

# **Week 8**

Computer Programming Lab

2020/11/03

# Recap: Header File

➤ MyLib.h `bool isEven(int number);`

➤ MyLib.cpp

```
bool isEven(int number) {  
    return (number % 2 == 0);  
}
```

```
#include <iostream>  
#include "MyLib.h"  
  
using namespace std;  
int main() {  
    cout << isEven(4) << endl;  
    cout << isEven(5) << endl;  
    return 0;  
}
```

➤ UseMyLib.cpp →→

➤ **MyLib.cpp** and  
**UseMyLib.cpp** in one project.

# Run multiple cpp file in VScode

- Modify the command in task.json

```
1  {
2    // See https://go.microsoft.com/fwlink/?LinkId=733558
3    // for the documentation about the tasks.json format
4    "version": "2.0.0",
5    "tasks": [
6      {
7        "label": "echo",
8        "type": "shell",
9        "command": "g++ -g UseMyLib.cpp MyLib.cpp",
10       "group": {
11         "kind": "build",
12         "isDefault": true
13       }
14     ]
15  }
16 }
```

# To show all the warning: -Wall

- Turns on (almost) all the warnings
- e.g.
  - unused variable, uninitialized... etc

```
main.cpp:5:12: warning: unused variable 'j' [-Wunused-variable]
    int i, j;
           ^
main.cpp:6:5: warning: variable 'i' is uninitialized when used here [-Wuninitialized]
    i++;
    ^
main.cpp:5:10: note: initialize the variable 'i' to silence this warning
    int i, j;
        ^
        = 0
2 warnings generated.
```

# Add -Wall in VScode

- Modify the command in task.json

```
1  {
2      // See https://go.microsoft.com/fwlink/?LinkId=733558
3      // for the documentation about the tasks.json format
4      "version": "2.0.0",
5      "tasks": [
6          {
7              "label": "echo",
8              "type": "shell",
9              "command": "g++ -g UseMyLib.cpp MyLib.cpp -Wall",
10             "group": {
11                 "kind": "build",
12                 "isDefault": true
13             }
14         }
15     ]
16 }
```