

CS201 RECITATION 1

Introduction to C++

Outline

Part 1 : Writing and debugging code with CodeBlocks

Part 2 : Porting, compiling and testing in Dijkstra

Part 3 : Using and understanding header files

Part 1: Writing and debugging code with CodeBlocks

- Consider the following class:

```
#include <string>
using std::string;
class GradeBook {
public:
    GradeBook (string name) {
        setCourseName ( name );
    }
    void setCourseName (string name) {
        if(name.length() <= 25)
            courseName = name;
        else
            courseName = name.substr (0,25);
    }
    string getCourseName() {
        return courseName;
    }
private:
    string courseName;
};
```

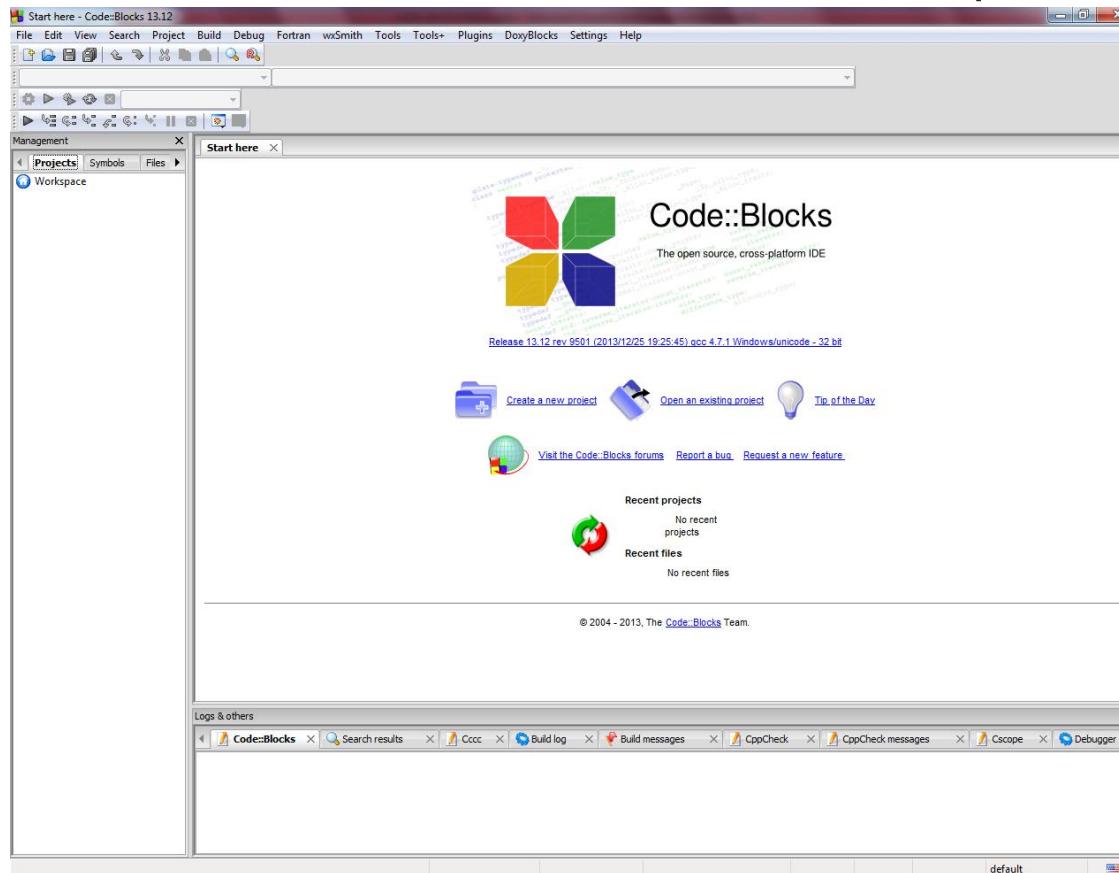
Let's modify the **GradeBook** class such that

- it keeps the **midterm**, **final**, **homework**, and **quiz** grades of a particular student as its data members
- it calculates a letter grade of the student using the **computeFinalGrade** member function that
 - takes four input grades from the user
 - computes the average grade acc. to the following weights
 - **midterm (30%)**, **final (35%)**, **homework (15%)**, **quiz (20%)**
 - assigns a letter grade according to the table

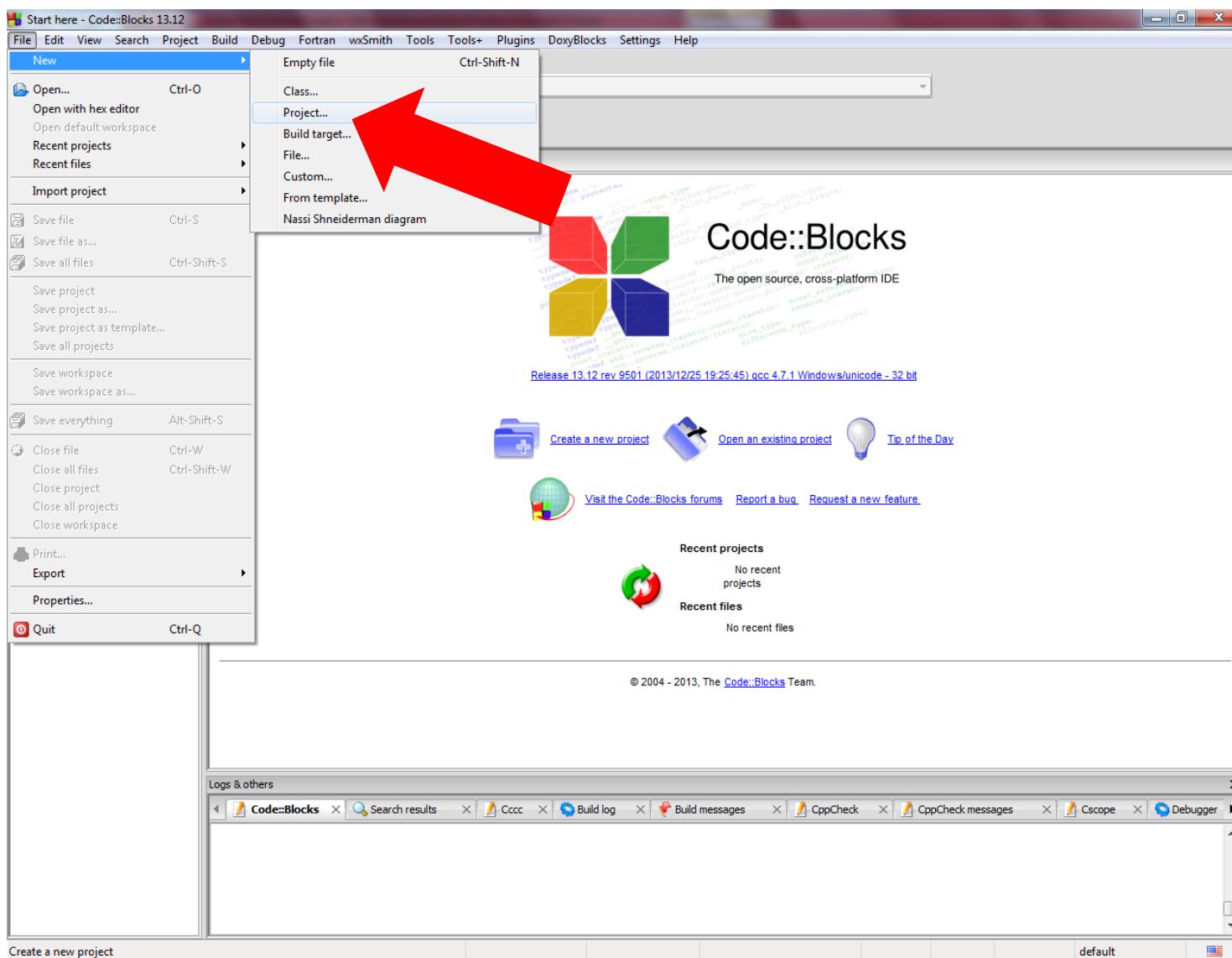
90 ≤ Grade	A
80 ≤ Grade ≤ 89	B
70 ≤ Grade ≤ 79	C
60 ≤ Grade ≤ 69	D
Grade < 60	F
Otherwise	U (unkown)

Let's do it using CodeBlocks

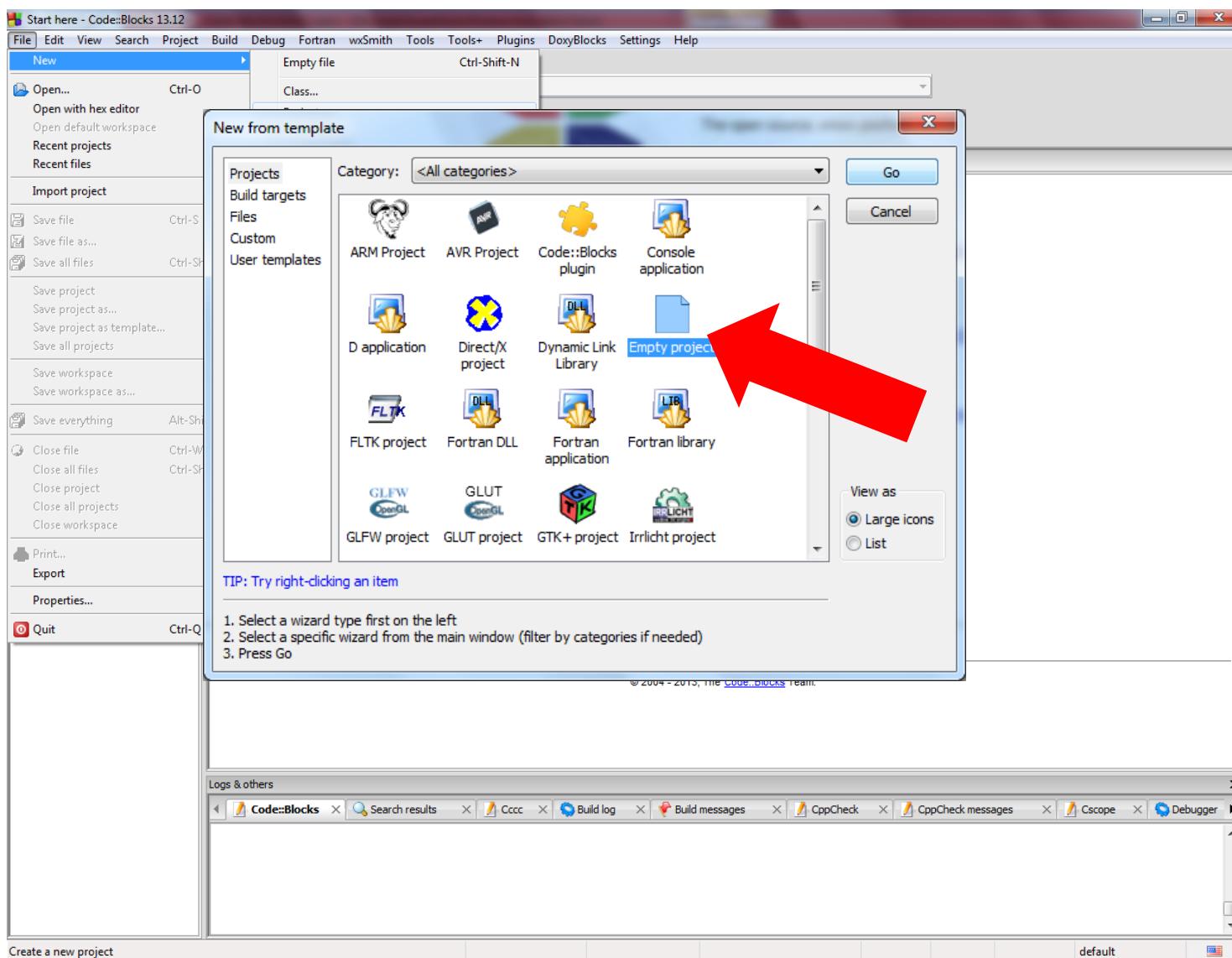
- CodeBlocks is an integrated development environment.
- <http://www.codeblocks.org/downloads/26>
- Make sure you download the IDE with its MinGW compiler.



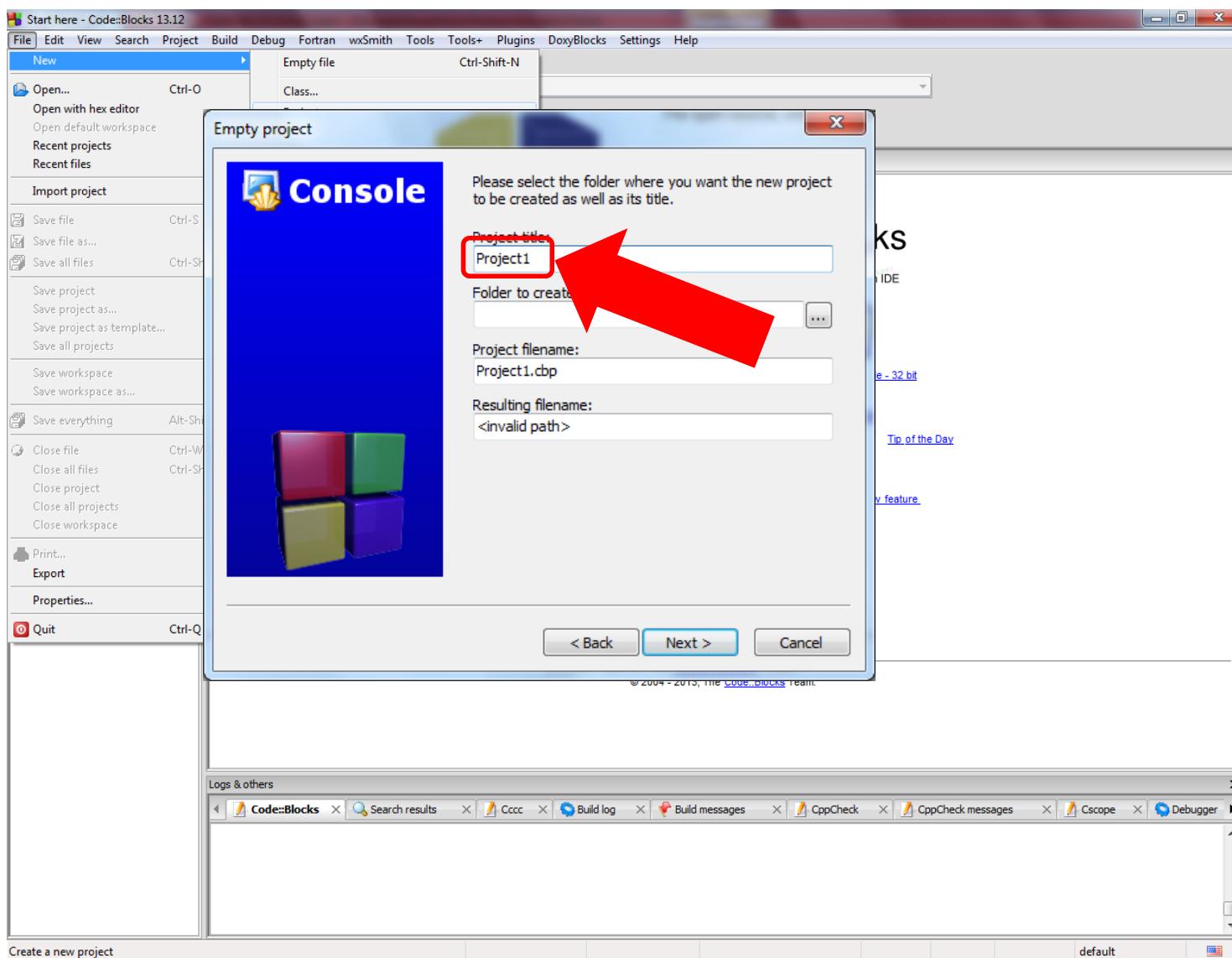
New C++ Project



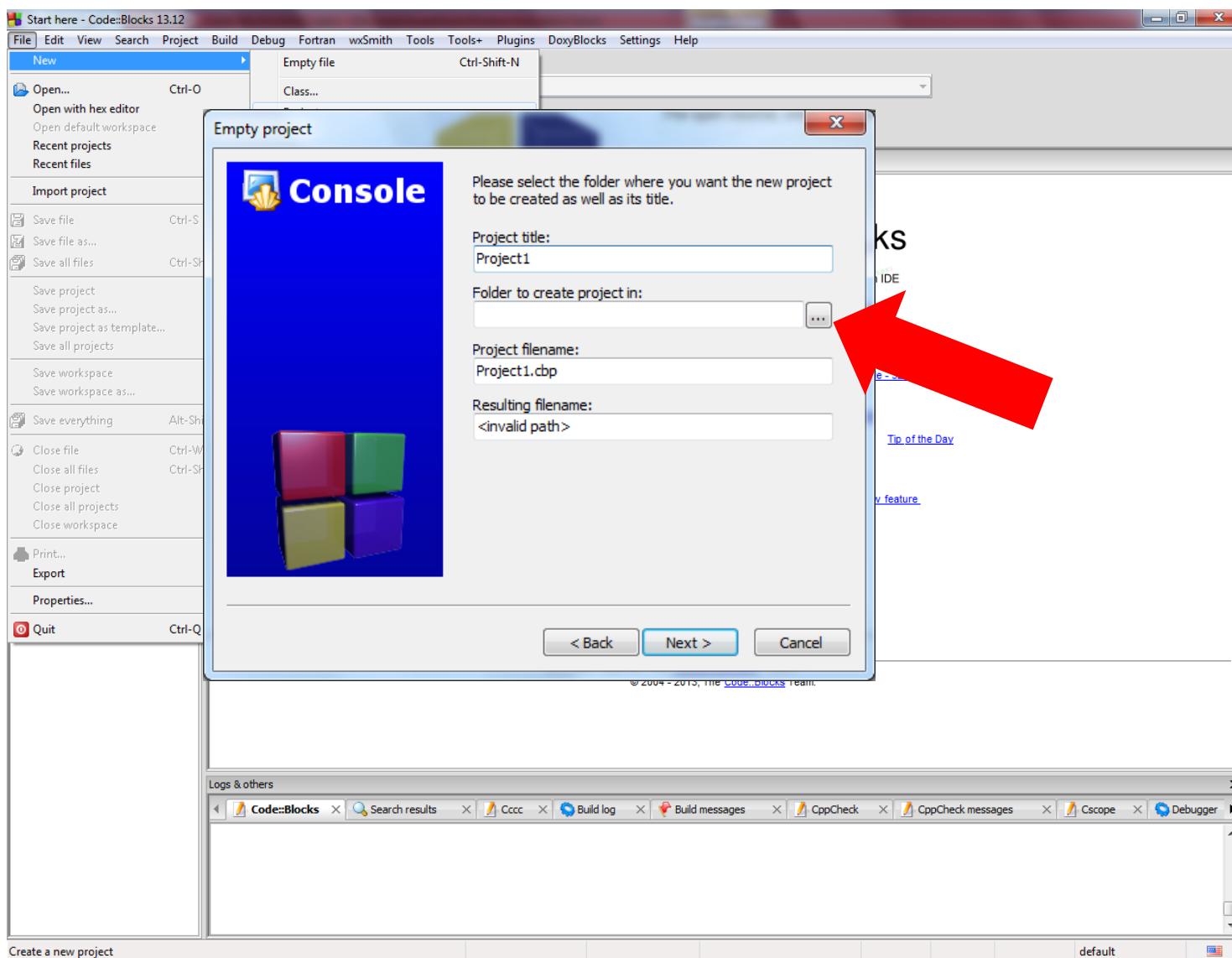
New C++ Project



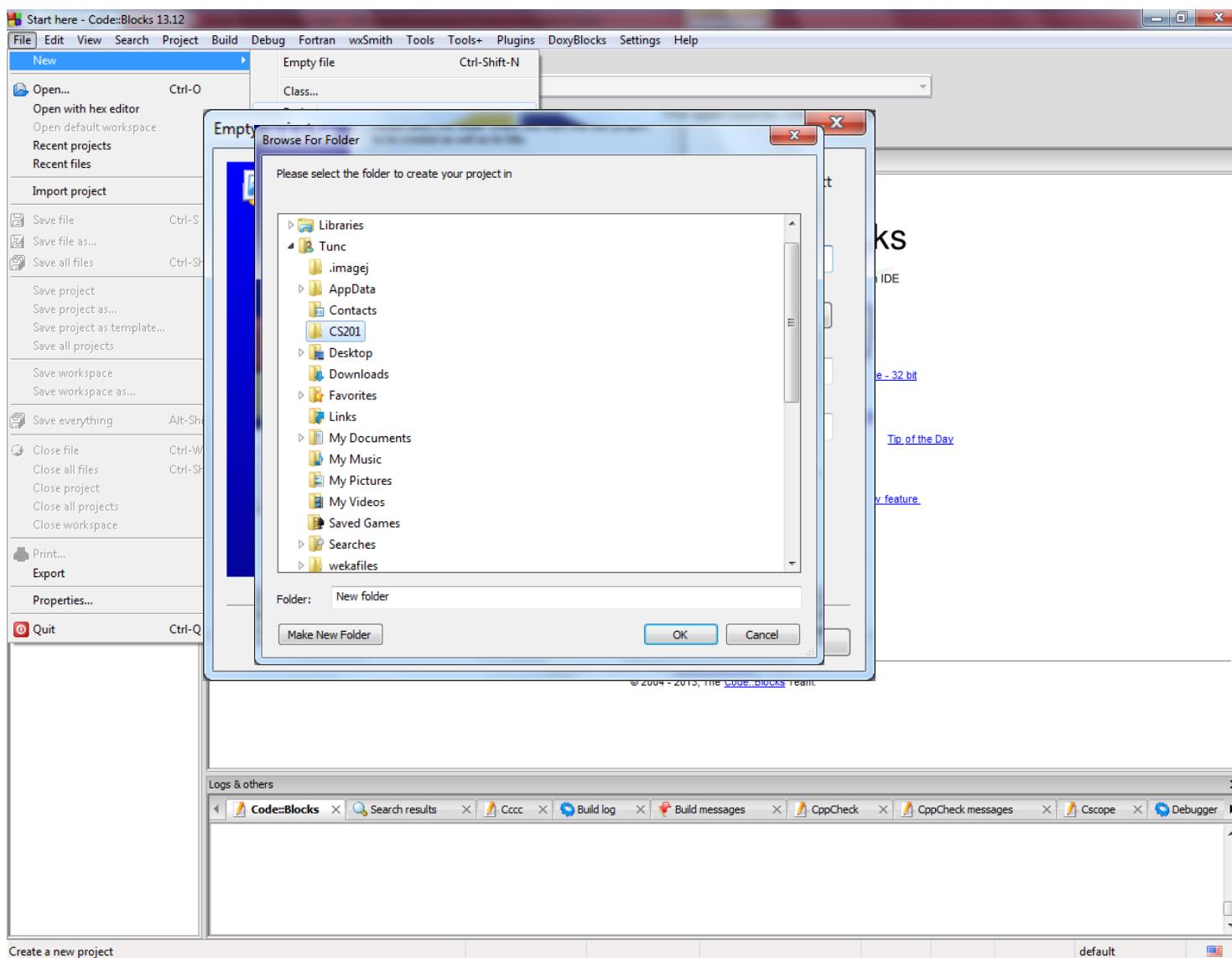
New C++ Project



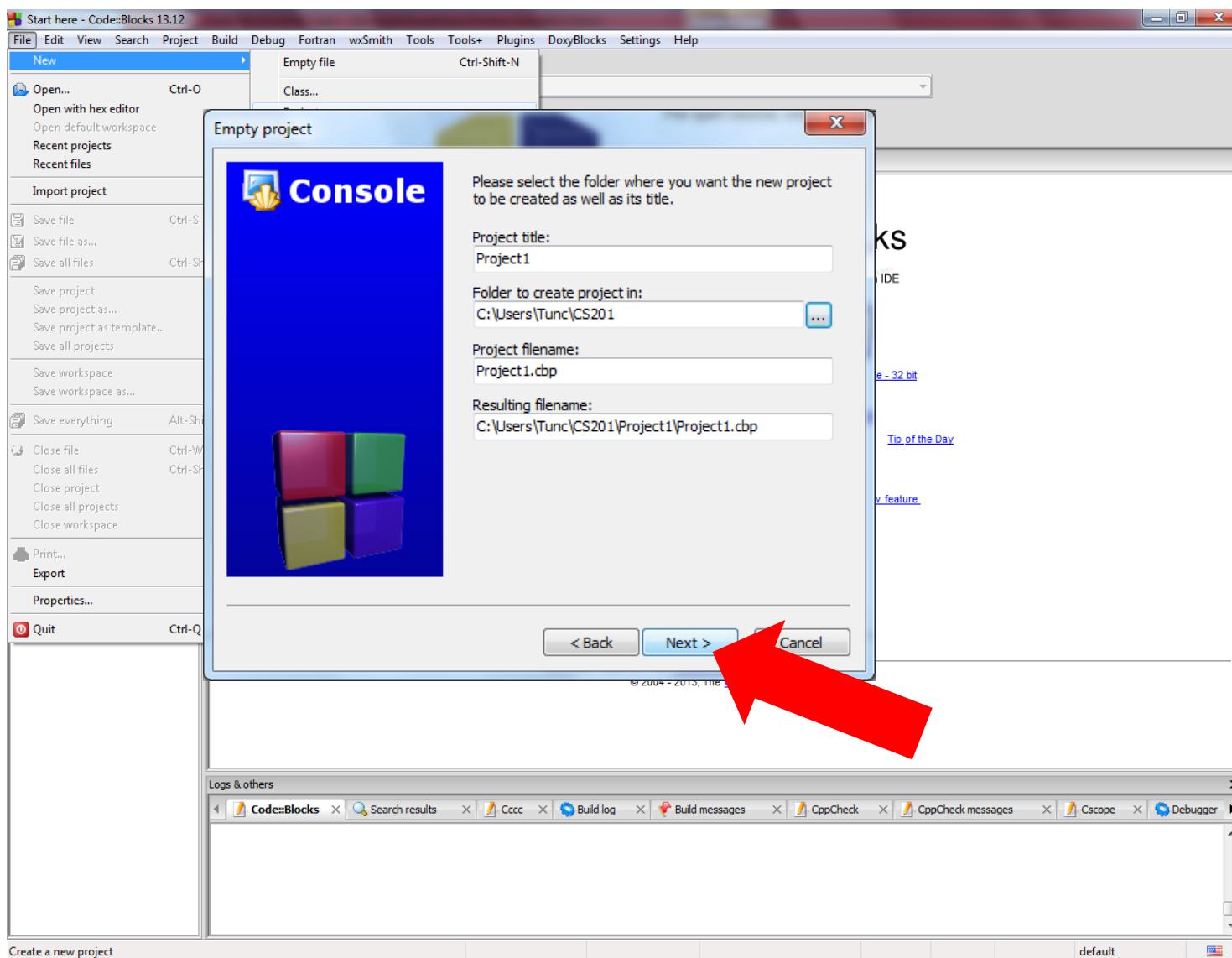
New C++ Project



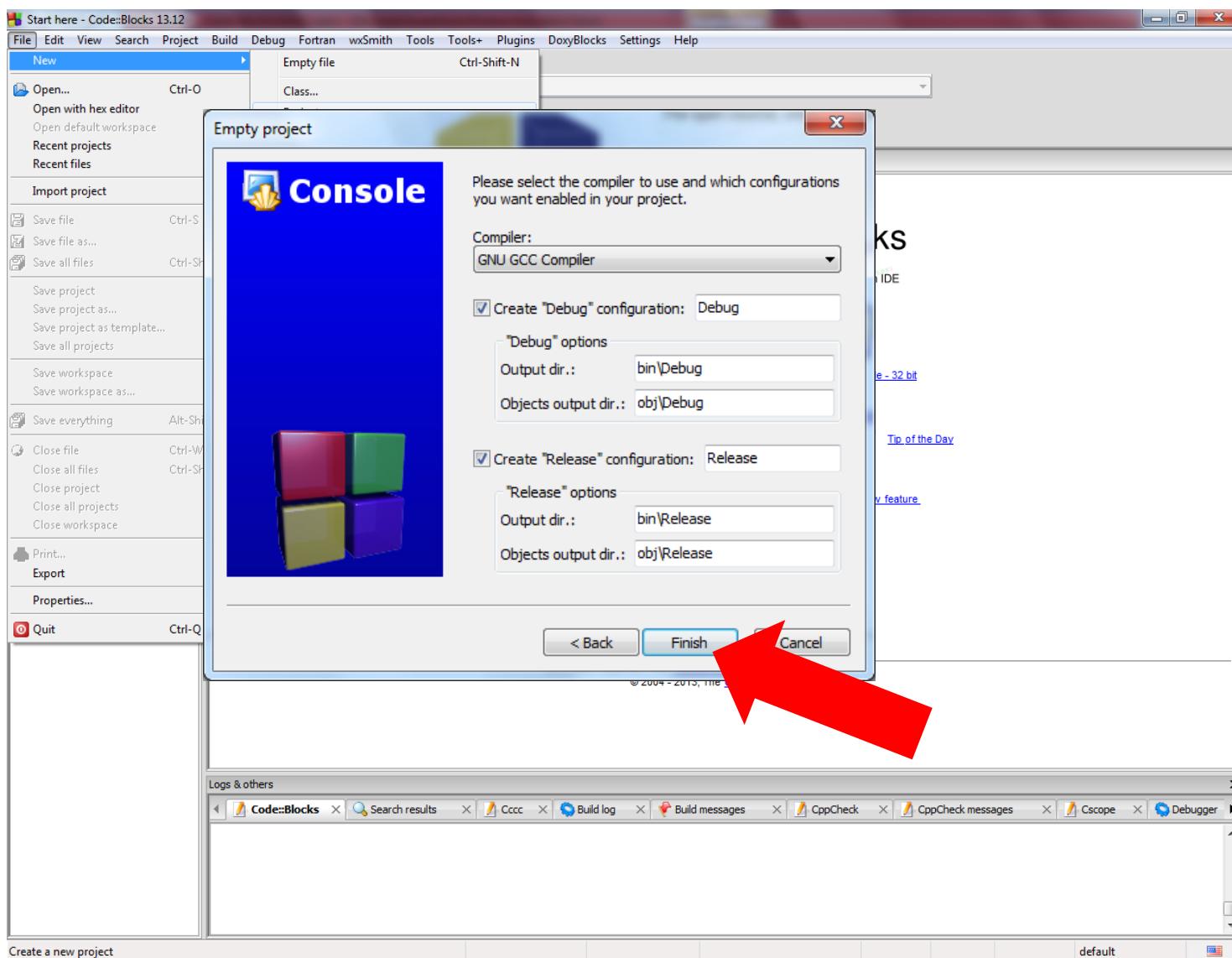
New C++ Project



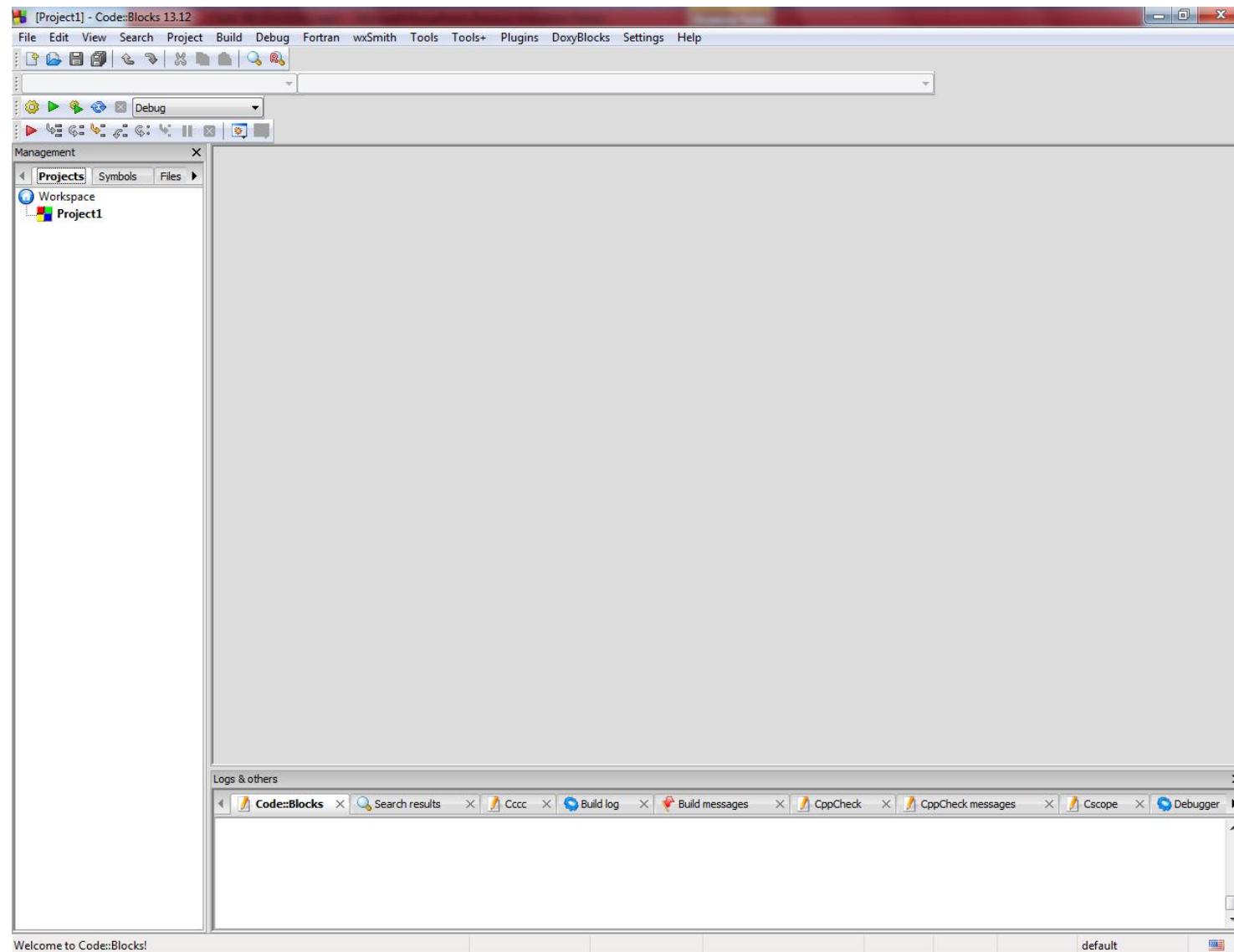
New C++ Project



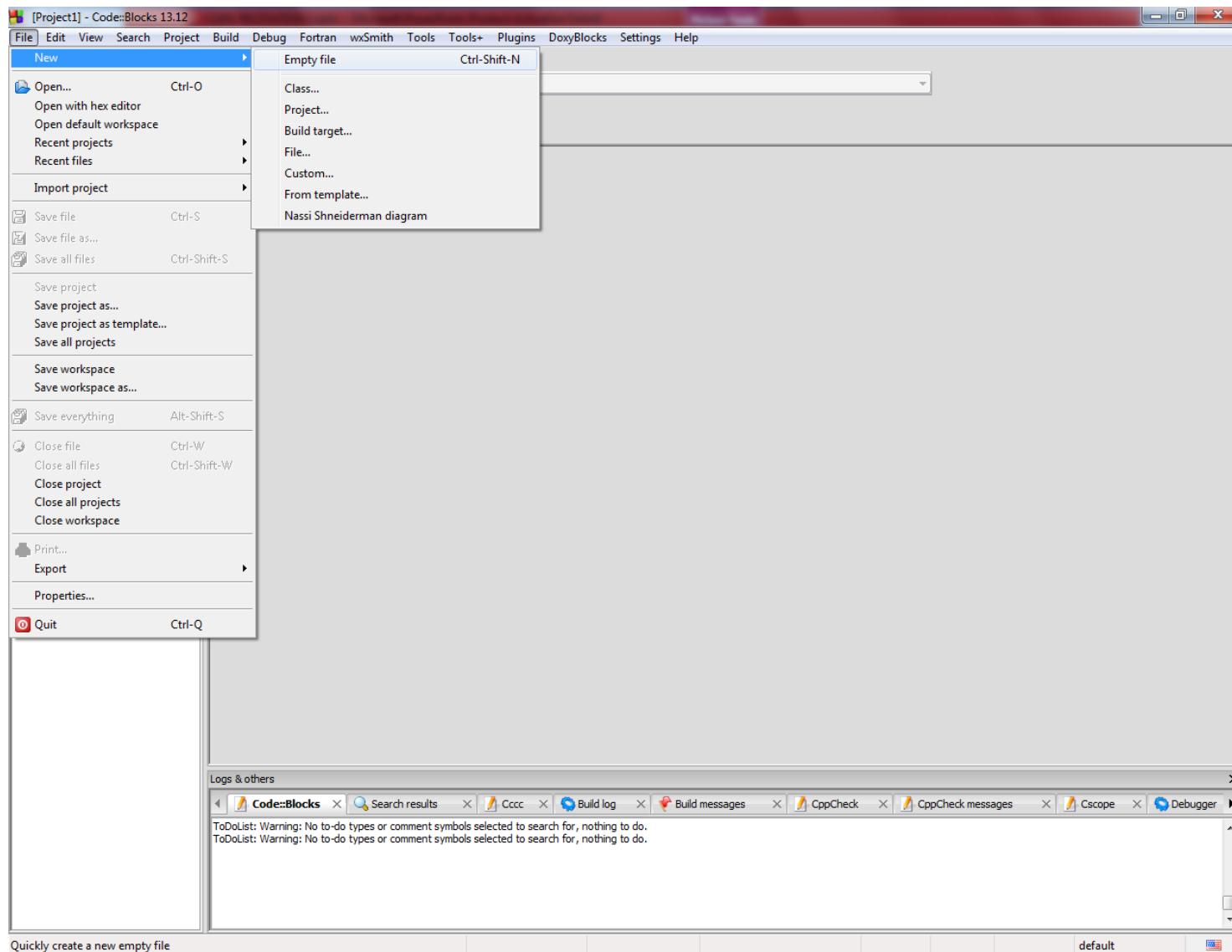
New C++ Project



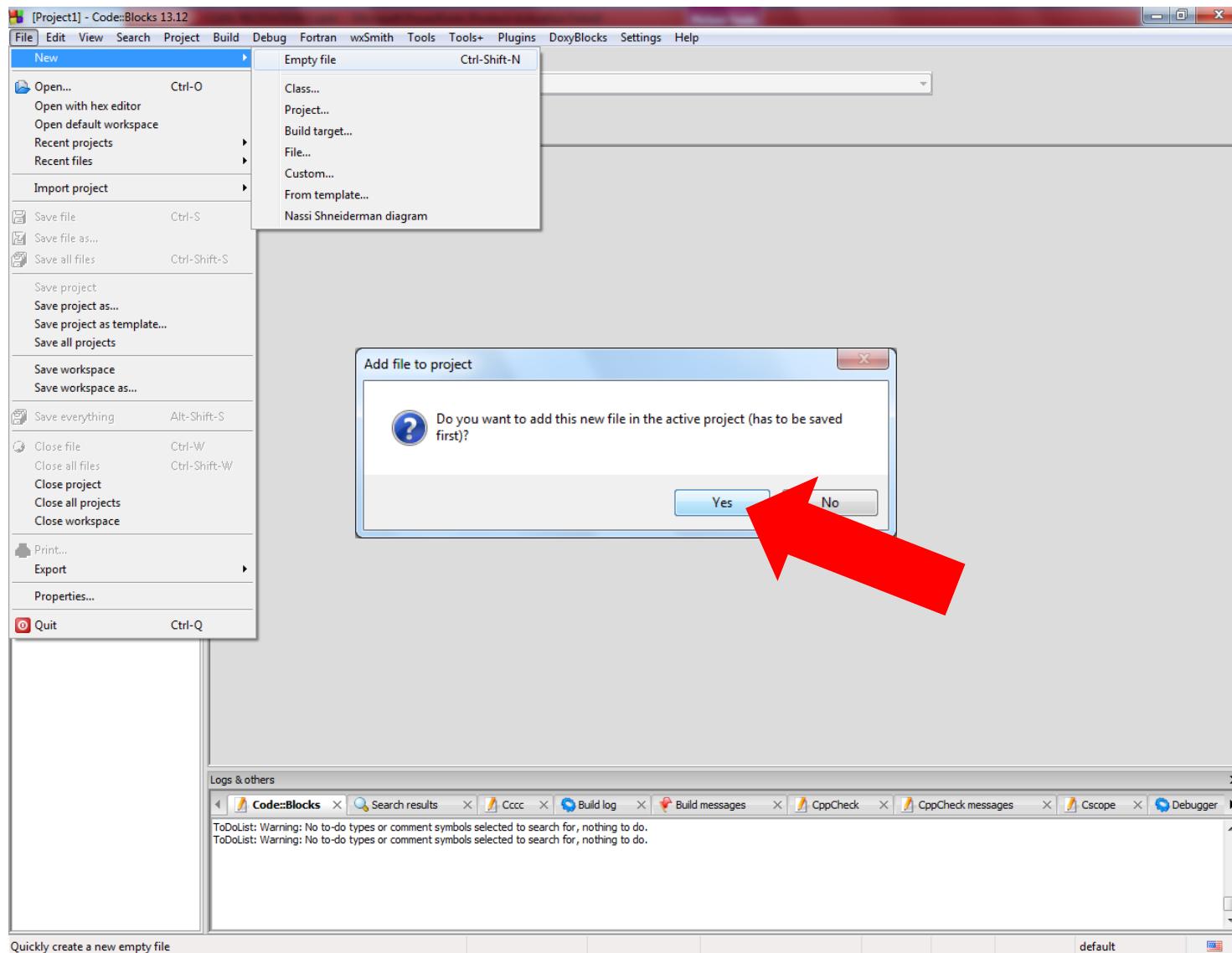
New Source File



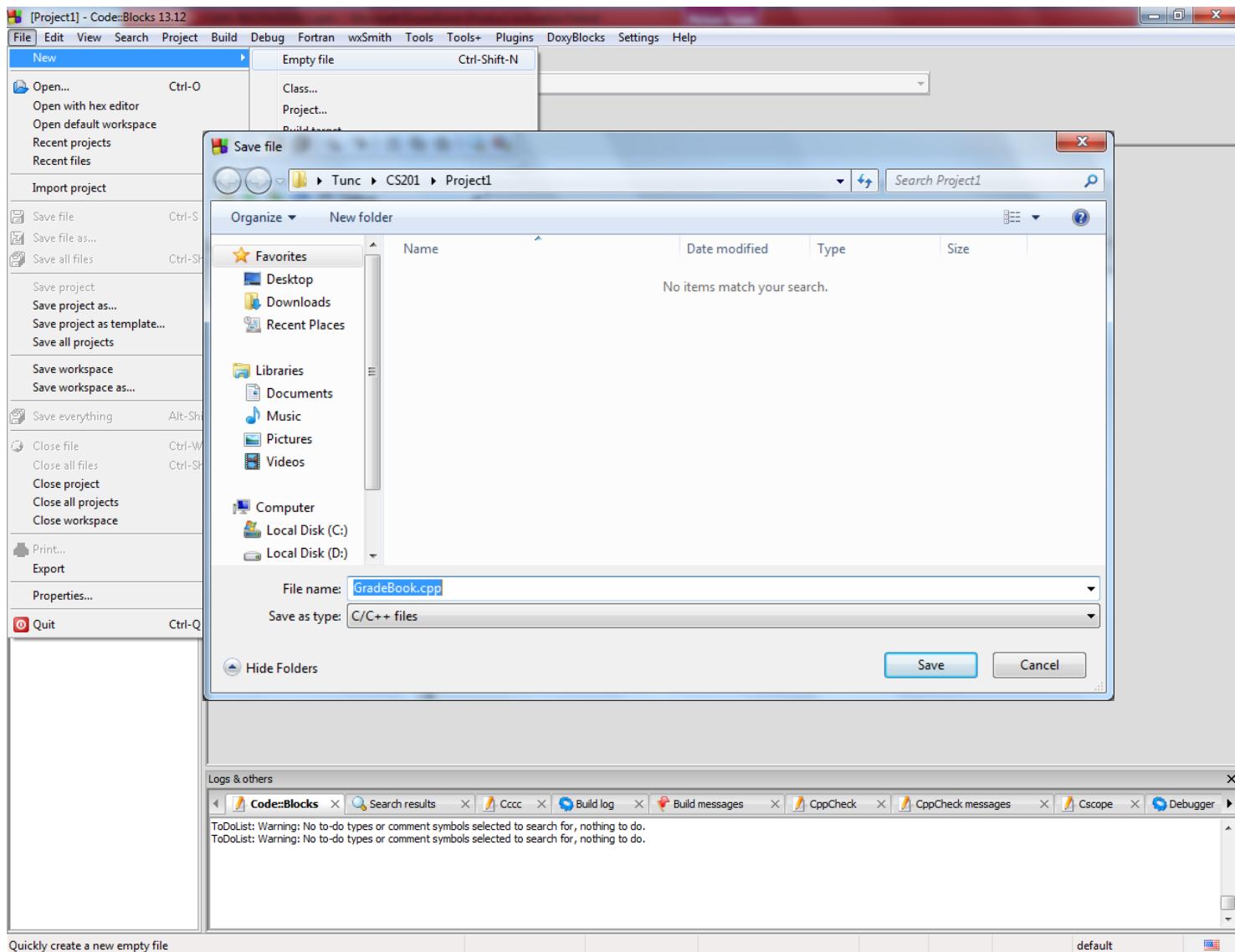
New Source File



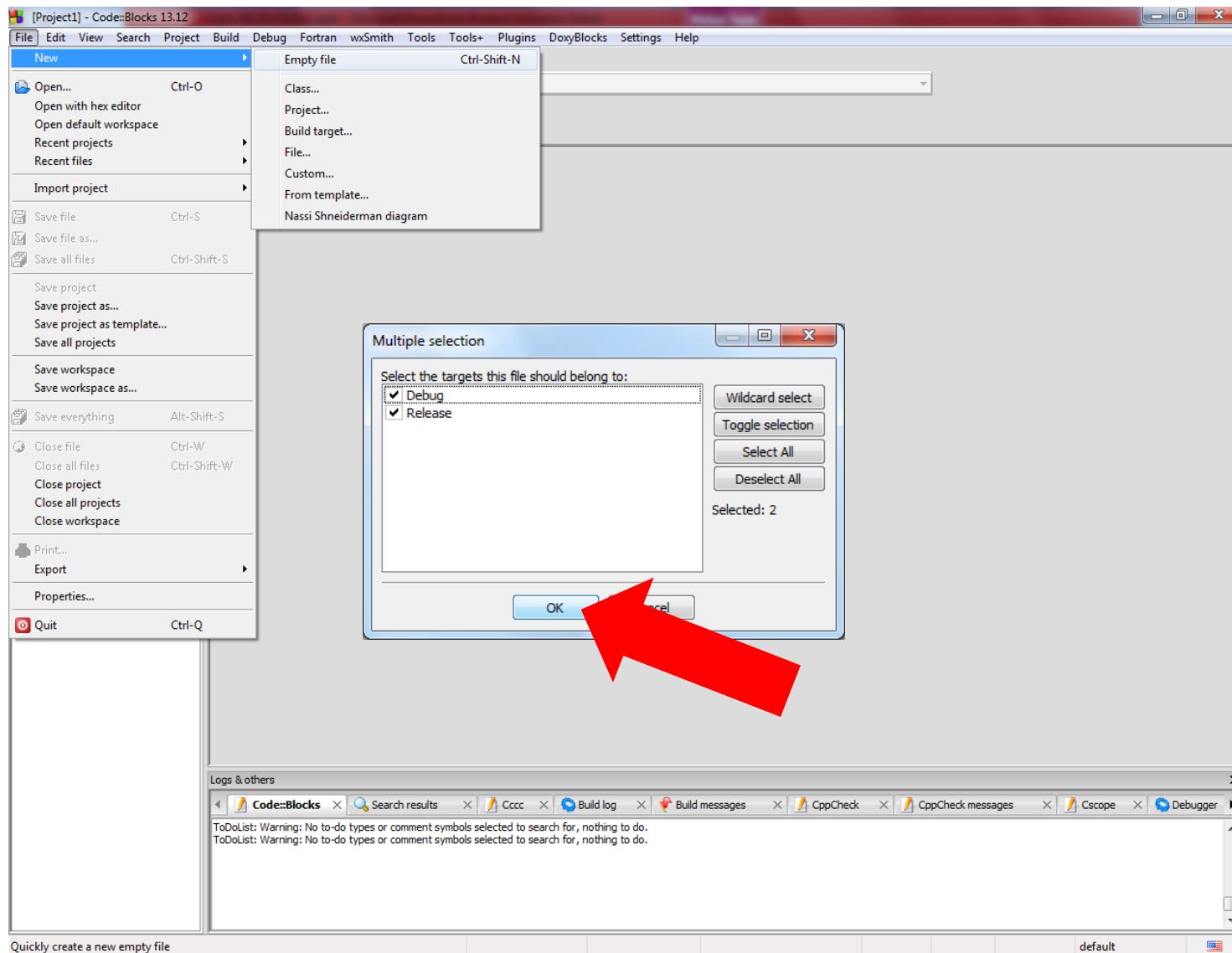
New Source File



New Source File



New Source File



GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface. The title bar reads "GradeBook.cpp [Project1] - Code::Blocks 13.12". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar contains various icons for file operations like Open, Save, and Build. The left sidebar has a "Management" section with "Projects", "Symbols", and "Files" tabs, showing "Project1" under "Workspace". The main code editor window is titled "GradeBook.cpp" and contains the following code:

```
1
2
```

The bottom status bar shows the path "C:\Users\Tunc\CS201\Project1\GradeBook.cpp", the operating system "Windows (CR+LF)", the default settings, the current line and column "Line 1, Column 1", and the insert mode. The logs window at the bottom displays build messages:

```
ToDoList: Warning: No to-do types or comment symbols selected to search for, nothing to do.
ToDoList: Warning: No to-do types or comment symbols selected to search for, nothing to do.
C:\Users\Tunc\CS201\Project1\GradeBook.cpp
NativeParser::CreateParser(): Finish creating a new parser for project "NONE"
NativeParser::DeleteParser(): Deleting parser for project "NONE"!
NativeParser::CreateParser(): Finish creating a new parser for project 'Project1'
NativeParser::OnParserEnd(): Project 'Project1' parsing stage done!
```

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Project Explorer (Management):** Shows a workspace with a Project1 folder containing Sources and GradeBook.cpp.
- Code Editor:** The GradeBook.cpp file is open, displaying the following code:

```
#include <string>
using std::string;
class GradeBook {
public:
private:
};
```
- Logs & others:** A tabbed log window showing build logs and messages. The build log contains:

```
ToDoList: Warning: No to-do types or comment symbols selected to search for, nothing to do.
ToDoList: Warning: No to-do types or comment symbols selected to search for, nothing to do.
C:\Users\Tunc\CS201\Project1\GradeBook.cpp
NativeParser::CreateParser(): Finish creating a new parser for project "NONE"
NativeParser::DeleteParser(): Deleting parser for project "NONE"!
NativeParser::CreateParser(): Finish creating a new parser for project 'Project1'
NativeParser::OnParserEnd(): Project 'Project1' parsing stage done!
```
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp, Windows (CR+LF), default, Line 7, Column 1, Insert, Read/Write, default, USA flag.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a search bar.
- Management View (Left):** Projects, Symbols, Files. Shows a workspace named "Project1" containing a source file "GradeBook.cpp".
- Code Editor (Center):** Displays the following C++ code:

```
#include <string>
using std::string;

class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }

    void setCourseName (string name) {
        if (name.length () <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }

    string getCourseName () {
        return courseName;
    }

private:
    string courseName;
};


```
- Logs & others (Bottom):** Shows build logs for "Project1 - Debug".

```
----- Clean: Debug in Project1 (compiler: GNU GCC Compiler) -----
Cleaned "Project1 - Debug"
Done.
```
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp, Windows (CR+LF), default, Line 25, Column 1, Insert, Read/Write, default, USA flag.

GradeBook.cpp

GradeBook.cpp [Project1] - Code::Blocks 13.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Symbols Files

GradeBook.cpp

```
#include <string>
using std::string;

class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }

    void setCourseName (string name) {
        if (name.length () <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }

    string getCourseName () {
        return courseName;
    }

private:
    string courseName;
};
```

Logs & others

C:\Users\Tunc\CS201\Project1\GradeBook.cpp

Windows (CR+LF) default Line 25, Column 1 Insert Read/Write default

Build & Run



GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for GradeBook.cpp:

```
#include <string>
using std::string;
class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }
    void setCourseName (string name) {
        if (name.length () <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }
    string getCourseName () {
        return courseName;
    }
private:
    string courseName;
};
```

A red arrow points from the top-left towards the toolbar. The toolbar contains icons for file operations like Open, Save, and Build, as well as project management and build-related tools.

```
== Build: Debug in Project1 (compiler: GNU GCC Compiler) ==
undefined reference to 'WinMain@16'
== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ==
```

The bottom part of the IDE shows the build log and message tabs. A red box highlights the 'Build messages' tab, which contains the same error message as the main console output. A red arrow points from the bottom of the main console area towards this tab.

Logs & others

File Line Message

```
== Build: Debug in Project1 (compiler: GNU GCC Compiler) ==
c:\program fil... undefined reference to 'WinMain@16'
== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ==
```

C:\Users\Tunc\CS201\Project1\GradeBook.cpp Windows (CR+LF) default Line 25, Column 1 Insert Read/Write default

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Management View:** Shows the Projects, Symbols, and Files tabs. Under Projects, there is a workspace named "Project1" containing a source file "GradeBook.cpp".
- Code Editor:** The "GradeBook.cpp" file is open. The code defines a class GradeBook with a constructor, a setCourseName method, a getCourseName method, and a main function. A red box highlights the entire main() function block.

```
#include <string>
using std::string;

class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }

    void setCourseName (string name) {
        if (name.length () <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }

    string getCourseName () {
        return courseName;
    }

private:
    string courseName;
};

int main () {
    GradeBook gb ("CS 201");
    cout << "Course created with course name " << gb.getCourseName () << endl;

    return 0;
}
```
- Logs & others:** Shows build logs and messages. The build log output is:

```
!!!  
==== Build: Debug in Project1 (compiler: GNU GCC Compiler) ====  
c:\program fil...  
==== undefined reference to 'WinMain@16'  
==== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp, Windows (CR+LF), default, Line 30, Column 14, Insert, Read/Write, default, USA flag.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins Doxygen Settings Help
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Toolbox:** Standard icons for project management, build, and debugging.
- Management View:** Shows the Projects, Symbols, and Files tabs. The Projects tab displays "Workspace" and "Project1" which contains "Sources" and "GradeBook.cpp".
- Code Editor:** The "GradeBook.cpp" file is open. The code defines a class GradeBook with a constructor, a setCourseName method, a getCourseName method, and a main function. A red box highlights the line "cout << "Course created with course name " << gb.getCourseName() << endl;" in the main function.
- Output Log:** A callout box points to the log window showing the following errors:

```
In function 'int main()':  
error: 'cout' was not declared in this scope  
error: 'endl' was not declared in this scope
```
- Build Log:** The bottom log window shows the build process and errors:

```
C:\Users\Tunc... In function 'int main()':  
C:\Users\Tunc... 28 error: 'cout' was not declared in this scope  
C:\Users\Tunc... 28 error: 'endl' was not declared in this scope  
==== Build failed: 2 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the file `GradeBook.cpp` open. The code editor displays the following C++ code:

```
#include <string>
#include <iostream>

using std::string;
using std::cout;
using std::endl;

class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }

    void setCourseName (string name) {
        if (name.length () <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }

    string getCourseName () {
        return courseName;
    }

private:
    string courseName;
};

int main()
{
    GradeBook gb ("CS 201");
    cout << "Course created with course name " << gb.getCourseName() << endl;

    return 0;
}
```

A red box highlights the first four lines of the code, which are the standard library includes and using declarations. A large black arrow points from this highlighted area to a callout box containing the same four lines of code, emphasizing their importance.

The bottom of the screen shows the terminal output of the build process:

```
mingw32-g++.exe -Wall -g -c C:\Users\Tunc\CS201\Project1\GradeBook.cpp -o obj\Debug\GradeBook.o
mingw32-g++.exe -o bin\Debug\Project1.exe obj\Debug\GradeBook.o
Output file is bin\Debug\Project1.exe with size 959.40 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

The status bar at the bottom indicates the file path `C:\Users\Tunc\CS201\Project1\GradeBook.cpp`, the operating system `Windows (CR+LF)`, the default encoding `default`, the current line and column `Line 35, Column 1`, and the current mode `Insert`.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Project Explorer:** Shows a workspace with a single project named "Project1" containing a source file "GradeBook.cpp".
- Code Editor:** Displays the C++ code for "GradeBook.cpp". A red rounded rectangle highlights the `computeFinalGrade()` function.

```
4  using std::string;
5  using std::cout;
6  using std::endl;
7
8  class GradeBook {
9  public:
10    GradeBook (string name) {
11      setCourseName (name);
12    }
13
14    void setCourseName (string name) {
15      if (name.length () <= 25)
16        courseName = name;
17      else
18        courseName = name.substr (0, 25);
19    }
20
21    string getCourseName () {
22      return courseName;
23    }
24
25    char computeFinalGrade () {
26      cout << "Enter the midterm grade : ";
27      cin >> midtermGrade;
28      cout << "Enter the final grade : ";
29      cin >> finalGrade;
30      cout << "Enter the quiz grade : ";
31      cin >> quizGrade;
32      cout << "Enter the homework grade : ";
33      cin >> hwGrade;
34
35      cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;
36      return 'U'; //Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
37    }
38
39  private:
40    string courseName;
41    double midtermGrade, finalGrade, quizGrade, hwGrade;
```

- Logs & others:** Shows the build log output for mingw32-g++.exe and the output file information.
- Status Bar:** Open an existing file, Windows (CR+LF), default, Line 34, Column 1, Insert, Read/Write, default, USA flag.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Management window showing Projects, Symbols, and Files. A project named "Project1" is selected, containing a source file "GradeBook.cpp".
- Code Editor:** The "GradeBook.cpp" file is open. The code defines a class GradeBook with methods for setting and getting course names, and calculating final grades. A red cursor is at line 27, column 11, where the error occurred.
- Output Window:** Shows the error message: "error: 'cin' was not declared in this scope". An arrow points from this message to the "Build log" tab in the bottom navigation bar.
- Logs & others:** A tabbed window showing various logs and messages. The "Build log" tab is active, displaying the error message.
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp, Windows (CR+LF), default, Line 27, Column 1, Insert, Read/Write, default, USA flag.

```
4 using std::string;
5 using std::cout;
6 using std::endl;
7
8 class GradeBook {
9 public:
10    GradeBook (string name) {
11        setCourseName (name);
12    }
13
14    void setCourseName (string name) {
15        if (name.length() <= 25)
16            courseName = name;
17        else
18            courseName = name.substr (0, 25);
19    }
20
21    string getCourseName () {
22        return courseName;
23    }
24
25    char computeFinalGrade () {
26        cout << "Enter the midterm grade : ";
27        cin >> midtermGrade;
28        cout << "Enter the final grade : ";
29        cin >> finalGrade;
30        cout << "Enter the quiz grade : ";
31        cin >> quizGrade;
32        cout << "Enter the homework grade : ";
33        cin >> hwGrade;
34
35        cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl
36        return 'U'; //Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
37    }
38
39 private:
```

cin >> midtermGrade;

error: 'cin' was not declared in this scope

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface. The main window displays the `GradeBook.cpp` file with the following code:

```
using std::string;
using std::cout;
using std::endl;
using std::cin;

class GradeBook {
public:
    GradeBook (string name) {
        setCourseName (name);
    }

    void setCourseName (string name) {
        if (name.length() <= 25)
            courseName = name;
        else
            courseName = name.substr (0, 25);
    }

    string getCourseName () {
        return courseName;
    }

    char computeFinalGrade () {
        cout << "Enter the midterm grade : ";
        cin >> midtermGrade;
        cout << "Enter the final grade : ";
        cin >> finalGrade;
        cout << "Enter the quiz grade : ";
        cin >> quizGrade;
        cout << "Enter the homework grade : ";
        cin >> hwGrade;
        cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl
        // Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
    }
}
```

A callout box highlights the line `using std::cin;` with the text **Alternative: using namespace std;**. A red box highlights the line `using std::cin;` in the code editor.

Successfully Compiled!

0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

Build log tab shows: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface. The title bar reads "GradeBook.cpp [Project1] - Code::Blocks 13.12". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has icons for file operations like Open, Save, and Build. The left sidebar shows the project structure under "Management" with "Projects" selected, showing "Project1" which contains "Sources" and "GradeBook". The main window displays the terminal output of the program "Project1.exe". The output text is:
Course created with course name CS 201
Process returned 0 <0x0> execution time : 0.016 s
Press any key to continue.

The code editor at the bottom shows the C++ code for the "GradeBook" class:

```
29     cout << "Enter the final grade : ";
30     cin >> finalGrade;
31     cout << "Enter the quiz grade : ";
32     cin >> quizGrade;
33     cout << "Enter the homework grade : ";
34     cin >> hwGrade;
35
36     cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl
37     return 'U'; //Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
38 }
39
40 private:
41     string courseName;
```

The status bar at the bottom indicates "Windows (CR+LF)" and "default" for the file type, and "Line 7, Column 16" for the current cursor position. The bottom tabs show various logs and build results, all of which are currently empty.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the file `GradeBook.cpp` open. The code defines a class `GradeBook` with methods for setting and getting course names, calculating final grades, and printing grade details. A tooltip for the `computeFinalGrade()` method is displayed, showing its signature and parameters.

```
14 void setCourseName (string name) {
15     if (name.length() <= 25)
16         courseName = name;
17     else
18         courseName = name.substr (0, 25);
19 }
20
21 string getCourseName() {
22     return courseName;
23 }
24
25
26 char computeFinalGrade() {
27     cout << "Enter the midterm grade : ";
28     cin >> midtermGrade;
29     cout << "Enter the final grade : ";
30     cin >> finalGrade;
31     cout << "Enter the quiz grade : ";
32     cin >> quizGrade;
33     cout << "Enter the homework grade : ";
34     cin >> hwGrade;
35
36     cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;
37     return 'U'; //Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
38 }
39
40 private:
41     string courseName;
42     double midtermGrade, finalGrade, quizGrade, hwGrade;
43 };
44
45 int main() {
46     GradeBook gb ("CS 201");
47     cout << "Course created with course name " << gb.getCourseName() << endl;
48
49
50
51 }
```

gb.
•(1) computeFinalGrade(): char
• courseName: string
• finalGrade: double
•(2) getCourseName(): string
• GradeBook():
• hwGrade: double
• midTermGrade: double
• quizGrade: double
•(3) setCourseName(): void

gb.
1 (1) computeFinalGrade (): char

C:\Users\Tunc\CS201\Project1\GradeBook.cpp

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface. The main window displays the output of a program named Project1.exe. The output text is:

```
Course created with course name CS 201
Enter the midterm grade : 90
Enter the final grade : 80
Enter the quiz grade : 85
Enter the homework grade : 65
The grades are 90, 80, 85, 65
Process returned 0 <0x0> execution time : 22.197 s
Press any key to continue.
```

The code editor window below shows the implementation of the GradeBook class and its main function. A cursor is visible at the end of the line "cout << hwGrade << endl;" in the main function.

```
39
40     private:
41         string courseName;
42         double midtermGrade, finalGrade, quizGrade, hwGrade;
43     };
44
45     int main() {
46         GradeBook gb ("CS 201");
47         cout << "Course created with course name " << gb.getCourseName() << endl;
48
49         gb.computeFinalGrade();
50         return 0;
51     }

```

The status bar at the bottom indicates the file path C:\Users\Tunc\CS201\Project1\GradeBook.cpp, the operating system Windows (CR+LF), default settings, Line 44, Column 1, Insert mode, Read/Write mode, and default settings again.

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the file GradeBook.cpp open. The code implements a class GradeBook with methods to set and get course name, and to compute final grade based on midterm, final, quiz, and homework grades.

```
void setCourseName (string name) {
    if (name.length() <= 25)
        courseName = name;
    else
        courseName = name.substr (0, 25);
}

string getCourseName () {
    return courseName;
}

char computeFinalGrade () {
    double overallGrade;
    cout << "Enter the midterm grade : ";
    cin >> midtermGrade;
    cout << "Enter the final grade : ";
    cin >> finalGrade;
    cout << "Enter the quiz grade : ";
    cin >> quizGrade;
    cout << "Enter the homework grade : ";
    cin >> hwGrade;

    cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;

    overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
    cout << "Overall grade : " << overallGrade << endl;
    return 'U'; //Since we didn't calculate the letter grade yet it returns 'U' (Unknown)
}
```

A red box highlights the line `double overallGrade;`. A black arrow points from this line to the word `overallGrade` in the line `cout << "Overall grade : " << overallGrade << endl;`. Below the code editor, a separate window shows the main function and its execution results.

```
overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
cout << "Overall grade : " << overallGrade << endl;
```

The bottom part of the IDE shows the main function definition and the build log.

```
int main()
{
    GradeBook gb ("CS 201");
    cout << "Course created with course name " << gb.getCourseName() << endl;
    !!!
```

The build log indicates the process terminated successfully with status 0.

```
Process terminated with status 0 (0 minute(s), 48 second(s))
```

File path: C:\Users\Tunc\CS201\Project1\GradeBook.cpp

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the file `GradeBook.cpp` open. The code implements a class `GradeBook` with a method `computeFinalGrade()` that calculates an overall grade based on four inputs: midterm, final, quiz, and homework grades. The output is then converted into letter grades using an if-else ladder.

```
GradeBook.cpp [Project1] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
GradeBook::: computeFinalGrade() : char
Management X
Projects Symbols Files
Project1
Sources
GradeBook.cpp
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
private:
    string courseName;
}

string getCourseName() {
    return courseName;
}

char computeFinalGrade() {
    double overallGrade;
    cout << "Enter the midterm grade : ";
    cin >> midtermGrade;
    cout << "Enter the final grade : ";
    cin >> finalGrade;
    cout << "Enter the quiz grade : ";
    cin >> quizGrade;
    cout << "Enter the homework grade : ";
    cin >> hwGrade;

    cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;

    overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
    cout << "Overall grade : " << overallGrade << endl;

    if (overallGrade > 100 || overallGrade < 0)
        return 'U';
    else if (overallGrade >= 90)
        return 'A';
    else if (overallGrade >= 80)
        return 'B';
    else if (overallGrade >= 70)
        return 'C';
    else if (overallGrade >= 60)
        return 'D';
    else
        return 'F';
}

Logs & others
Code::Blocks X Search results X Cccc X Build log X Build message
Process terminated with status 0 (0 minute(s), 48 second(s))
C:\Users\Tunc\CS201\Project1\GradeBook.cpp Windows (CR+LF) default Line
```

A red box highlights the if-else ladder logic for determining the letter grade. A black arrow points from this highlighted section to the identical code in the bottom right corner, which appears to be a copy or a reference version of the same logic.

```
if (overallGrade > 100 || overallGrade < 0)
    return 'U';
else if (overallGrade >= 90)
    return 'A';
else if (overallGrade >= 80)
    return 'B';
else if (overallGrade >= 70)
    return 'C';
else if (overallGrade >= 60)
    return 'D';
else
    return 'F';
```

GradeBook.cpp

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
- Toolbars:** Standard toolbar with icons for file operations.
- Project Explorer:** Shows a workspace with a Project1 folder containing Sources and GradeBook.
- Output Window:** Displays the program's execution output:

```
Course created with course name CS 201
Enter the midterm grade : 90
Enter the final grade : 80
Enter the quiz grade : 90
Enter the homework grade : 75
The grades are 90, 80, 90, 75
Overall grade : 84.25
Your letter grade : B

Process returned 0 (0x0)   execution time : 14.641 s
Press any key to continue.
```
- Code Editor:** Shows the C++ code for GradeBook.cpp:

```
56     private:
57         string courseName;
58         double midtermGrade, finalGrade, quizGrade, hwGrade;
59     };
60
61     int main() {
62         GradeBook gb ("CS 201");
63         cout << "Course created with course name " << gb.getCourseName() << endl;
64
65         cout << "Your letter grade : " << gb.computeFinalGrade() << endl;
66         return 0;
67     }
68 }
```
- Logs & others:** Shows build logs and messages, indicating the process terminated successfully.
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp | Windows (CR+LF) | WINDOWS-1254 | Line 37, Column 6 | Insert | Read/Write | default |

GradeBook.cpp (Debugging)

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Tool Buttons:** Includes icons for Run, Stop, Breakpoint, and others.
- Management View:** Shows the Project tree with 'Project1' selected, containing 'Sources' and 'GradeBook.cpp'.
- Code Editor:** Displays the 'GradeBook.cpp' source code. A red circle highlights line 67, which contains the assignment statement `letterGrade = gb.computeFinalGrade();`. A callout box labeled "Breakpoint" points to this line.
- Logs & others:** Shows the status "Debugger finished with status 0".
- Status Bar:** C:\Users\Tunc\CS201\Project1\GradeBook.cpp, Windows (CR+LF), WINDOWS-1254, Line 47, Column 24, Insert, Read/Write, default.

```
cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;
overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
cout << "Overall grade : " << overallGrade << endl;

if (overallGrade > 100 || overallGrade < 0)
    return 'U';
else if (overallGrade >= 90)
    return 'A';
else if (overallGrade >= 80)
    return 'B';
else if (overallGrade >= 70)
    return 'C';
else if (overallGrade >= 60)
    return 'D';
else
    return 'F';

private:
    string courseName;
    double midtermGrade, finalGrade, quizGrade, hwGrade;
};

int main()
{
    GradeBook gb ("CS 201");
    char letterGrade;
    string courseName;

    courseName = gb.getCourseName();
    letterGrade = gb.computeFinalGrade();

    cout << "Course created with course name " << courseName << endl;
    cout << "Your letter grade : " << letterGrade << endl;

    return 0;
}
```

GradeBook.cpp (Debugging)

The screenshot shows the Code::Blocks IDE interface with the project "GradeBook" open. The code editor displays GradeBook.cpp, which contains a main function that creates a GradeBook object and prints its course name and final grade. A red circular breakpoint marker is placed on line 67, where the computeFinalGrade() method is called. The code window has syntax highlighting for C++ and includes a vertical green margin line. The status bar at the bottom indicates the code is on line 47, column 24.

```
55
56     private:
57         string courseName;
58         double midtermGrade, finalGrade, quizGrade, hwGrade;
59     };
60
61 int main() {
62     GradeBook gb ("CS 201");
63     char letterGrade;
64     string courseName;
65
66     courseName = gb.getCourseName();
67     letterGrade = gb.computeFinalGrade();
68
69     cout << "Course created with course name " << courseName << endl;
70     cout << "Your letter grade : " << letterGrade << endl;
71
72     return 0;
73 }
```

Logs & others

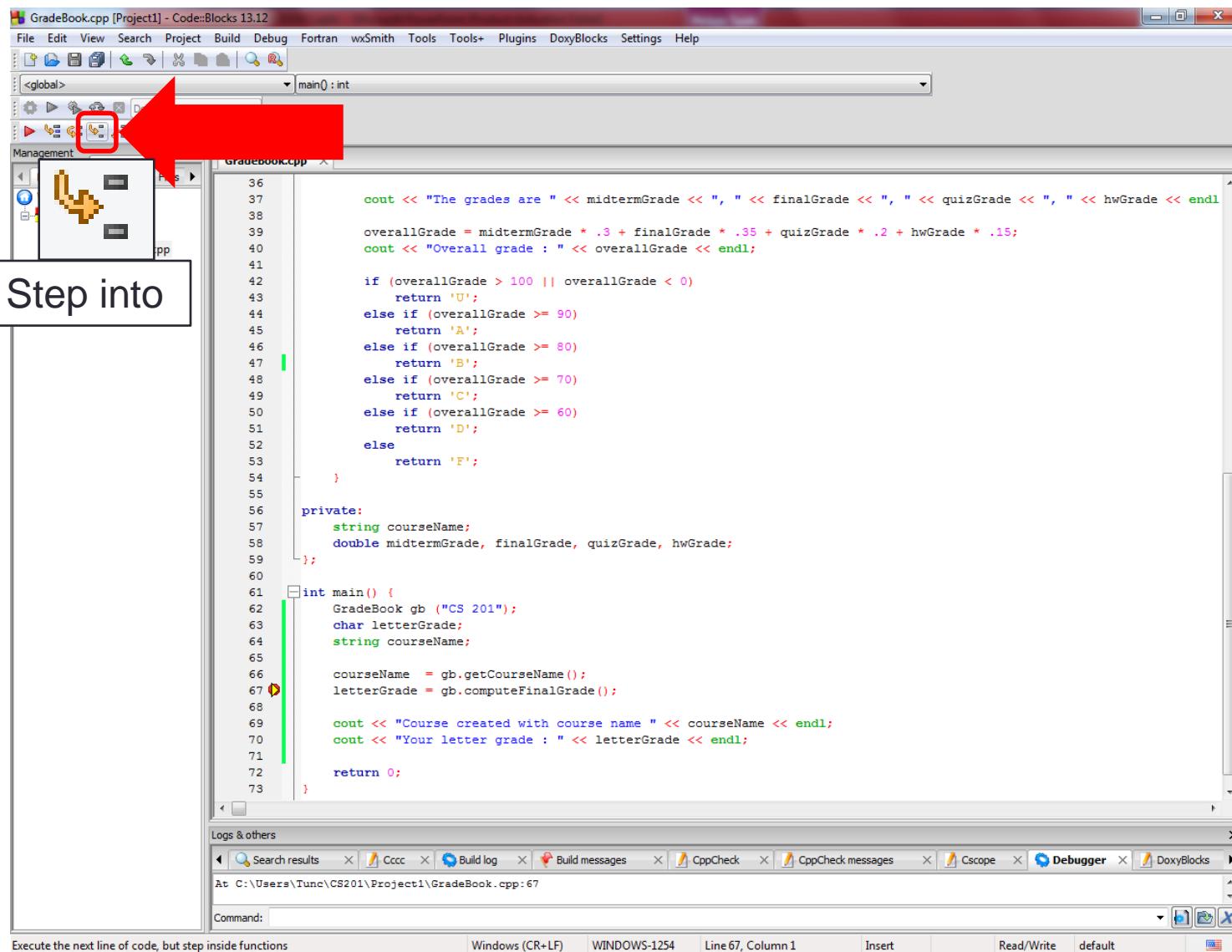
Search results | Ccc | Build log | Build messages | CppCheck | CppCheck messages | Cscope | Debugger | DoxyBlocks

Debugger finished with status 0

Command:

Windows (CR+LF) | WINDOWS-1254 | Line 47, Column 24 | Insert | Read/Write | default |

GradeBook.cpp (Debugging)



The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for New, Open, Save, Print, Find, Replace, and others.
- Left Panel:** Shows a project tree labeled "Management" with a file icon and a "Gradebook.cpp" entry.
- Code Editor:** Displays the C++ code for GradeBook.cpp. A red arrow points to the "Step into" button in the toolbar, which is highlighted with a white border and a callout box containing the text "Step into".
- Code Content:**

```
cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;
overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
cout << "Overall grade : " << overallGrade << endl;

if (overallGrade > 100 || overallGrade < 0)
    return 'U';
else if (overallGrade >= 90)
    return 'A';
else if (overallGrade >= 80)
    return 'B';
else if (overallGrade >= 70)
    return 'C';
else if (overallGrade >= 60)
    return 'D';
else
    return 'F';

private:
    string courseName;
    double midtermGrade, finalGrade, quizGrade, hwGrade;
};

int main()
{
    GradeBook gb ("CS 201");
    char letterGrade;
    string courseName;

    courseName = gb.getCourseName();
    letterGrade = gb.computeFinalGrade();

    cout << "Course created with course name " << courseName << endl;
    cout << "Your letter grade : " << letterGrade << endl;

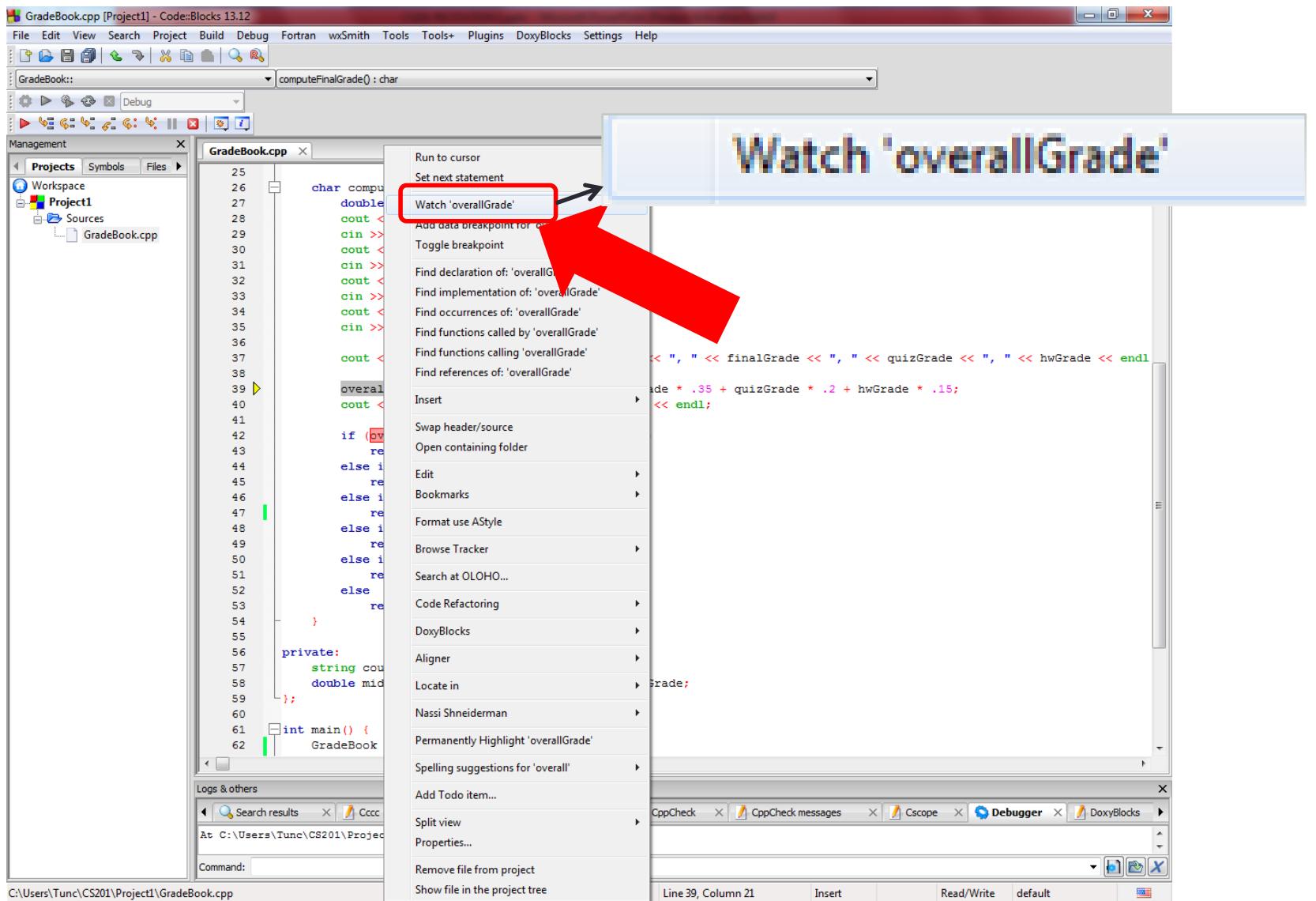
    return 0;
}
```
- Logs & others:** A tab bar at the bottom showing various logs: Search results, Ccc, Build log, Build messages, CppCheck, CppCheck messages, Cscope, Debugger, and Doxygen.
- Status Bar:** At C:\Users\Tunc\CS201\Project1\GradeBook.cpp:67, Line 67, Column 1, Insert, Read/Write, default.

GradeBook.cpp (Debugging)

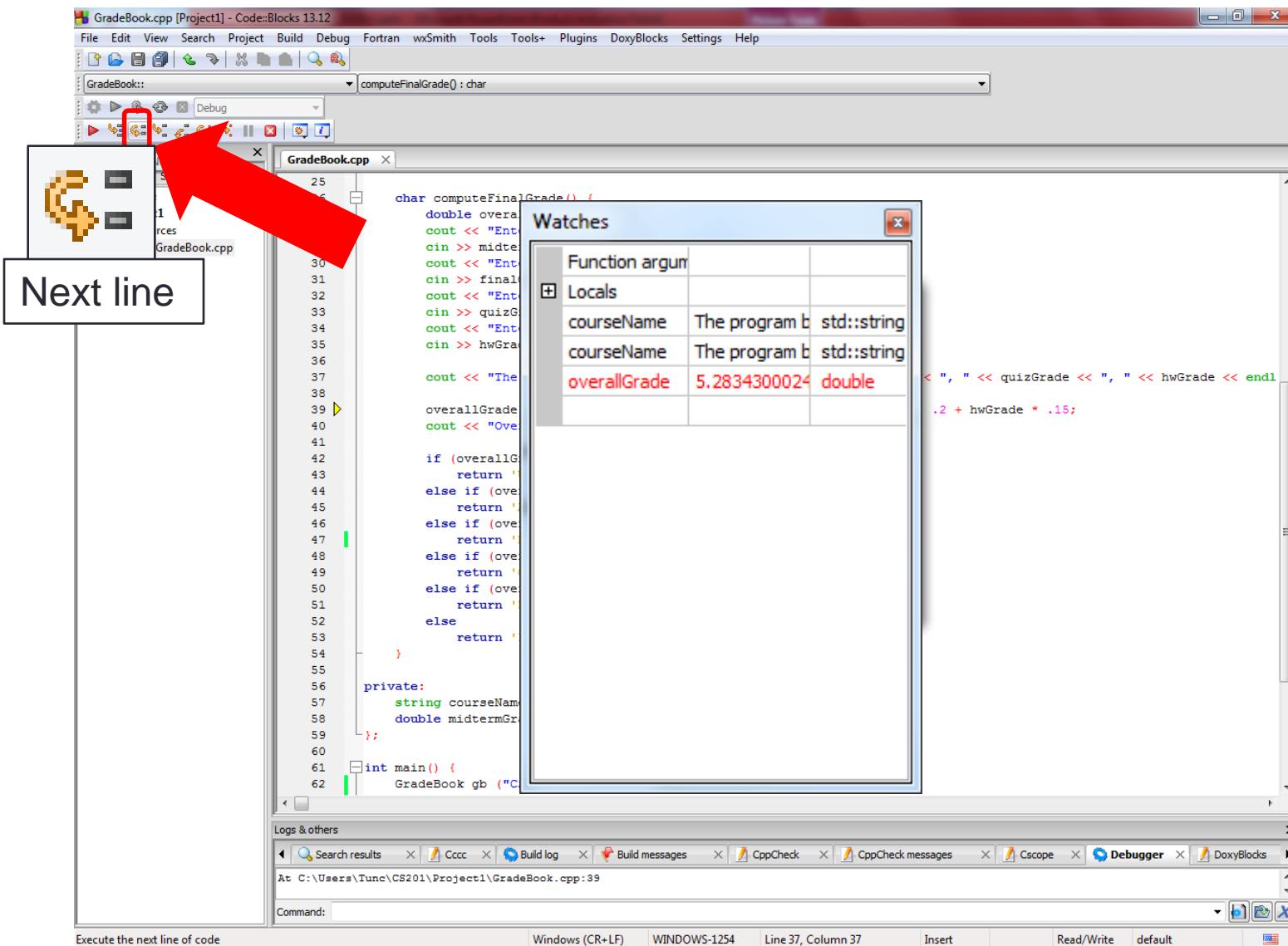
The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** GradeBook.cpp [Project1] - Code::Blocks 13.12
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Project Explorer:** Shows a workspace with a single project named "Project1" containing a source file "GradeBook.cpp".
- Code Editor:** Displays the "GradeBook.cpp" code. The code defines a class "GradeBook" with a private member "string courseName" and a public member "int main()". The "main" function creates an object "gb" of type "GradeBook" with the argument ("CS 201"). The "computeFinalGrade" method is implemented to calculate an overall grade based on four inputs: midterm, final, quiz, and homework grades, each weighted differently. It also includes a grading scale for letter grades (U, A, B, C, D, F).
- Logs & others:** A tab bar at the bottom showing various logs: Search results, Ccc, Build log, Build messages, CppCheck, CppCheck messages, Cscope, Debugger, and DoxyBlocks. The "Build log" tab is active, showing the path "At C:\Users\Tunc\CS201\Project1\GradeBook.cpp:28".
- Status Bar:** Shows the file path "C:\Users\Tunc\CS201\Project1\GradeBook.cpp", the encoding "Windows (CR+LF)", the code page "WINDOWS-1254", the current line "Line 50, Column 36", and the current mode "Insert".

GradeBook.cpp (Debugging)



GradeBook.cpp (Debugging)



GradeBook.cpp (Debugging)

The screenshot shows the Code::Blocks IDE interface during a debugging session. The main window displays the `GradeBook.cpp` file with the following code:

```
char computeFinalGrade() {
    double overallGrade;
    cout << "Enter course name: ";
    cin >> midtermGrade;
    cout << "Enter quiz grade: ";
    cin >> quizGrade;
    cout << "Enter homework grade: ";
    cin >> hwGrade;
    cout << "The overall grade is ";
    overallGrade = .2 * midtermGrade + .2 * quizGrade + .6 * hwGrade;
    cout << endl;
    if (overallGrade >= 90)
        return 'A';
    else if (overallGrade >= 80)
        return 'B';
    else if (overallGrade >= 70)
        return 'C';
    else if (overallGrade >= 60)
        return 'D';
    else
        return 'F';
}

private:
    string courseName;
    double midtermGrade;
};

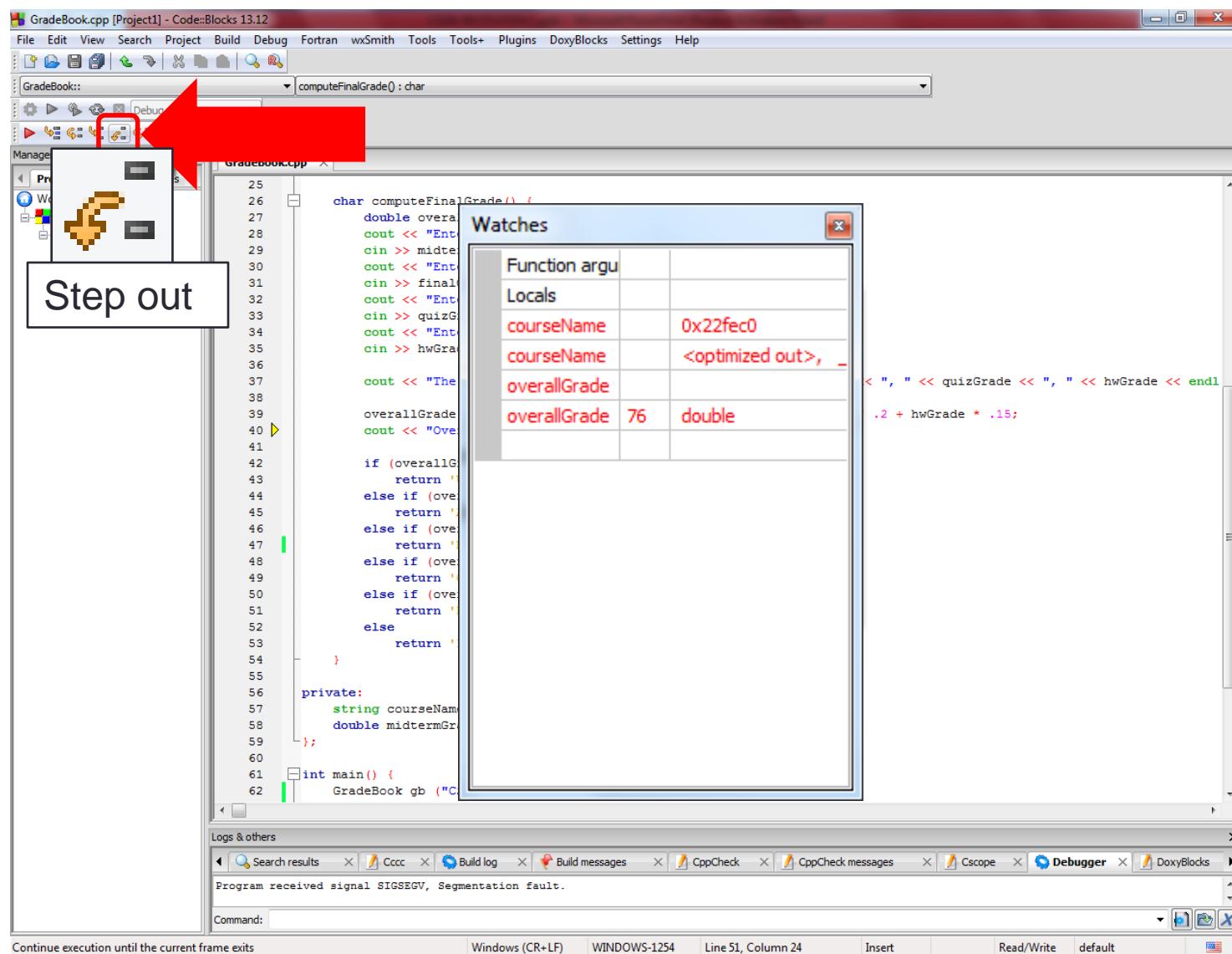
int main() {
    GradeBook gb ("Computer Science");
    cout << gb.computeFinalGrade();
}
```

A red arrow points to the `overallGrade` entry in the **Locals** section of the **Watches** window, which is highlighted with a red box. The value `76` is displayed next to it.

Function argu	
Locals	
courseName	0x22fec0
courseName	<optimized out>, _
overallGrade	
overallGrade	76

The **Logs & others** tab at the bottom shows the message: `Program received signal SIGSEGV, Segmentation fault.`

GradeBook.cpp (Debugging)



GradeBook.cpp (Debugging)

The screenshot shows the Code::Blocks IDE interface during debugging of the `GradeBook.cpp` project. The main window displays the code for `GradeBook.cpp`, which includes functions for calculating overall grades based on midterm, final, quiz, and homework scores. A floating terminal window titled "Logs & others" shows the command "Continuing..." and the file path "At C:\Users\Tunc\CS201\Project1\GradeBook.cpp:39". The code editor has several breakpoints set, indicated by red circles with yellow exclamation marks. A "Watches" window is open, showing the current values of local variables: `courseName` is "The p" and `overallGrade` is "No sy". The bottom part of the image shows a floating terminal window with the command "Continuing..." and the file path "At C:\Users\Tunc\CS201\Project1\GradeBook.cpp:39".

```
cout << "The grades are " << midtermGrade << " " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;
        .2 + hwGrade * .15;

private:
    string courseName;
    double midtermGrade;
    double finalGrade;
    double quizGrade;
    double hwGrade;
    char overallGrade;

    overallGrade = .2 + hwGrade * .15;

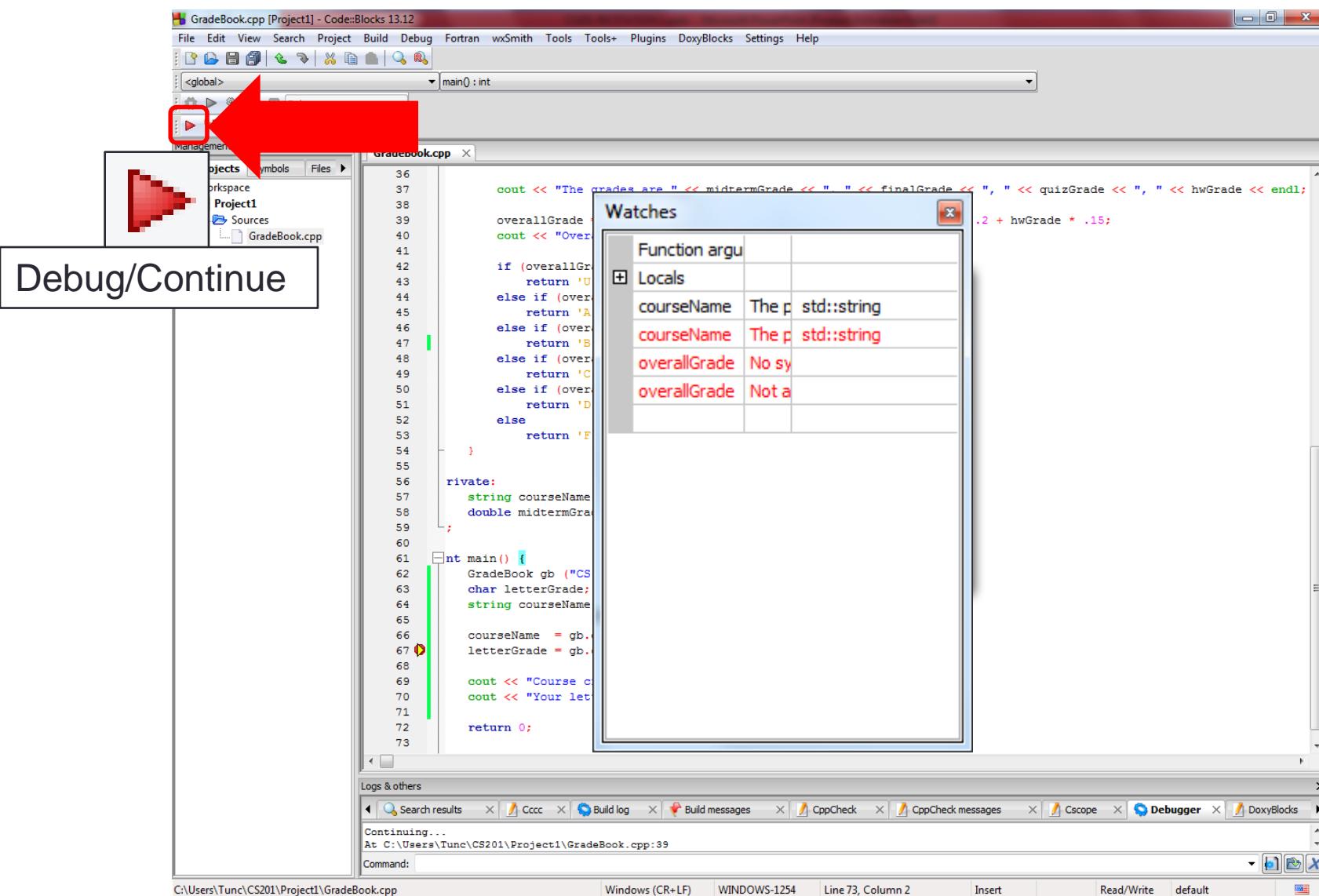
    if (overallGrade >= 90)
        return 'A';
    else if (overallGrade >= 80)
        return 'B';
    else if (overallGrade >= 70)
        return 'C';
    else if (overallGrade >= 60)
        return 'D';
    else
        return 'F';

    cout << "Course name: " << courseName << endl;
    cout << "Your letter grade is " << overallGrade << endl;

    return 0;
}
```

Function argu	
Locals	
courseName	The p std::string
courseName	The p std::string
overallGrade	No sy
overallGrade	Not a

GradeBook.cpp (Debugging)



GradeBook.cpp (Debugging)

- Step Into :
 - Runs the program until the next instruction is reached.
- Next Line :
 - Runs the program until the next line of code is reached.
- Step Out :
 - Runs the program until the current procedure is completed.

Step Out ≥ Next Line ≥ Step Into

At Break Point

Output :
Instruction A1
—

```
1 #include <iostream>
2 using namespace std;
3 void procedureB(){
4     cout << "Instruction B1" << endl;
5     cout << "Instruction B2" << endl;
6 }
7 void procedureA(){
8     cout << "Instruction A1" << endl;
9     procedureB();
10    cout << "Instruction A2" << endl;
11    cout << "Instruction A3" << endl;
12 }
13 int main(){
14     cout << "Output : " << endl;
15     procedureA();
16     cout << "End of Procedure A" << endl;
17 }
```

Step Into



Output :
Instruction A1

```
3 void procedureB(){
4     cout << "Instruction B1" << endl;
5     cout << "Instruction B2" << endl;
6 }
```

Next Line



Output :
Instruction A1
Instruction B1
Instruction B2

```
7 void procedureA(){
8     cout << "Instruction A1" << endl;
9     procedureB();
10    cout << "Instruction A2" << endl;
11    cout << "Instruction A3" << endl;
12 }
```

Step Out



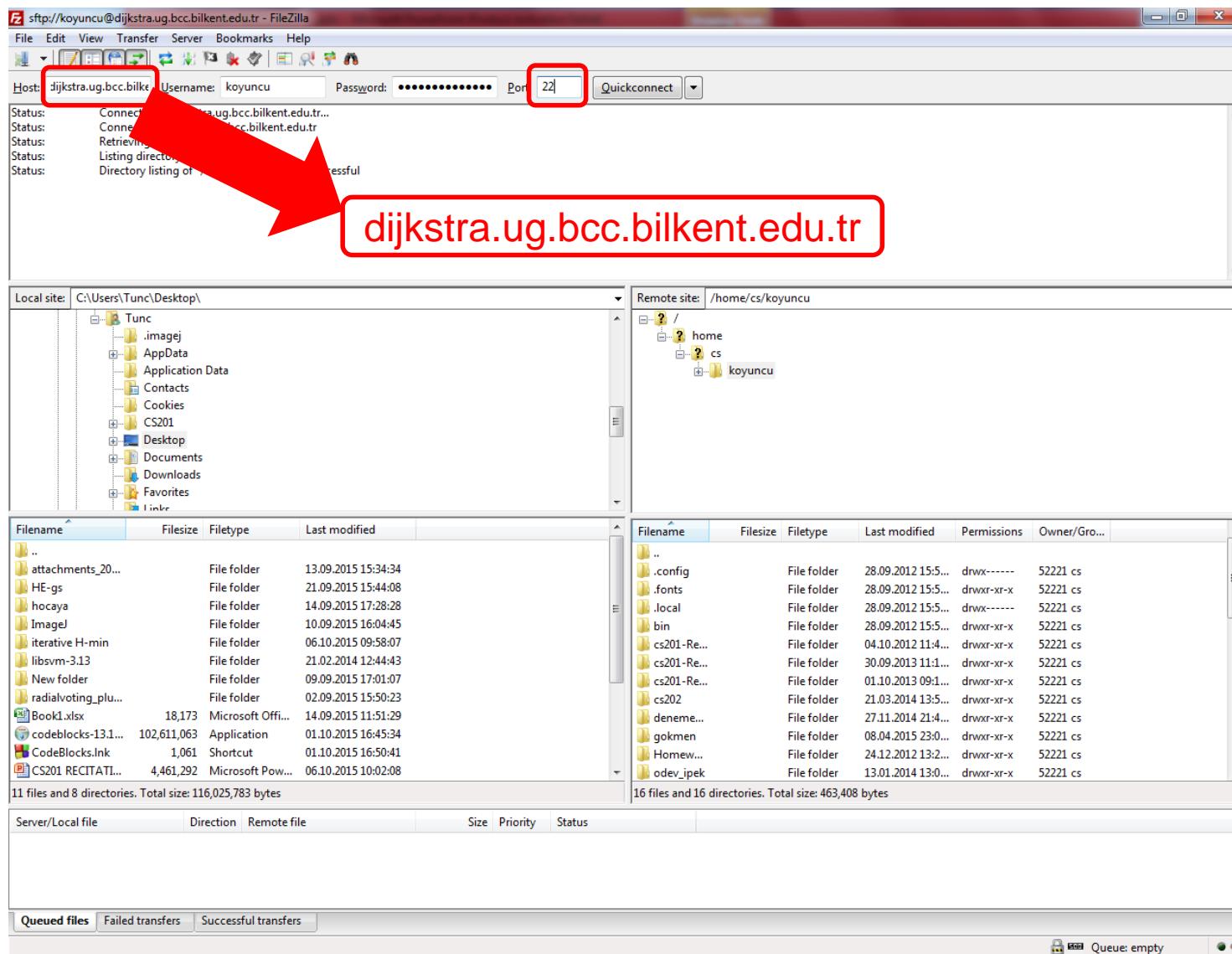
Output :
Instruction A1
Instruction B1
Instruction B2
Instruction A2
Instruction A3

```
13 int main(){
14     cout << "Output : " << endl;
15     procedureA();
16     cout << "End of Procedure A" << endl;
17 }
```

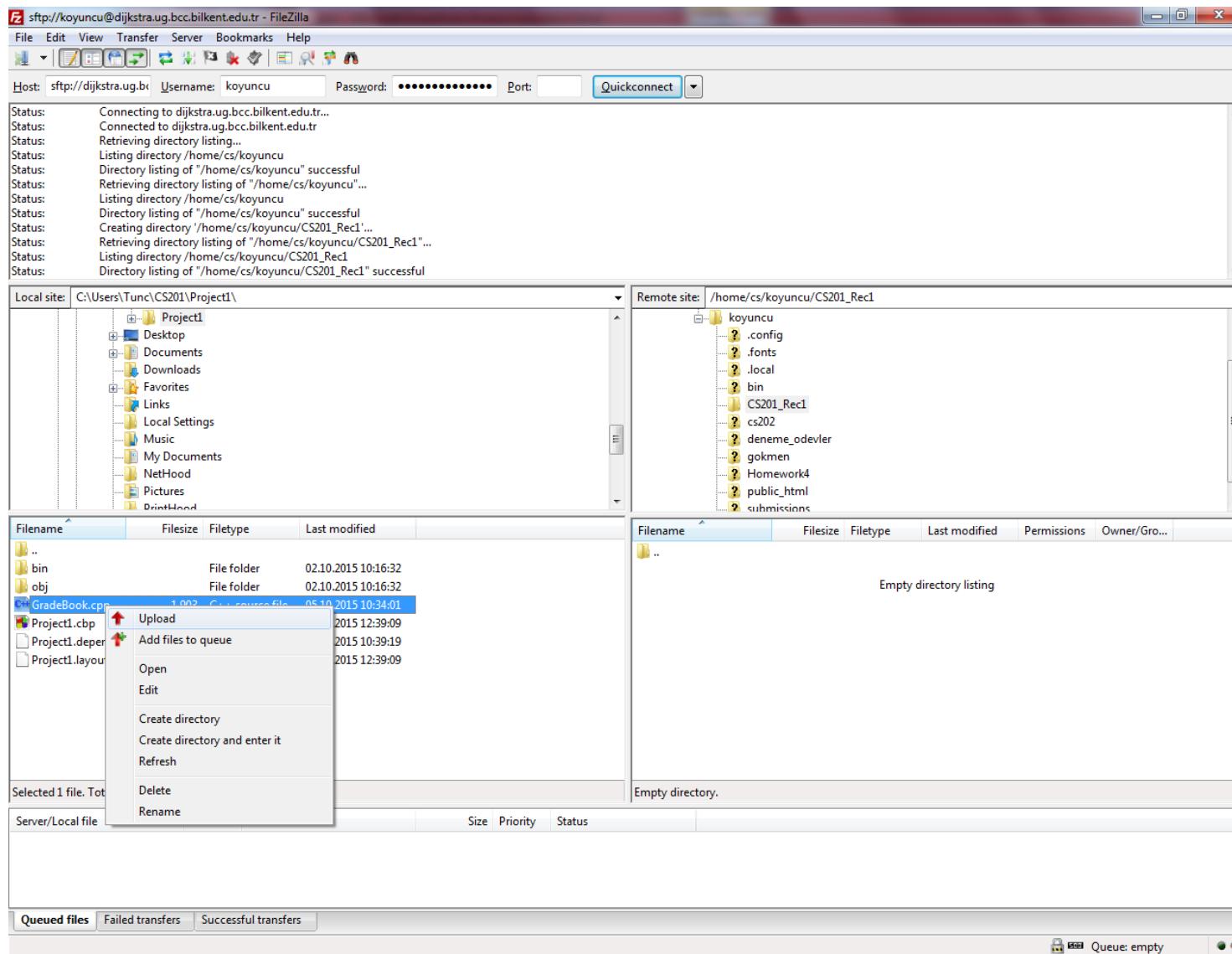
Part 2: Porting, compiling and testing in Dijkstra

- FileZilla (FTP client) + PuTTY (SSH Client)
 - FileZilla
 - <https://filezilla-project.org/download.php?type=client>
 - PuTTY
 - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- SSH Secure Shell

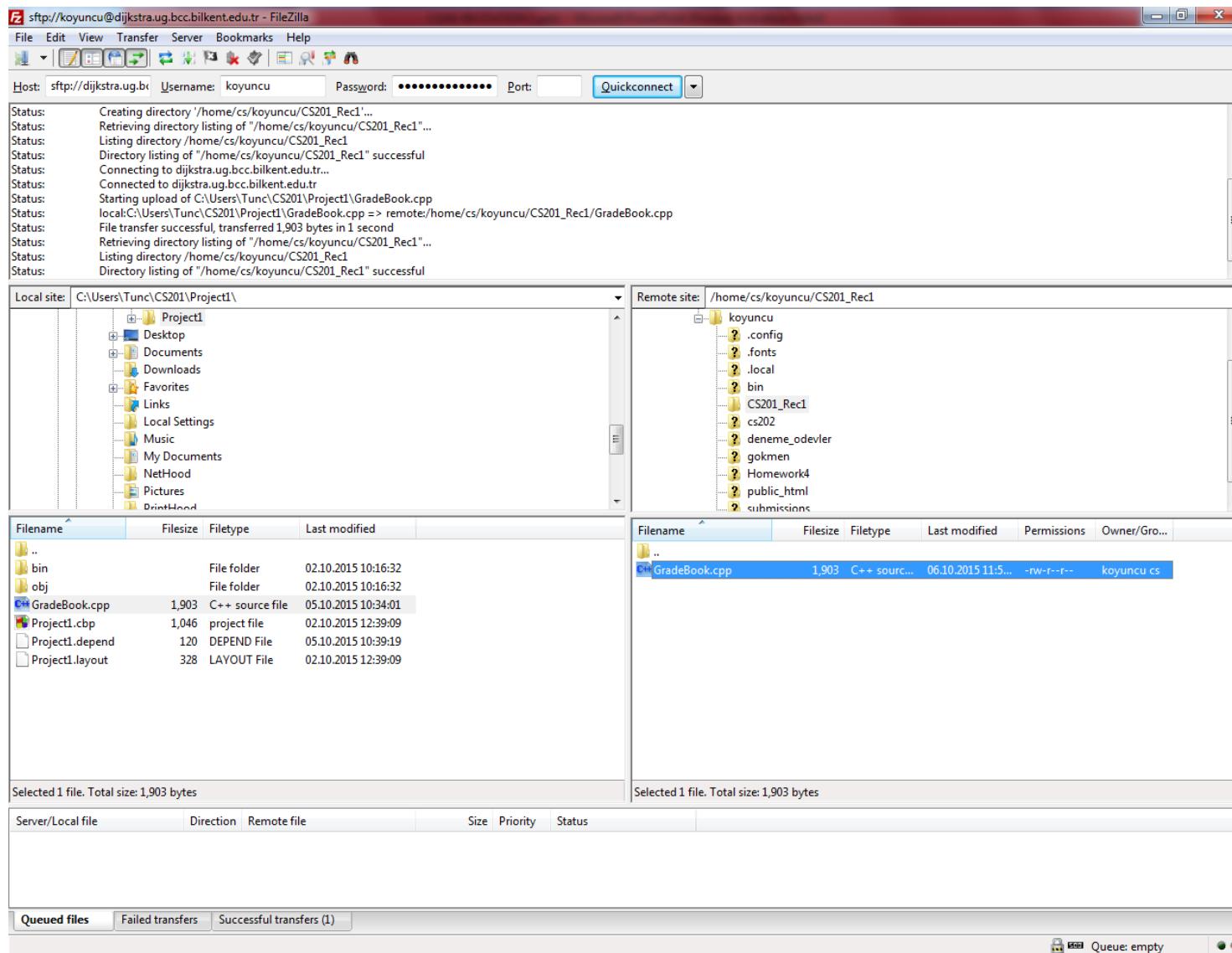
Part 2: FileZilla



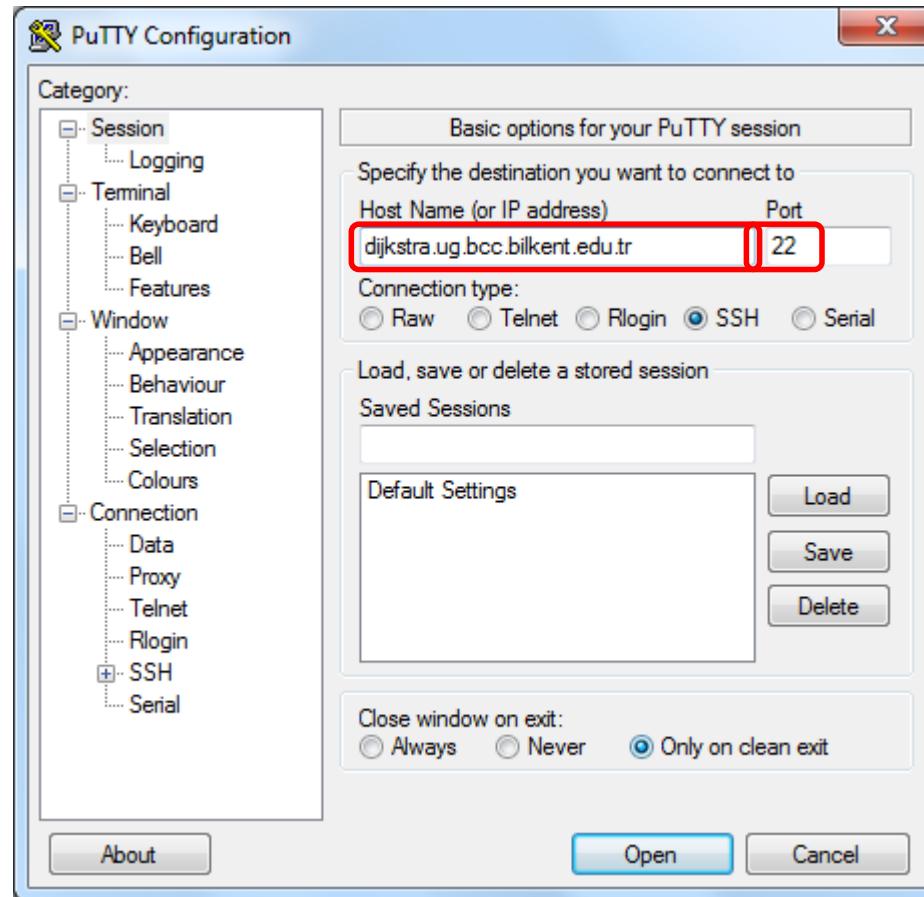
Part 2: FileZilla



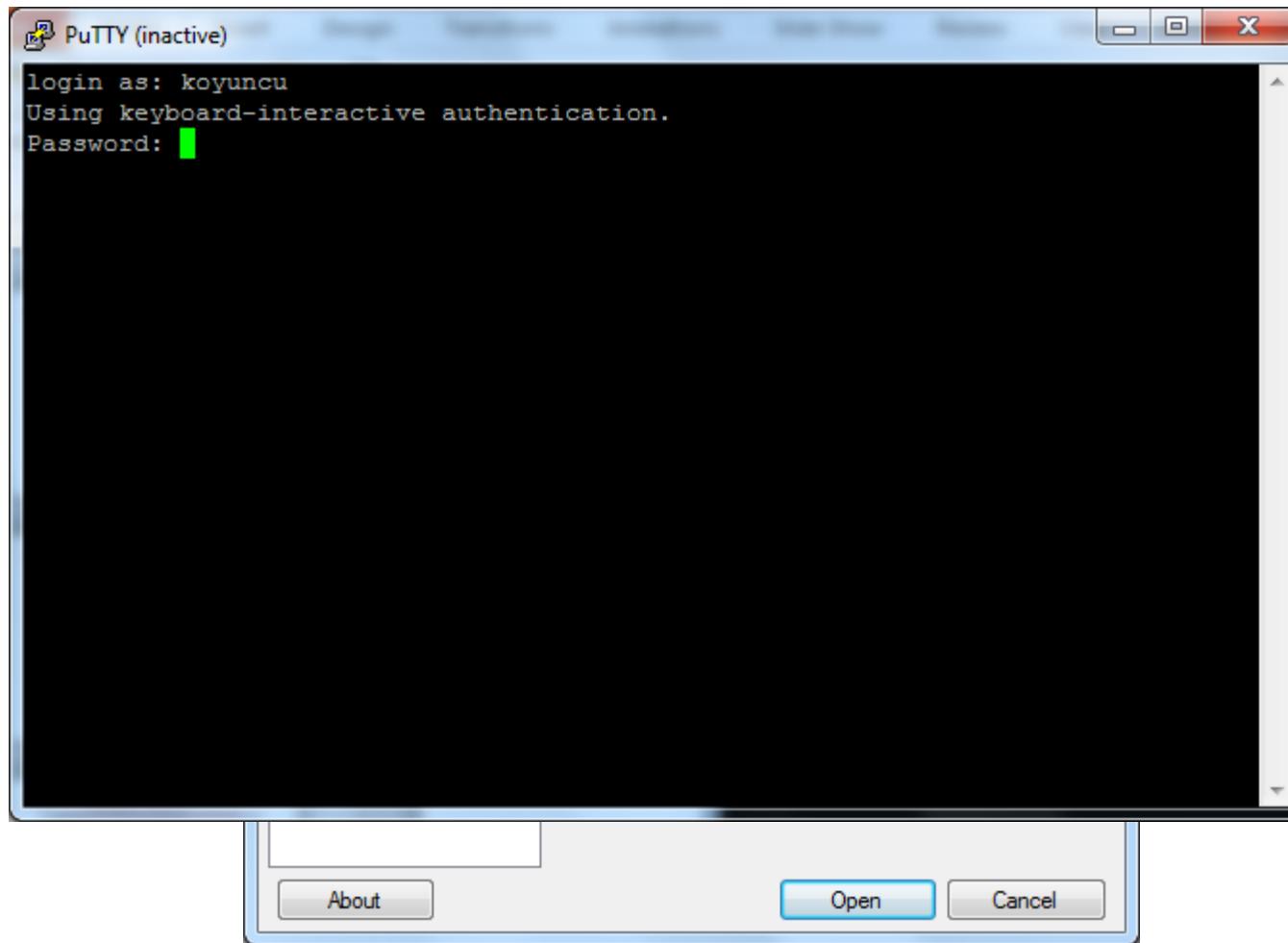
Part 2: FileZilla



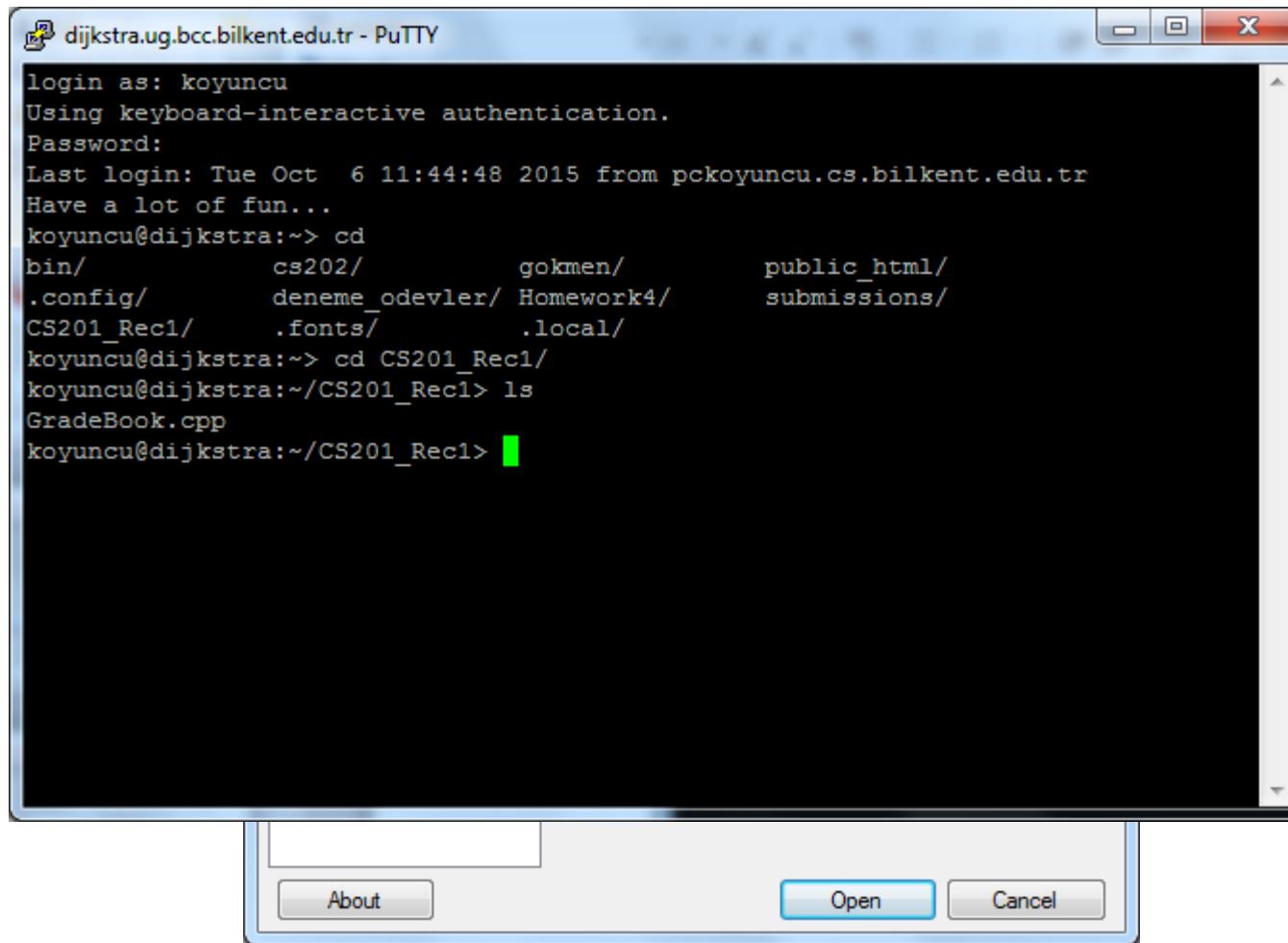
Part 2: PuTTY



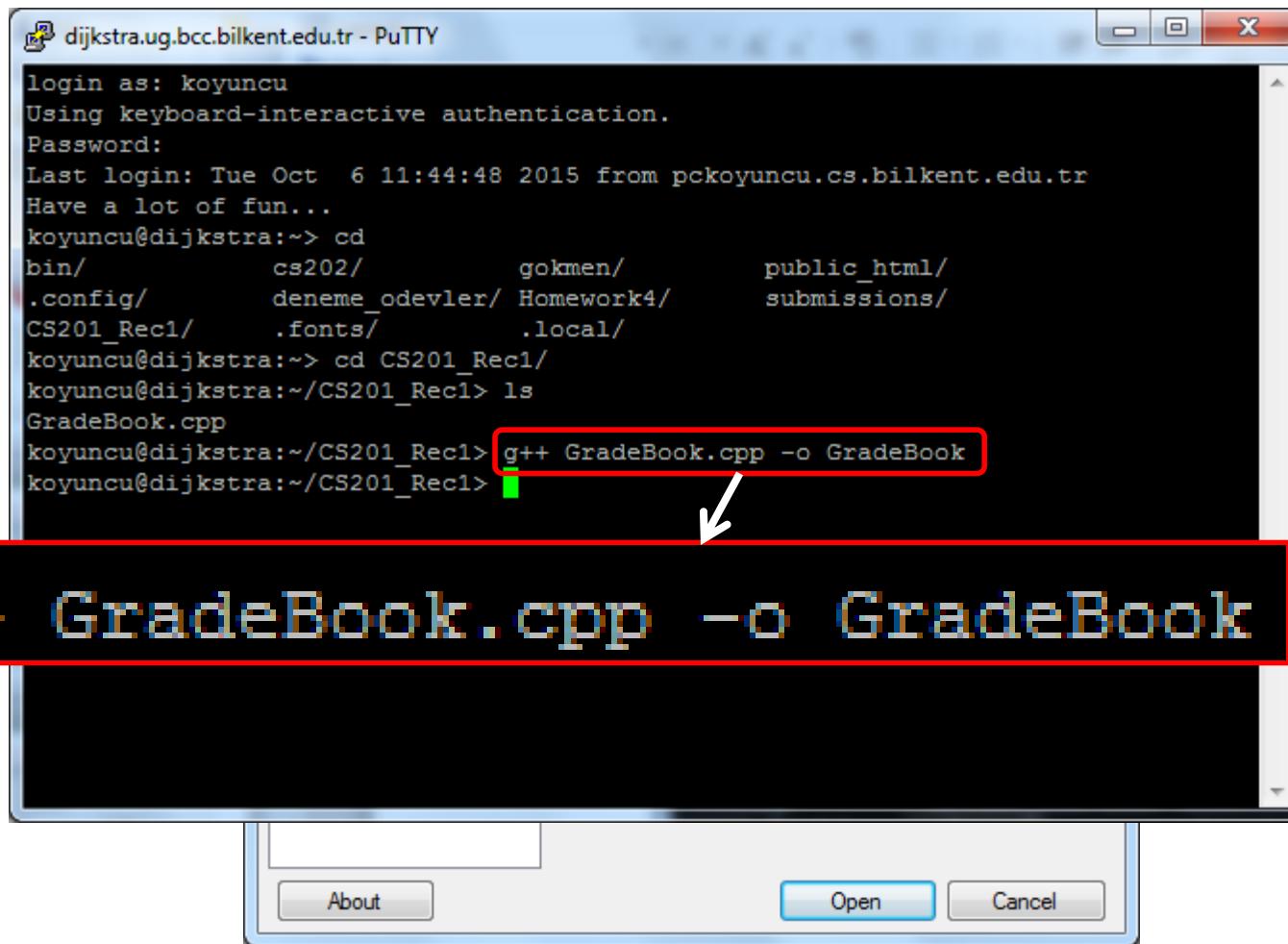
Part 2: PuTTY



Part 2: PuTTY



Part 2: PuTTY



```
dijkstra.ug.bcc.bilkent.edu.tr - PuTTY
login as: koyuncu
Using keyboard-interactive authentication.
Password:
Last login: Tue Oct  6 11:44:48 2015 from pckoyuncu.cs.bilkent.edu.tr
Have a lot of fun...
koyuncu@dijkstra:~> cd
bin/          cs202/          gokmen/          public_html/
.config/      deneme_odevler/  Homework4/      submissions/
CS201_Rec1/   .fonts/         .local/
koyuncu@dijkstra:~> cd CS201_Rec1/
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook.cpp
koyuncu@dijkstra:~/CS201_Rec1> g++ GradeBook.cpp -o GradeBook
koyuncu@dijkstra:~/CS201_Rec1>
```

g++ GradeBook.cpp -o GradeBook

Part 2: PuTTY

The screenshot shows a PuTTY terminal window titled "dijkstra.ug.bcc.bilkent.edu.tr - PuTTY". The session log displays the following commands and output:

```
login as: koyuncu
Using keyboard-interactive authentication.
Password:
Last login: Tue Oct  6 11:44:48 2015 from pckoyuncu.cs.bilkent.edu.tr
Have a lot of fun...
koyuncu@dijkstra:~> cd
bin/          cs202/          gokmen/          public_html/
.config/      deneme_odevler/  Homework4/      submissions/
CS201_Rec1/   .fonts/         .local/
koyuncu@dijkstra:~> cd CS201_Rec1/
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook.cpp
koyuncu@dijkstra:~/CS201_Rec1> g++ GradeBook.cpp -o GradeBook
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook  GradeBook.cpp
koyuncu@dijkstra:~/CS201_Rec1> ./GradeBook
```

A red box highlights the command `./GradeBook`, and a red arrow points from this command to a large, bold, multi-colored text overlay at the bottom of the terminal window that also reads `./GradeBook`.

Part 2: PuTTY

```
dijkstra.ug.bcc.bilkent.edu.tr - PuTTY

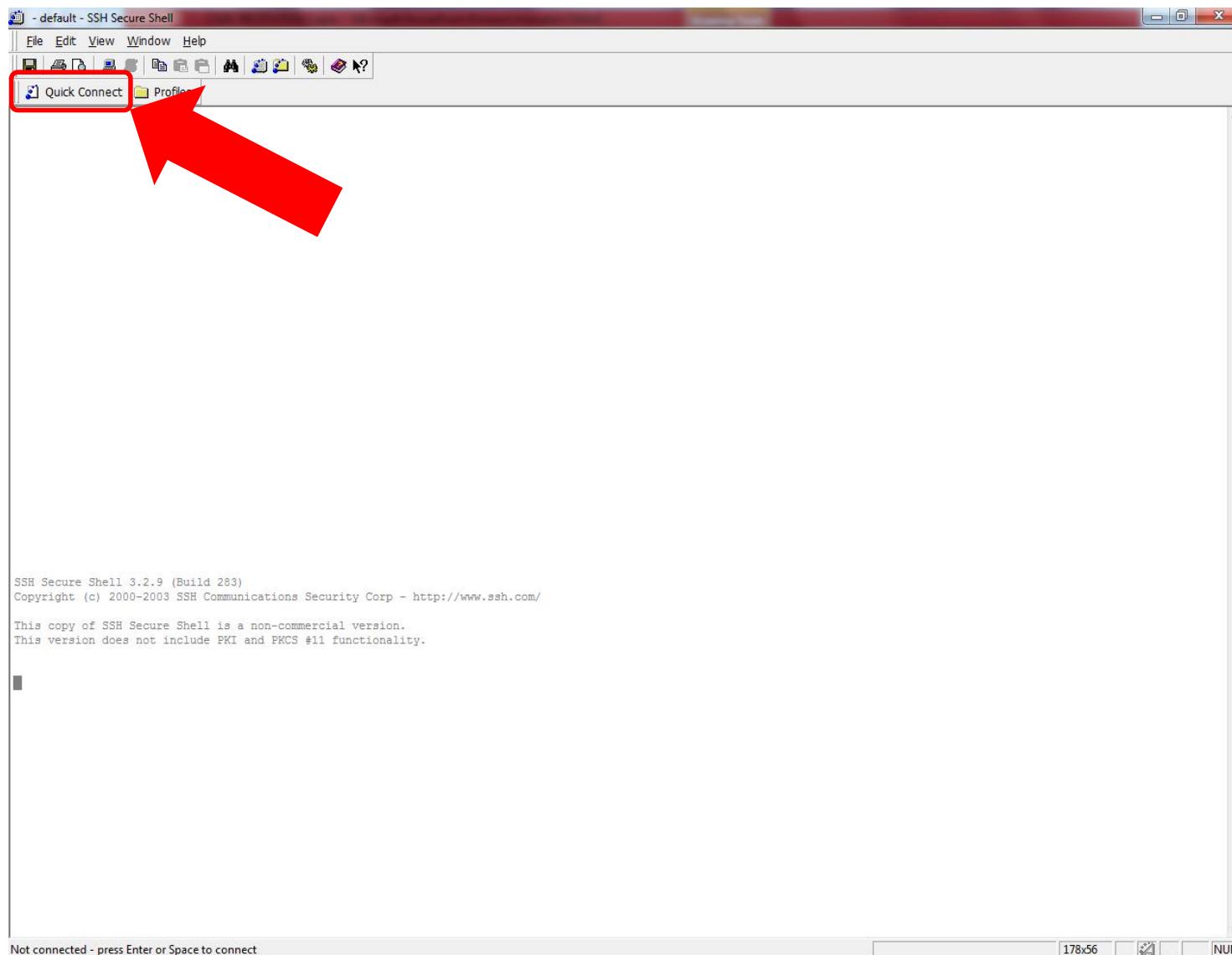
login as: koyuncu
Using keyboard-interactive authentication.
Password:
Last login: Tue Oct  6 13:21:41 2015 from 139.179.50.40
Have a lot of fun...
koyuncu@dijkstra:~> cd
bin/          cs202/          gokmen/          public_html/
.config/      deneme_odevler/  Homework4/      submissions/
CS201_Rec1/   .fonts/         .local/
koyuncu@dijkstra:~> cd CS201_Rec1/
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook  GradeBook.cpp
koyuncu@dijkstra:~/CS201_Rec1> ./GradeBook
Enter the midterm grade : 85
Enter the final grade : 90
Enter the quiz grade : 70
Enter the homework grade : 85
The grades are 85, 90, 70, 85
Overall grade : 83.75
Course created with course name CS 201
Your letter grade : B
koyuncu@dijkstra:~/CS201_Rec1>
```

The screenshot shows a PuTTY terminal window with a blue title bar containing the text "dijkstra.ug.bcc.bilkent.edu.tr - PuTTY". The main area of the window is a black terminal session. It starts with a login prompt "login as: koyuncu", followed by a password prompt. Below that, it shows the user's last login details and a message "Have a lot of fun...". The user then changes directory to their home folder and lists its contents. They change to the "CS201_Rec1" directory and list its contents again, showing files named "GradeBook" and "GradeBook.cpp". The user runs the "GradeBook" script, which asks for four grades: midterm, final, quiz, and homework. The script then calculates the average and outputs the overall grade (83.75) and the corresponding letter grade (B). A red rectangular box highlights the command-line interaction from the first grade prompt ("Enter the midterm grade : 85") to the final output ("Your letter grade : B"). At the bottom of the terminal window, there is a small menu with three buttons: "About", "Open", and "Cancel".

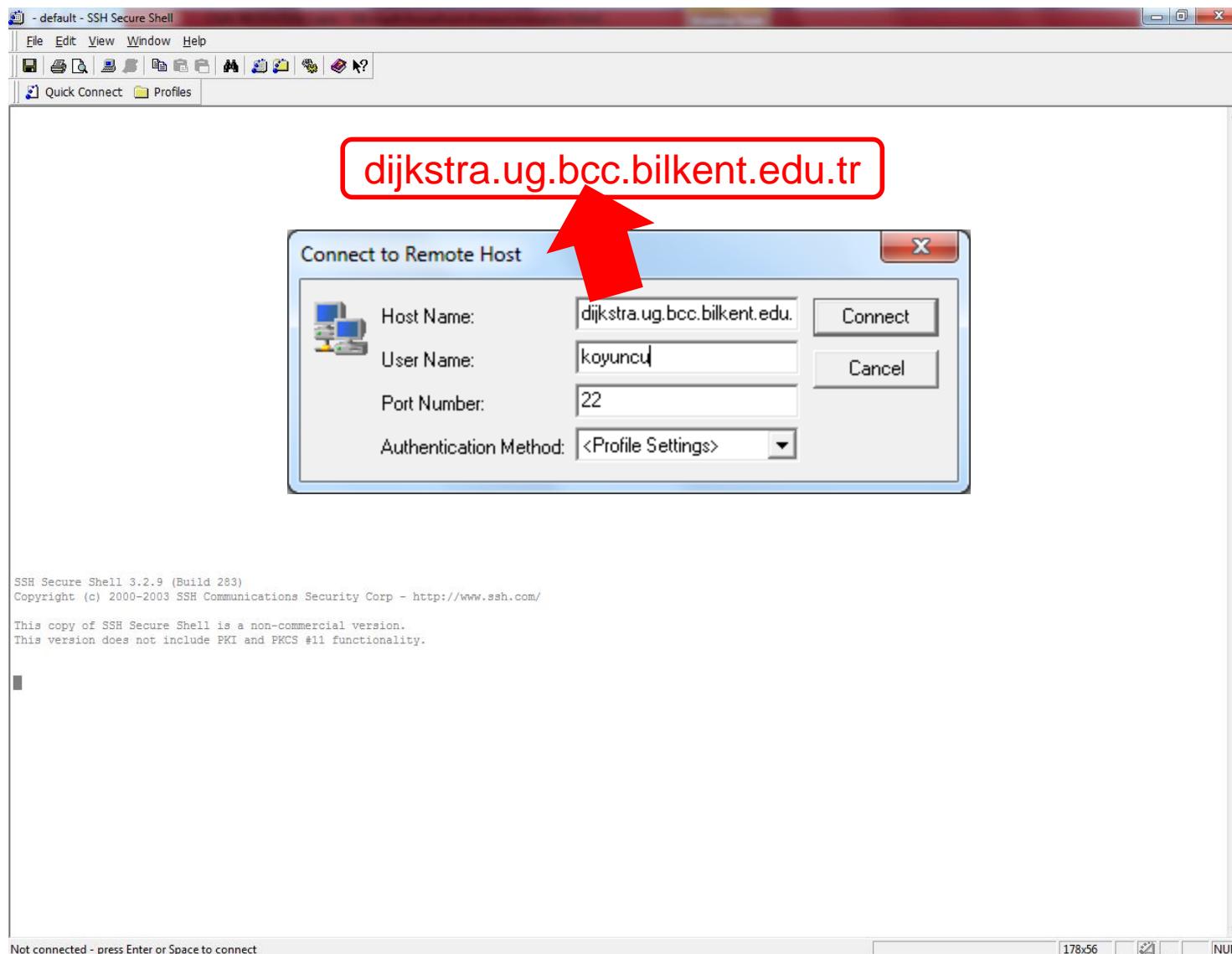
Part 2: PuTTY

- The base command for the Gnu C++ compiler is **g++**
- Single File Programs
 - The easiest compilation uses the command format:
 - `g++ -o <outputName> <cppFile>`
 - Example:
 - `g++ -o myExe prog1.cpp`
- Multiple File Programs
 - `g++ -o <outputName> <cppFile1> <cppFile2> ...`
 - Example:
 - `g++ -o myProgram thing.cpp main.cpp`
 - This command compiles and links the code files "thing.cpp" and "main.cpp" together into the executable program called "myProgram".
 - `g++ -o myProgram *.cpp`
 - This command compiles and links all the code files with ".cpp" extension.

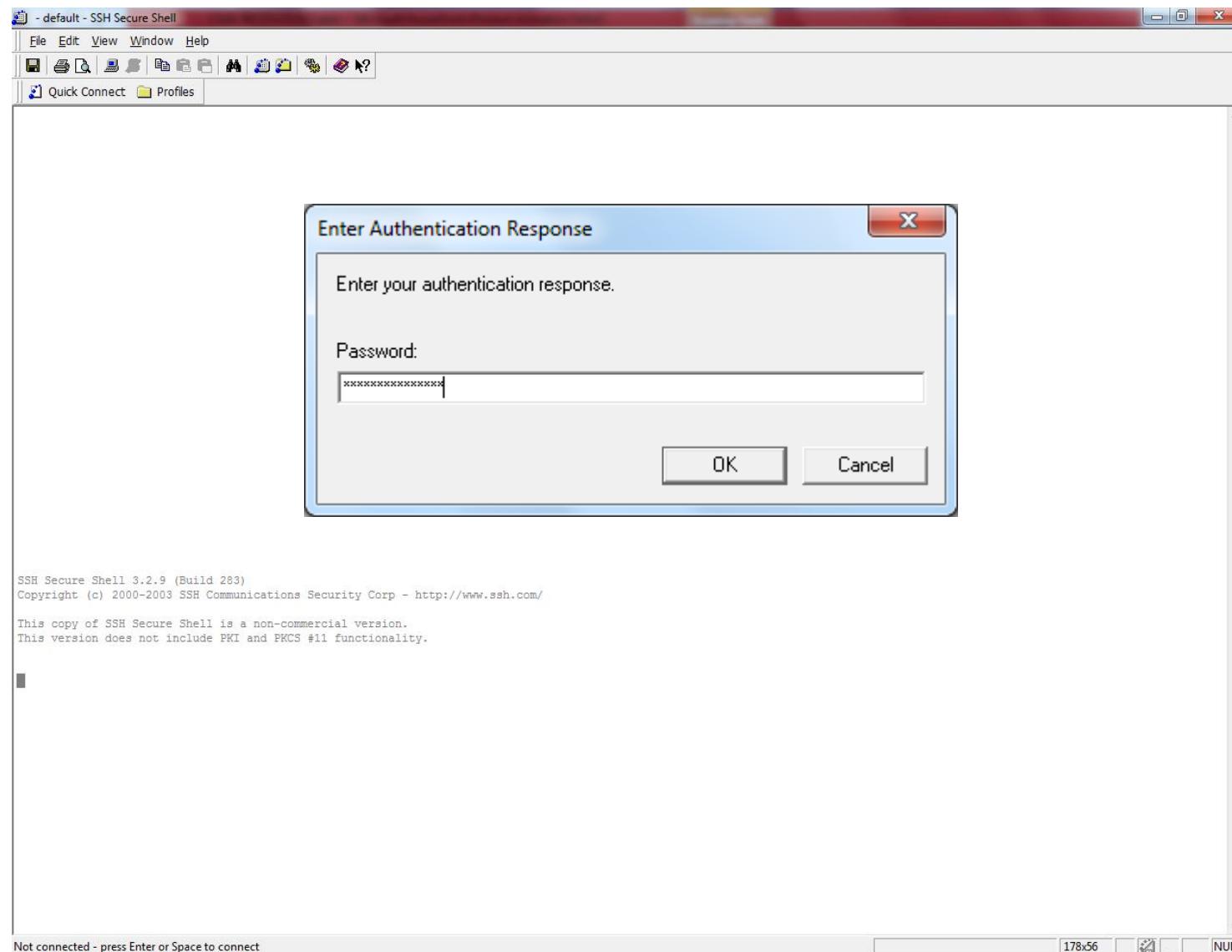
Part 2: SSH Secure Shell



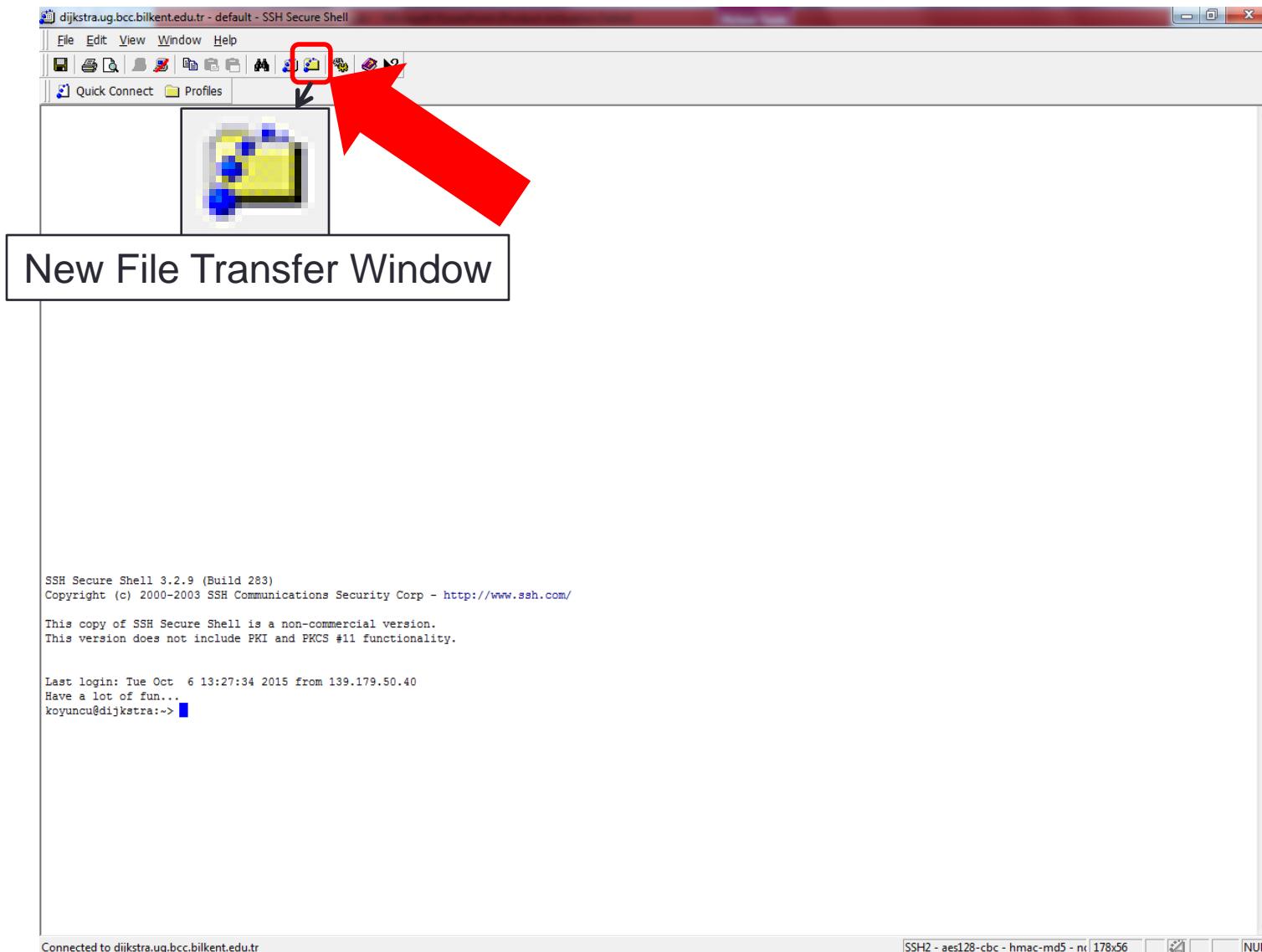
Part 2: SSH Secure Shell



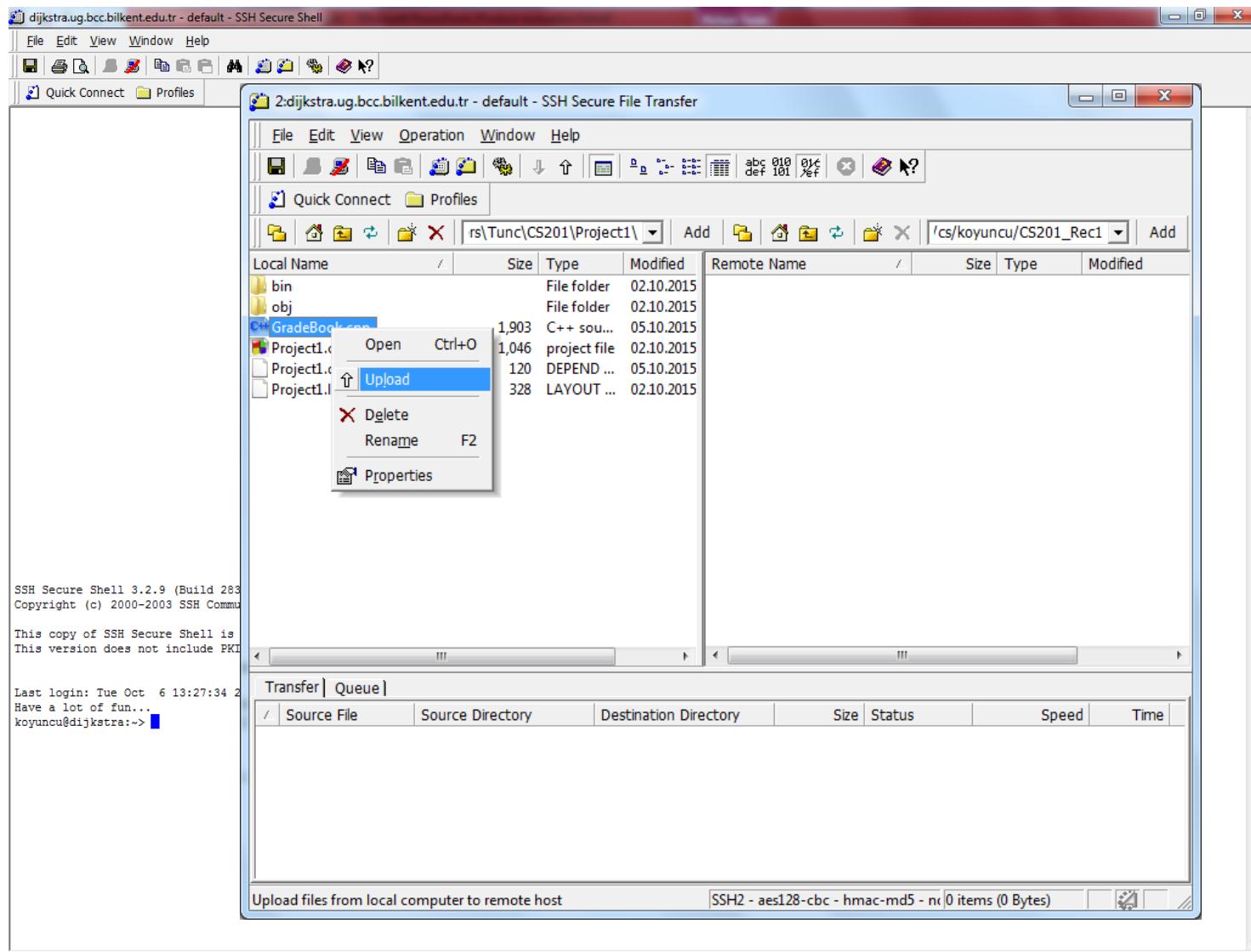
Part 2: SSH Secure Shell



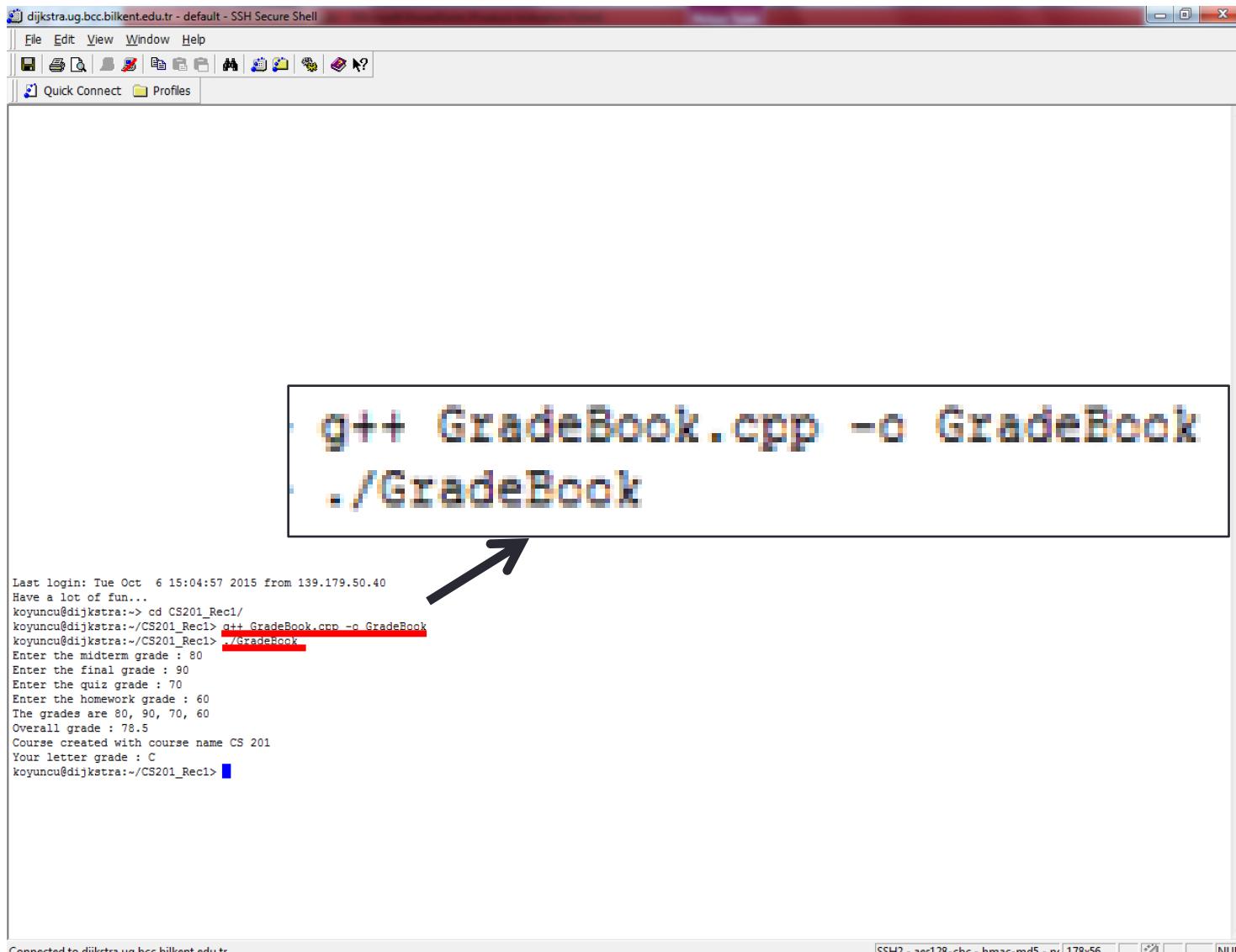
Part 2: SSH Secure Shell



Part 2: SSH Secure Shell



Part 2: SSH Secure Shell



The screenshot shows a Windows-style SSH Secure Shell window titled "dijkstra.ug.bcc.bilkent.edu.tr - default - SSH Secure Shell". The window has a menu bar with File, Edit, View, Window, Help and a toolbar with various icons. Below the toolbar is a status bar with "Quick Connect" and "Profiles". The main terminal area contains the following text:

```
g++ GradeBook.cpp -o GradeBook
./GradeBook
```

A black arrow points from the bottom of the terminal area up towards the command line above it. The terminal history at the bottom shows the user's session:

```
Last login: Tue Oct  6 15:04:57 2015 from 139.179.50.40
Have a lot of fun...
koyuncu@dijkstra:~> cd CS201_Rec1/
koyuncu@dijkstra:~/CS201_Rec1> g++ GradeBook.cpp -o GradeBook
koyuncu@dijkstra:~/CS201_Rec1> ./GradeBook
Enter the midterm grade : 80
Enter the final grade : 90
Enter the quiz grade : 70
Enter the homework grade : 60
The grades are 80, 90, 70, 60
Overall grade : 78.5
Course created with course name CS 201
Your letter grade : C
koyuncu@dijkstra:~/CS201_Rec1>
```

At the very bottom of the window, a status bar displays "Connected to dijkstra.ug.bcc.bilkent.edu.tr" and "SSH2 - aes128-cbc - hmac-md5 - nc | 178x56".

Part 2: Types of error

- Compile Time errors
 - Syntax errors
 - Undeclared variables and functions, improper function calls etc.
 - e.g. Forgetting to put semicolon(;) at the end of an instruction.

```
cout << "Hello World" << endl // No Semicolon();
```
 - Result : `error: expected ';' before 'return'`
- Linker errors
 - Undefined functions or multiply defined functions or symbols
 - e.g. Not including correct header files → `#include <iostream>`
 - Not using the correct namespace → `using std::cin;`
 - Result : `error: 'cin' was not declared in this scope`

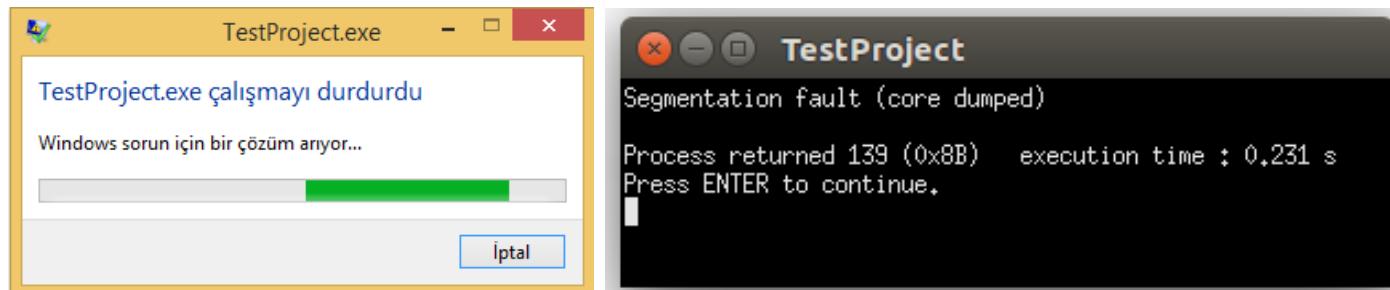
- Run-time errors

- Fatal Errors

- Typically cause the program to crash during execution
 - e.g. Trying to access a non-existent memory location.

```
char* p = NULL;  
cout << *p << endl;
```

Result :



- Non-Fatal(Logical) Errors

- Does not crash the program but produce erroneous results
 - Typically hardest to detect
 - Result : Incorrect program behaviour

Part 3: Using header files

Why do we need header files?

1. Speeds up compilation time
 - Upon the change of a single line of code;
Without headers : All of the code needs to be recompiled
With headers : Only changing parts need to be recompiled
2. Keeps the code organized
 - Necessary for big projects
 - Allows multiple people to work on the same project

For more info : [Headers and Includes: Why and How](#)

Part 3: Using header files

- Back to GradeBook.cpp
- Let's try and separate this file into multiple files separating the interface of the class from its implementation as well as separating the user program that uses this code.

Part 3: Using header files

The screenshot shows the Code::Blocks IDE interface. The title bar reads "GradeBook.cpp [Project1] - Code::Blocks 13.12". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. A context menu is open over the code editor, with "New" selected. Sub-options under "New" include Empty file (Ctrl-Shift-N), Class..., Project..., Build target..., File..., Custom..., and From template... . The main code editor window displays C++ code for calculating an overall grade based on midterm, final, quiz, and homework grades. Below the code editor is a "Logs & others" panel showing build logs and messages. The status bar at the bottom indicates "Quickly create a new empty file" and provides file path information: "Windows (CR+LF)" and "C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe". It also shows line and column numbers: "Line 28, Column 45".

```
double overallGrade;
cout << "Enter the midterm grade : ";
cin >> midtermGrade;
cout << "Enter the final grade : ";
cin >> finalGrade;
cout << "Enter the quiz grade : ";
cin >> quizGrade;
cout << "Enter the homework grade : ";
cin >> hwGrade;

cout << "The grades are " << midtermGrade << ", " << finalGrade << ", " << quizGrade << ", " << hwGrade << endl;

overallGrade = midtermGrade * .3 + finalGrade * .35 + quizGrade * .2 + hwGrade * .15;
cout << "Overall grade : " << overallGrade << endl;

if (overallGrade > 100 || overallGrade < 0)
    return 'U';
else if (overallGrade >= 90)
    return 'A';
else if (overallGrade >= 80)
    return 'B';
else if (overallGrade >= 70)
    return 'C';
else if (overallGrade >= 60)
    return 'D';
else
    return 'F';

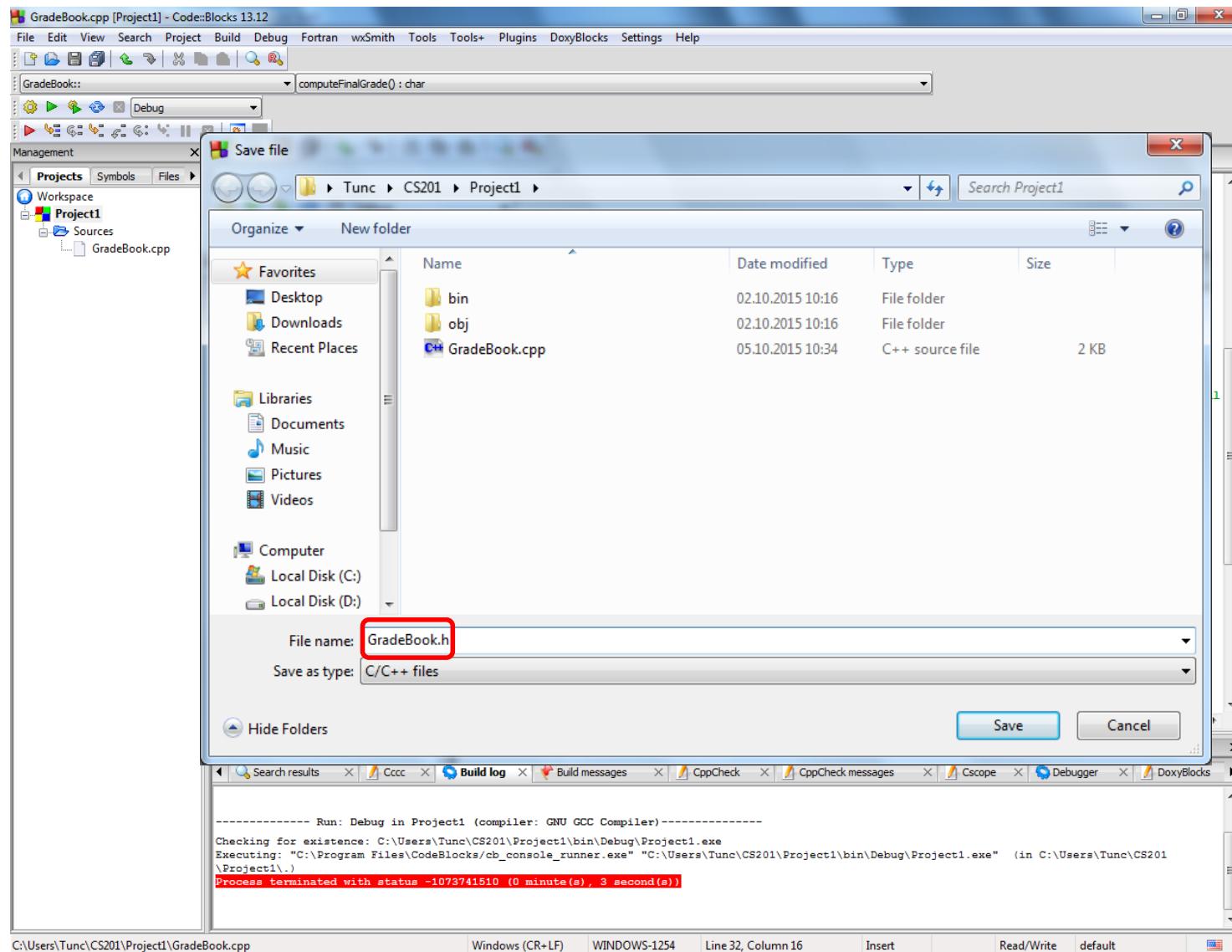
private:
```

Logs & others

```
----- Run: Debug in Project1 (compiler: GNU GCC Compiler) -----
Checking for existence: C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe"  (in C:\Users\Tunc\CS201\Project1.)
Process terminated with status -1073741510 (0 minute(s), 3 second(s))
```

Windows (CR+LF) | C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe | Line 28, Column 45 | Insert | Read/Write | default |

Part 3: Using header files



Part 3: Using header files

The screenshot shows the Code::Blocks IDE interface. The title bar reads "GradeBook.h - Code::Blocks 13.12". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar contains various icons for file operations like Open, Save, and Build. A "Management" dock window is open, showing a "Projects" tab with "Project1" selected, which contains a "Sources" folder with "GradeBook.cpp". The main code editor window has two tabs: "GradeBook.cpp" and "GradeBook.h". The "GradeBook.h" tab is active and displays the following C++ code:

```
#include <string>
using std::string;

class GradeBook {
public:
    GradeBook();
    void setCourseName (string);
    string getCourseName();
    char computeFinalGrade();

private:
    string courseName;
    double midtermGrade, finalGrade, quizGrade, hwGrade;
};
```

A large white box with a black border is overlaid on the code editor, containing the text "Interface (Header, .h) File". The bottom part of the interface shows the "Logs & others" dock window with several tabs: Search results, Ccc, Build log, Build messages, CppCheck, CppCheck messages, Cscope, Debugger, and DoxyBlocks. The "Build log" tab is active and shows the following output:

```
----- Run: Debug in Project1 (compiler: GNU GCC Compiler) -----
Checking for existence: C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe"  (in C:\Users\Tunc\CS201\Project1)
Process terminated with status -1073741510 (0 minute(s), 3 second(s))
```

The status bar at the bottom shows the path "C:\Users\Tunc\CS201\Project1\GradeBook.h", file format "Windows (CR+LF)", encoding "default", line "Line 15, Column 1", and write status "Insert Read/Write default".

Part 3: Using header files

Implementation (.cpp) File

The screenshot shows the Code::Blocks IDE interface. The title bar reads "GradeBook.cpp [Project1] - Code::Blocks 13.12". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has icons for file operations like Open, Save, and Build. The left sidebar has a "Management" tab with "Projects", "Symbols", and "Files" options, showing "Project1" with "Sources" and "GradeBook.cpp". The main code editor window displays "GradeBook.cpp" with the following content:

```
#include "GradeBook.h"
#include <iostream>
using std::cout;
using std::endl;
using std::cin;

GradeBook::GradeBook (string name) {
    setCourseName (name);
}

void GradeBook::setCourseName (string name) {
    if (name.length () <= 25)
        courseName = name;
    else
        courseName = name.substr (0, 25);

}

string GradeBook::getCourseName () {
    return courseName;
}

char GradeBook::computeFinalGrade () {
    double overallGrade;
    cout << "Enter the midterm grade : ";
    cin >> midTermGrade;
    cout << "Enter the final grade : ";
    cin >> finalGrade;
    cout << "Enter the quiz grade : ";
    cin >> quizGrade;
    cout << "Enter the homework grade : ";
    cin >> hwGrade;
}
```

The "Logs & others" window at the bottom shows the build log:

```
----- Run: Debug in Project1 (compiler: GNU GCC Compiler) -----
Checking for existence: C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe"  (in C:\Users\Tunc\CS201\Project1)
Process terminated with status -1073741510 (0 minute(s), 3 second(s))
```

The status bar at the bottom shows the file path "C:\Users\Tunc\CS201\Project1\GradeBook.cpp", encoding "Windows (CR+LF)", character set "WINDOWS-1254", line "Line 1, Column 23", and other settings like "Insert", "Read/Write", and "default".

Part 3: Using header files

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for `main.cpp`. A callout box highlights the line `#include "GradeBook.h"`. The code block is as follows:

```
#include "GradeBook.h"
#include <iostream>

using std::cout;
using std::endl;
int main() {
    GradeBook gb ("CS 201");
    char letterGrade;
    string courseName;

    courseName = gb.getCourseName();
    letterGrade = gb.computeFinalGrade();

    cout << "Course created with course name " << courseName << endl;
    cout << "Your letter grade : " << letterGrade << endl;

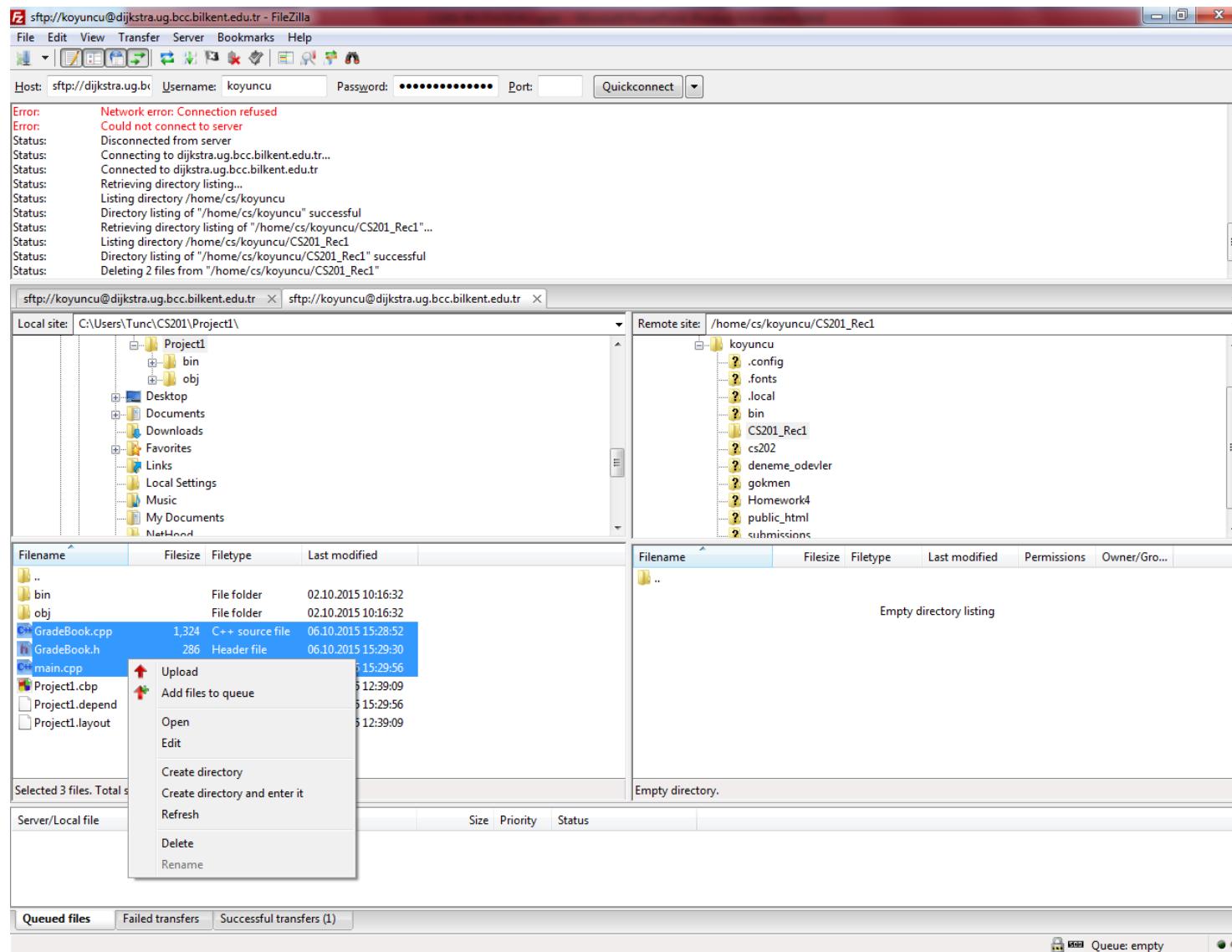
    return 0;
}
```

The bottom panel shows the build log output:

```
----- Run: Debug in Project1 (compiler: GNU GCC Compiler) -----
Checking for existence: C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\Tunc\CS201\Project1\bin\Debug\Project1.exe"  (in C:\Users\Tunc\CS201\Project1\)
Process terminated with status 0 (0 minute(s), 7 second(s))
```

File navigation bar at the bottom: C:\Users\Tunc\CS201\Project1\main.cpp, Windows (CR+LF), default, Line 5, Column 17, Insert, Read/Write, default.

Part 3: Using header files



Part 3: Using header files

```
dijkstra.ug.bcc.bilkent.edu.tr - PuTTY
login as: koyuncu
Using keyboard-interactive authentication.
Password:
Last login: Tue Oct  6 15:12:13 2015 from 139.179.50.40
Have a lot of fun...
koyuncu@dijkstra:~/CS201_Rec1/
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook.cpp  GradeBook.h  main.cpp
koyuncu@dijkstra:~/CS201_Rec1> g++ GradeBook.cpp main.cpp -o GradeBook
koyuncu@dijkstra:~/CS201_Rec1> ls
GradeBook  GradeBook.cpp  GradeBook.h  main.cpp
koyuncu@dijkstra:~/CS201_Rec1> ./GradeBook
Enter the midterm grade : 77
Enter the final grade : 99
Enter the quiz grade : 88
Enter the homework grade : 100
The grades are 77, 99, 88, 100
Overall grade : 90.35
Course created with course name CS 201
Your letter grade : A
koyuncu@dijkstra:~/CS201_Rec1>
```

g++ GradeBook.cpp main.cpp -o GradeBook

