Framework

- The following program is the framework for processing data in the input stream and putting processed data to the output stream.

Framework

```
// framework.h
#ifndef FRAMEWORK_H_INCLUDED
#define FRAMEWORK_H_INCLUDED
struct Filter
{  virtual void start() { }
   virtual int read()=0;
   virtual void write() { }
   virtual void compute() { }
   virtual int result()=0;
   virtual ~Filter() { }
};
#include "yourApp.h“
int main_loop(Filter* p);
#endif // FRAMEWORK_H_INCLUDED
```

```
// framework.cpp
#include <iostream>
#include <fstream>
using namespace std;
#include "framework.h"
int main_loop(Filter* p)
{    p->start();
     while (p->read())
     {    p->compute();   p->write();   }
     return p->result();
}
```

```
//test.cpp
#include <iostream>
#include "framework.h"
using namespace std;
int main(int argc,char**argv)
// argv[1]: gradeIn File,                 argv[2]: gradeOut File
// argv[3]: studentCode File,             argv[4]: courseCode File
{
    ifstream fin(argv[1]);      // file to be processed
    ofstream fout(argv[2]);   // file processed
    if (!fin || !fout) cerr << "Error opening file!" << endl;
    Your_filter f(fin,fout,argv[3],argv[4]);
    return main_loop(&f);
}
```

Framework

```
// yourApp.h
#ifndef YOUAPP_H_INCLUDED
#define YOUAPP_H_INCLUDED
#include "framework.h"
class Your_filter: public Filter
{  // TODO :   your data representation
   // TODO :   your function prototypes
};

#endif // YOUAPP_H_INCLUDED
```

Framework

```
// yourApp.cpp
#include "yourApp.h"
// TODO:   your function definitions
```

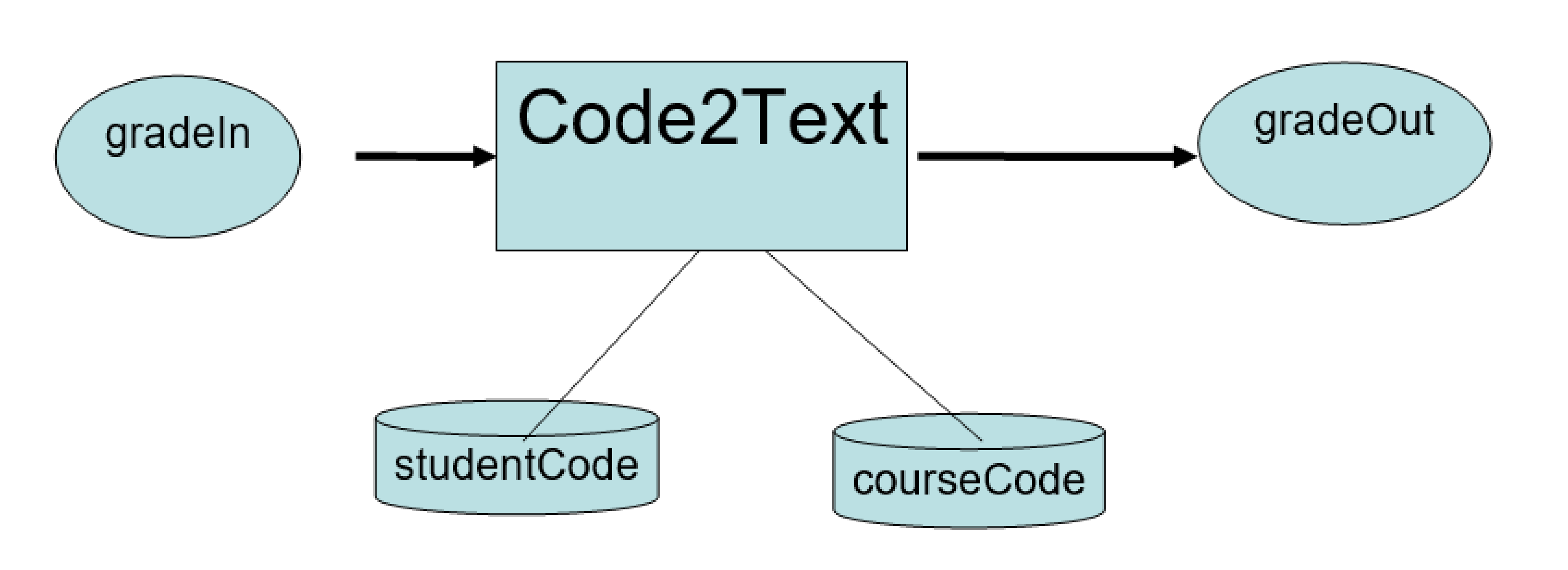
Framework

- Define class **Your_filter** to ***translate student code and course code to student name and course name*** using above framework, and to display the number of total lines translated.

- Assume that the command is executed as:

```
translate  gradeIn.txt   gradeOut.txt   studentCode.txt   courseCode.txt
```

Framework



Framework

- **argv[1]: gradeIn File**

Format : studentCode courseCode Grade

an example line: 10185402101 213005 98

- **argv[2]: gradeOut File**

Format : studentName courseName Grade

an example line: 张三 面向对象程序设计 98

Framework

- **argv[3]: studentCode File**

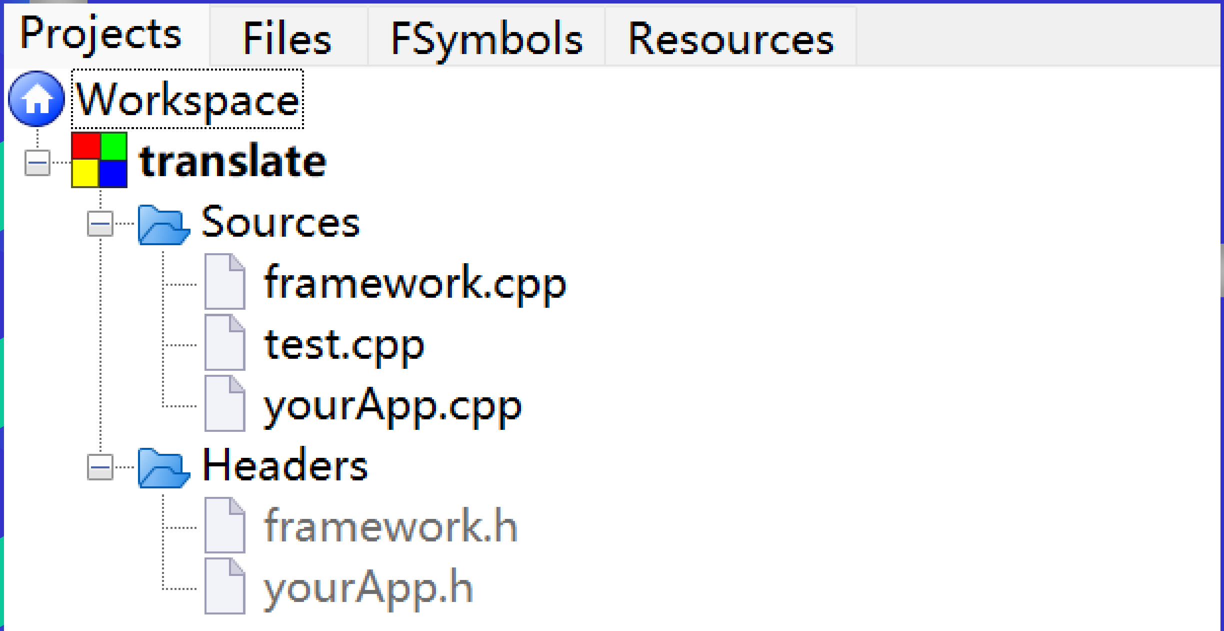
Format : studentCode studentName

an example line: 10185402101 张三

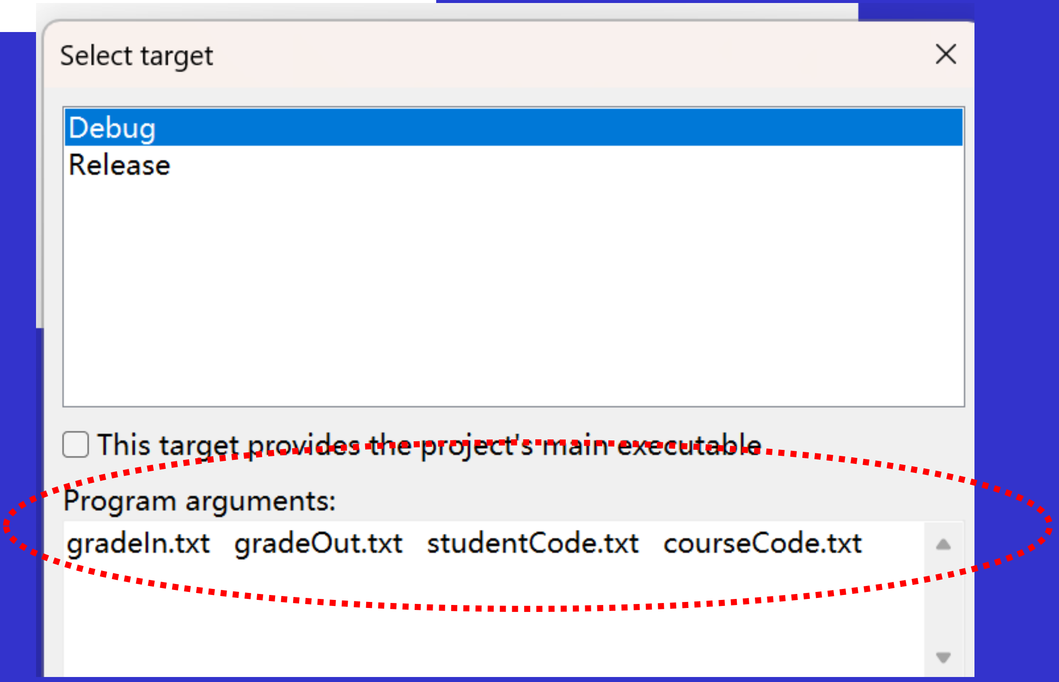
- **argv[4]: courseCode File**

Format : courseCode courseName

an example line: 213005 面向对象程序设计



- Workspace
  - translate
    - Sources
      - framework.cpp
      - test.cpp
      - yourApp.cpp
    - Headers
      - framework.h
      - yourApp.h



Debug
Release

□ This target provides the project's main executable

Program arguments:

```
gradeIn.txt  gradeOut.txt  studentCode.txt  courseCode.txt
```