0); }
private:
    int inStockQuantity;
};

int main(void)
{
    Item a;
    a.title="comp";
    a.price=2000;

    cout << a.title <<endl;
    cout << a.SalePrice() << endl;

    return 0;
}
```

## Example2

```
#include <iostream>
#include <string>
#include <assert.h>
using namespace std;

class String {
public:
    String(const char *s) {
        len = strlen(s);
        str = new char[ len + 1];
        assert(str != 0);
        strcpy(str,s);
    }
    ~String() { delete [] str; }
    void showStr()
    {
        cout<<str<<endl;
    }

private:
    int len;
    char *str;
};

int main(void)
{
    String str = String("str"); // Definition
    str.showStr();

    return 0;
}
```