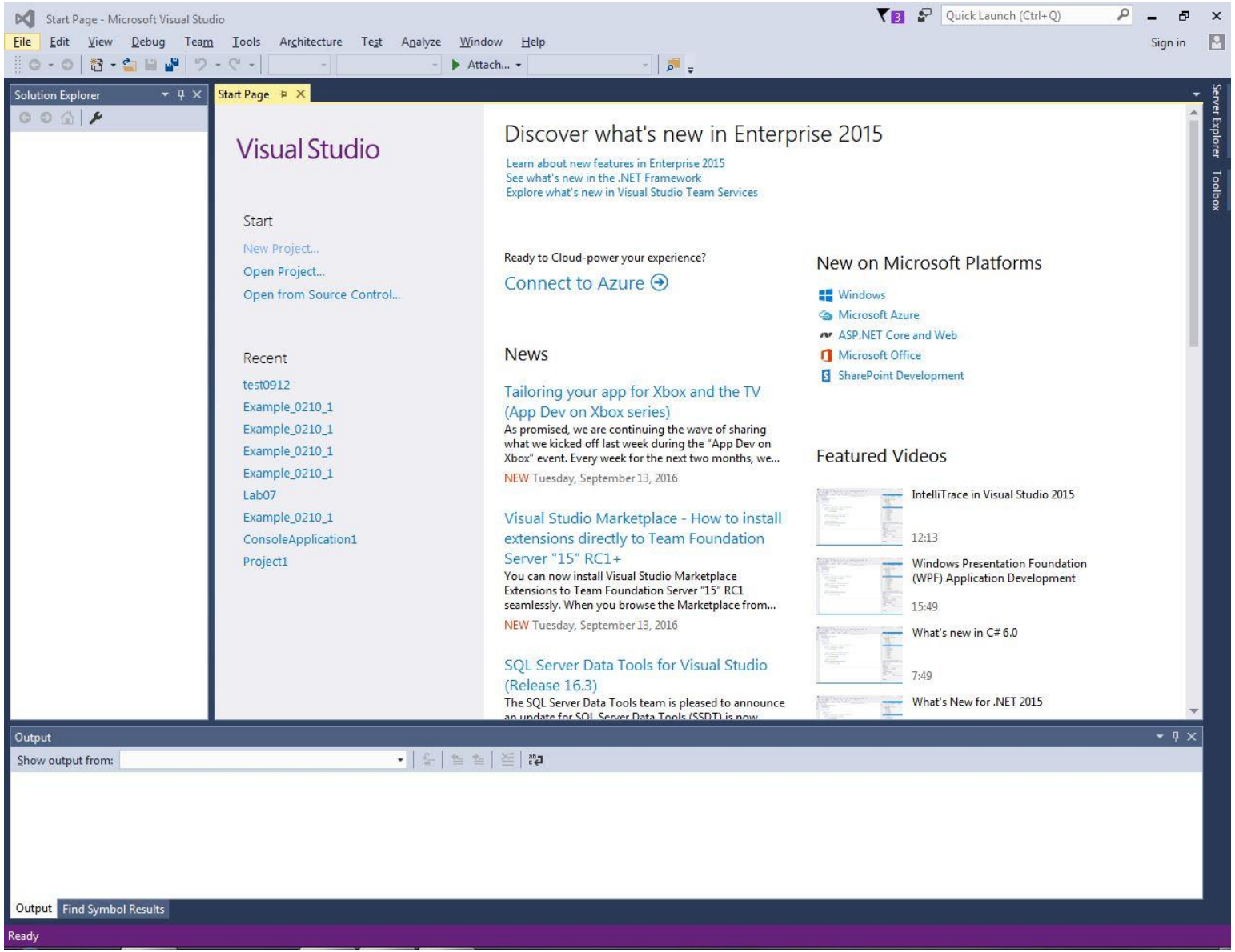
Microsoft Visual Studio Tutorial

Colorado School of Mines Department of Electrical Engineering and Computer Science

Create Your First Program in Microsoft Visual C++

Start Microsoft Visual Studio.

This will bring up the main window as shown below.

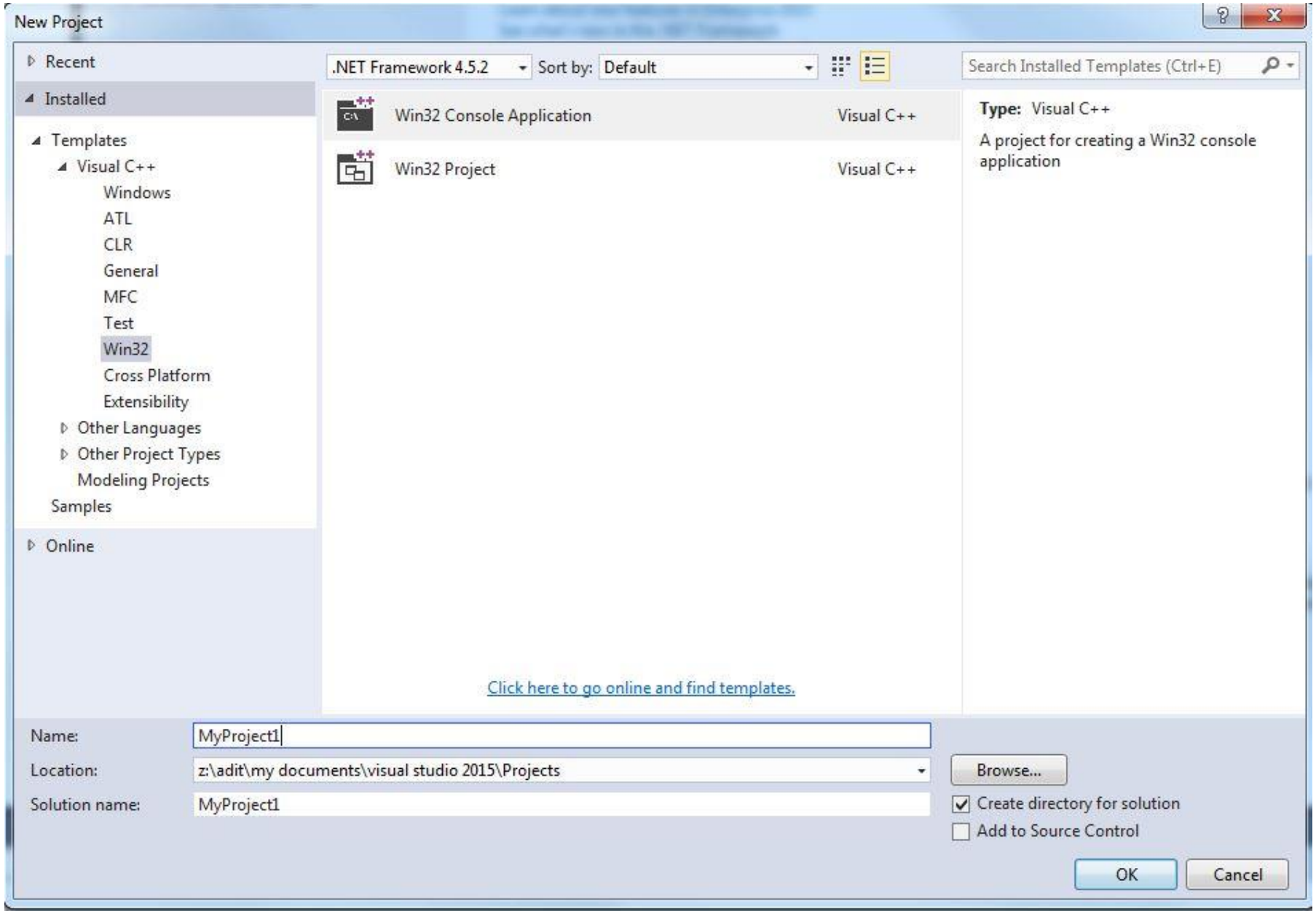


Select the menu item **File -> New -> Project**.

Under the "**Templates**" menu on the left, select "**Visual C++**" and "**Win32**" as shown below. Also select "**Win32 Console Application Visual C++**", which is the simplest type of application.

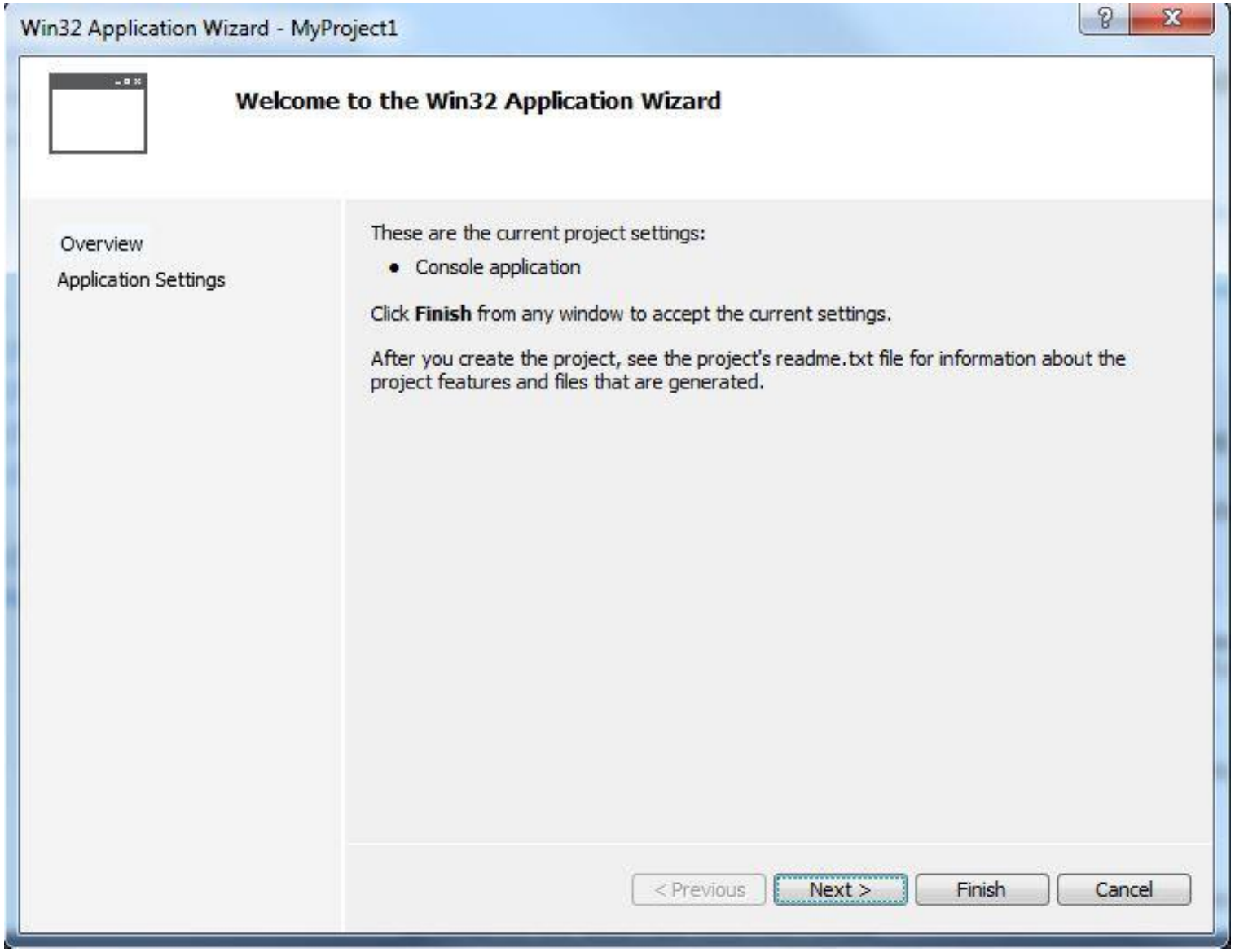
Create Your First Program in Microsoft Visual C++

At the bottom, enter a **location** that you can save the files to, such as your home directory on "adit". Enter a name for the project such as "**myprog1**" (or you can leave it with the default name).



Create Your First Program in Microsoft Visual C++

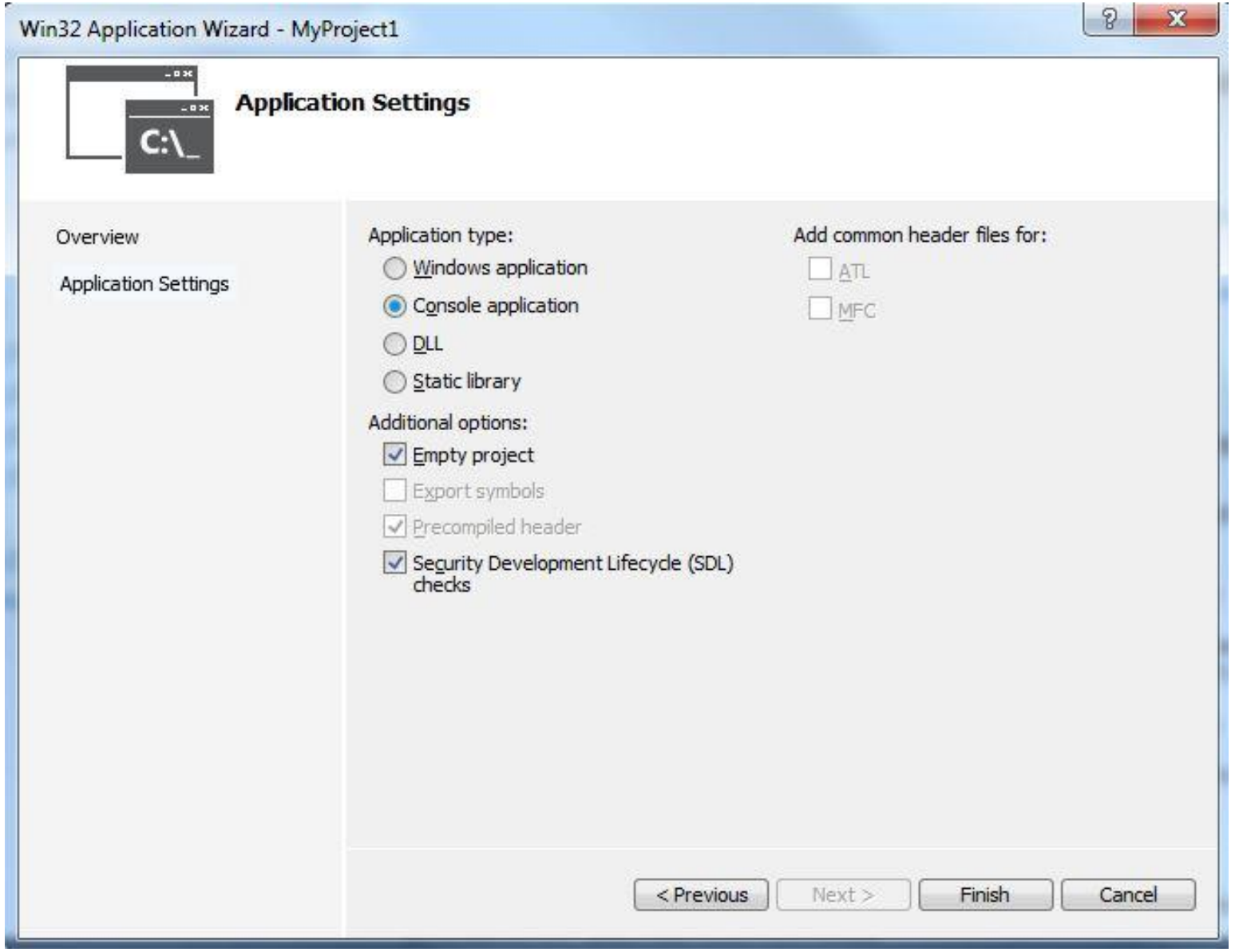
After clicking “**Ok**”, it will bring up the window below:



Click on “**Next**” to go to the next screen:

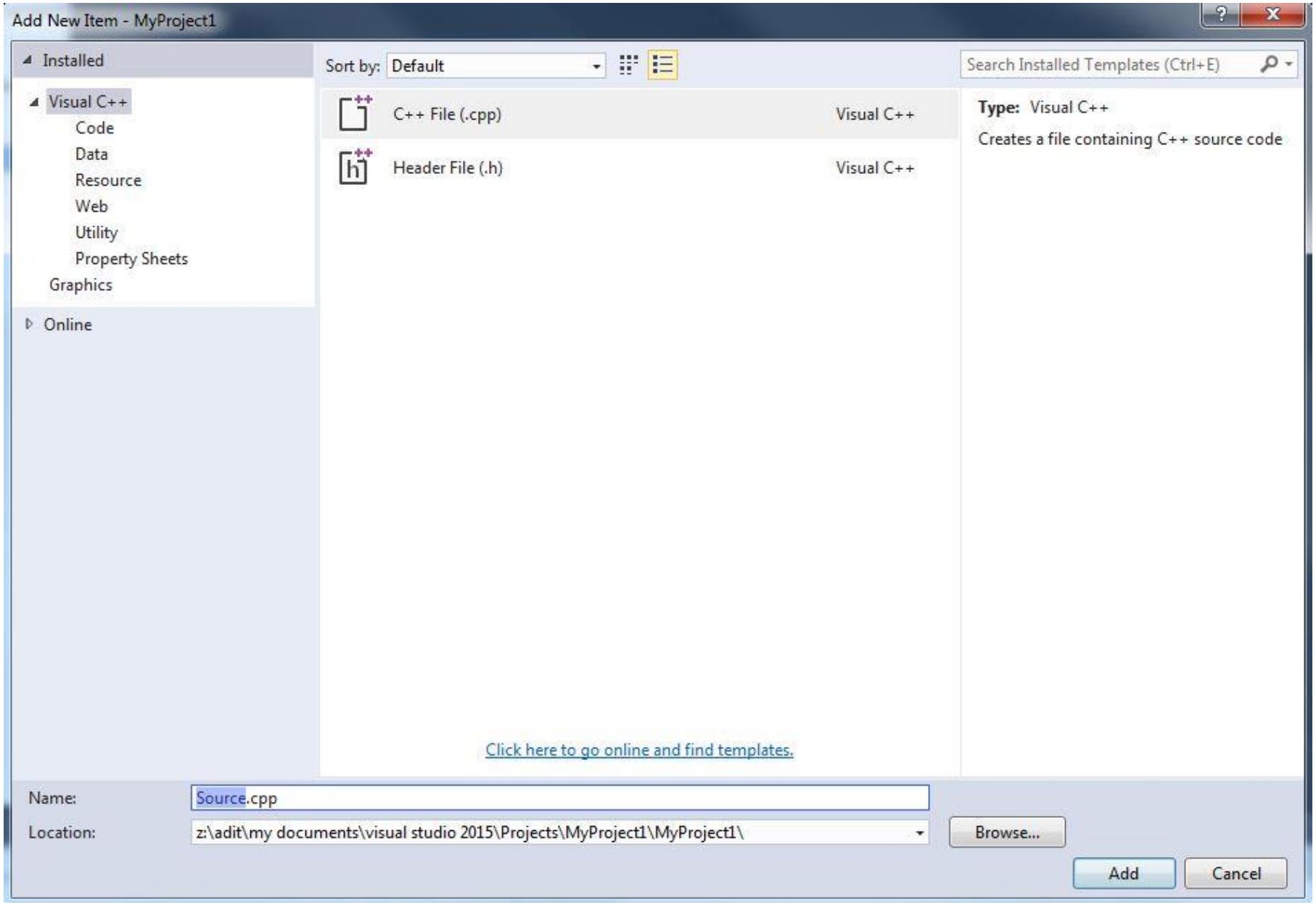
Create Your First Program in Microsoft Visual C++

Select the buttons for “**Console application**” and “**Empty project**”, and click on **Finish.**



Create Your First Program in Microsoft Visual C++

Back on the main window, go to the right box where your project files are displayed, and right click on the folder called "**Source Files**". Choose **Add** -> **New Item** … to get to a pop-up window as shown below.

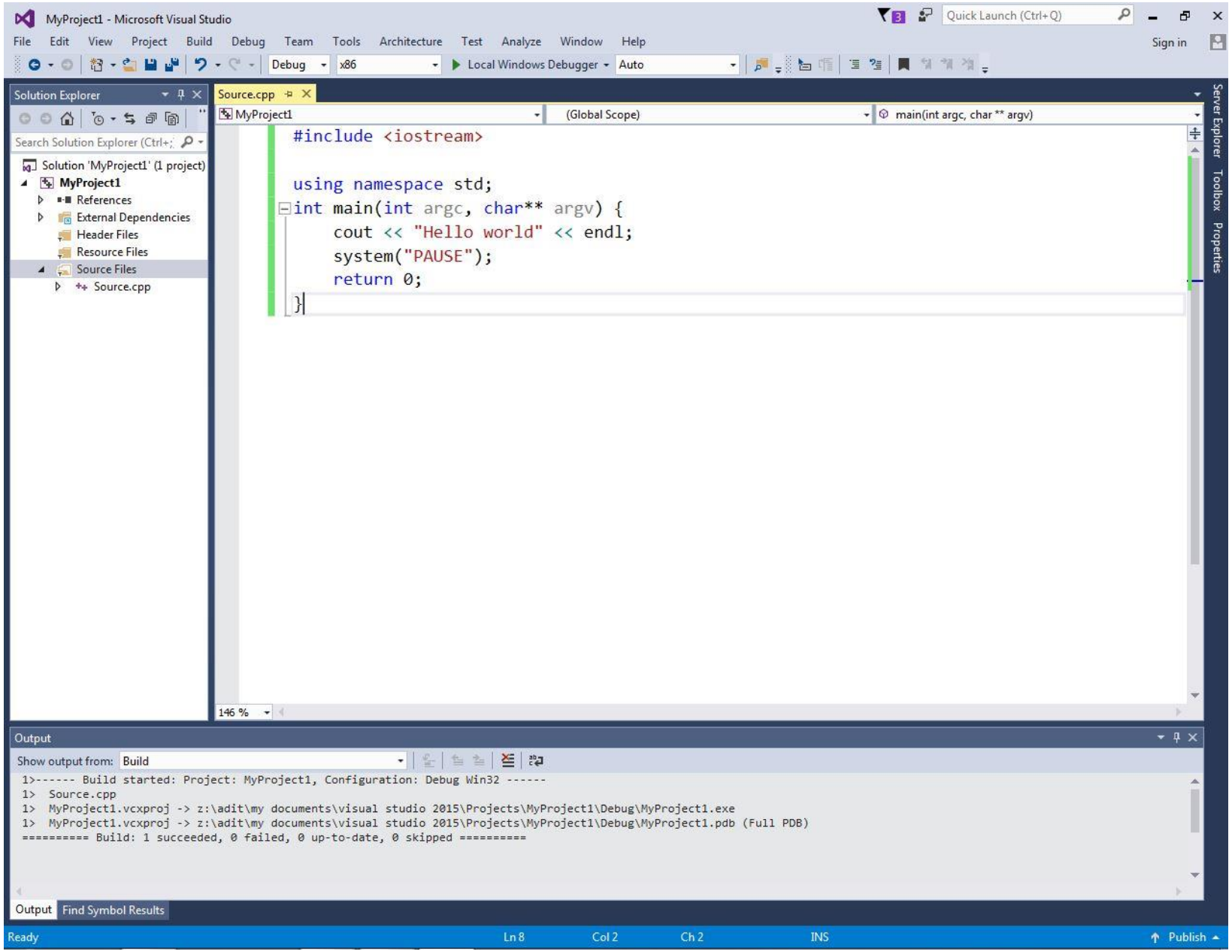


Select type "**C++ File (.cpp)**" and type a name for the file, such as "**Source**". Then click on **Add**.

Create Your First Program in Microsoft Visual C++

This will bring up the editor, where you can type in the code for the program.

Type in the code as shown below:



Then choose **Debug** -> **Start Debugging**. This will compile and run the program.

C++ Programming Tutorial

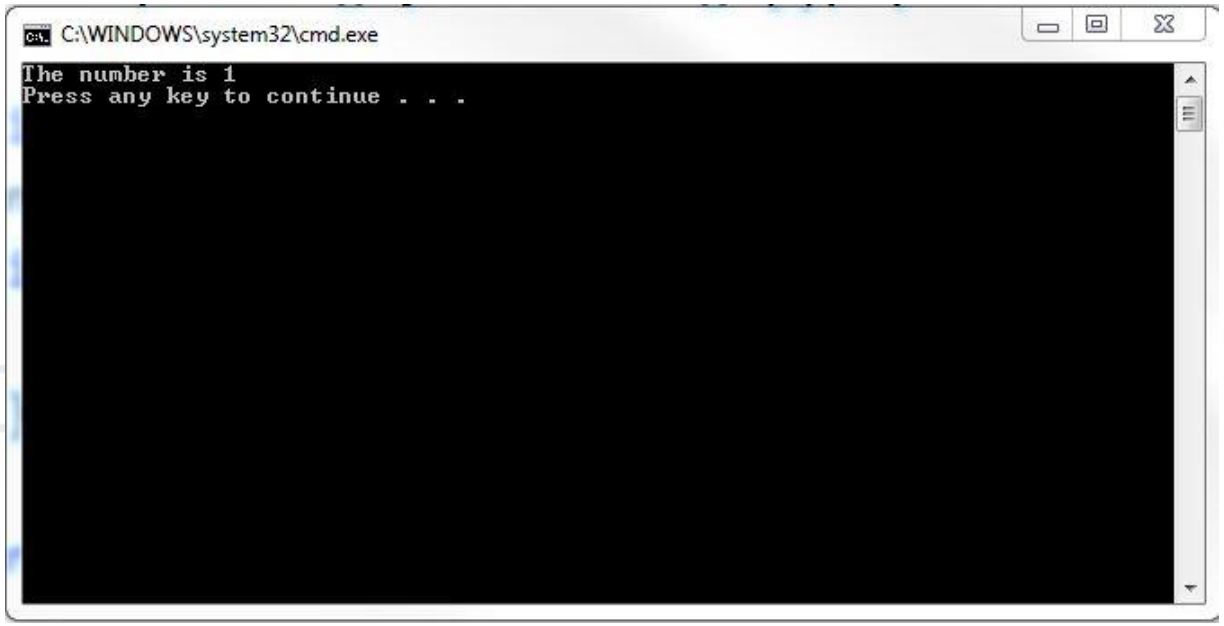
- Control Structures
  - **if** structure – single-selection
  - **if/else** structure – double-selection
    - **switch** – multiple selection structure with breaks
  - **while** structure – repetition structure
    - **do/while** – repetition structure
  - **for** structure – counter-controlled repetition structure

if structure – single selection

Program:

```
#include <iostream>

using namespace std;
int main(int argc, char* argv[]) {

    int number;
    number = 1;
    if (number == 1) {
        cout << "The number is 1" << endl;
    }

    return 0;
}
```

Output:

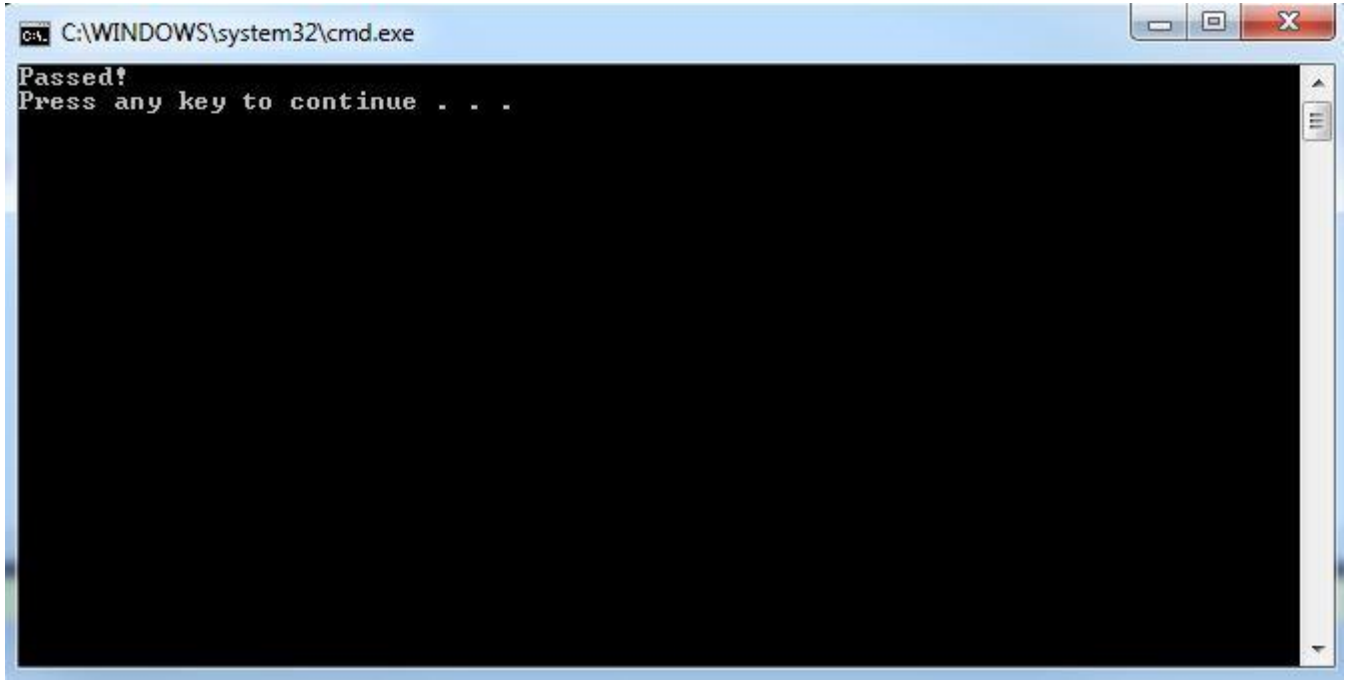


If/else structure – double selection

Program:

```
using namespace std;
int main(int argc, char* argv[]) {

    int grade;
    grade = 72;
    if (grade > 60) {
        cout << "Passed!" << endl;
    }
    else {
        cout << "Failed!" << endl;
    }

    return 0;
}
```

Output:



Program:

```
using namespace std;
int main(int argc, char* argv[]) {

    int month = 8;
    switch (month) {
        case 1:
            cout << "January" << endl;
            break;
        case 2:
            cout << "February" << endl;
            break;
        case 3:
            cout << "March" << endl;
            break;
        case 4:
            cout << "April" << endl;
            break;
        case 5:
            cout << "May" << endl;
            break;
        case 6:
            cout << "June" << endl;
            break;
        case 7:
            cout << "July" << endl;
            break;
        case 8:
            cout << "August" << endl;
            break;
        case 9:
            cout << "September" << endl;
            break;
        case 10:
            cout << "October" << endl;
            break;
        case 11:
            cout << "November" << endl;
            break;
        case 12:
            cout << "December" << endl;
            break;
        default:
            cout << "Invalid month" << endl;
            break;
    }

    return 0;
}
```

switch structure – multiple selection

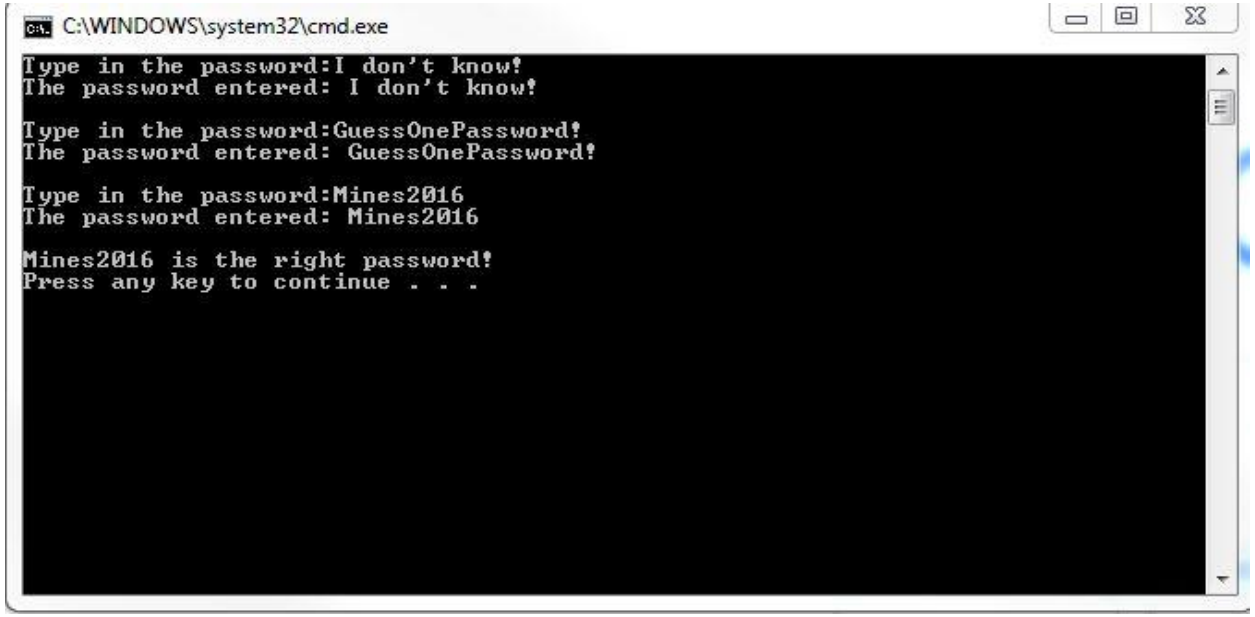
Output:

while structure – repetition structure

Program:

```
#include <iostream>
#include <cstring>
using namespace std;
int main(int argc, char* argv[]) {

    char password[1024];
    while (strcmp(password, "Mines2016")) {
        cout << "\nType in the password:";
        cin.getline(password, 1024);
        cout << "The password entered: " << password << endl;
    }
    cout << "\n" << password << " is the right password!" << endl;


    return 0;
}
```

Output:

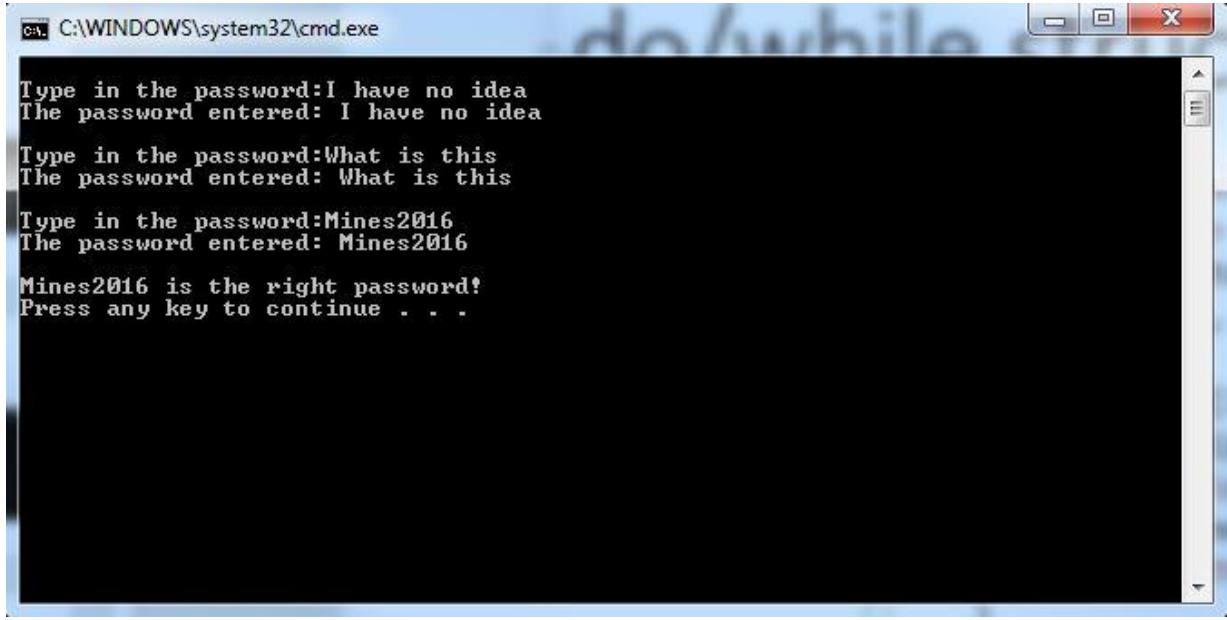


do/while structure – repetition structure

Program:

```
#include <iostream>
#include <cstring>
using namespace std;
int main(int argc, char* argv[]) {

    char password[1024];

    do{
        cout << "\nType in the password:";
        cin.getline(password, 1024);
        cout << "The password entered: " << password << endl;
    } while (strcmp(password, "Mines2016"));
    cout << "\n" << password << " is the right password!" << endl;


    return 0;
}
```

Output:

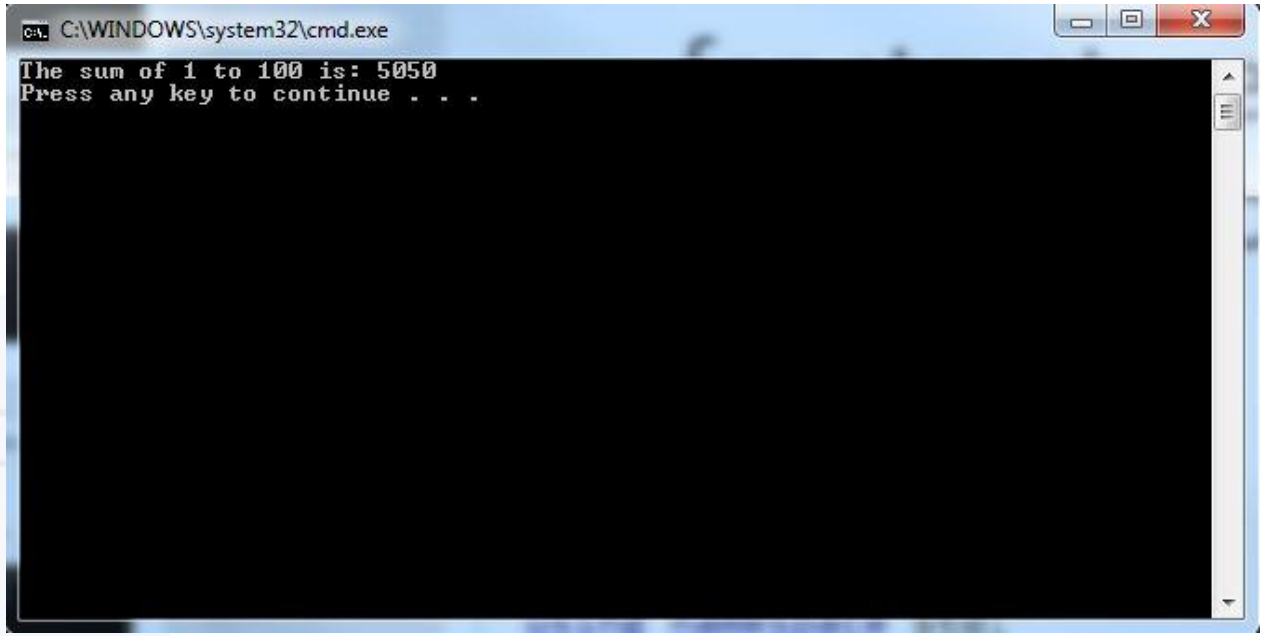


for structure
– counter-controlled repetition structure

Program:

```
#include <iostream>
#include <cstring>
using namespace std;
int main(int argc, char* argv[]) {

    int size = 100;
    int sum = 0;
    for (int i = 1; i <= size; i++) {
        sum += i;
    }
    cout << "The sum of 1 to 100 is: " << sum << endl;

    return 0;
}
```

Output:



C++ Programming Tutorial

- Functions
  - Library Functions
  - User-defined Functions
  - Header Files
  - Recursive Function

Library functions

IO Library -- `#include <iostream>`
Math Library -- `#include <cmath>`
String Library -- `#include <string>`
Standard Library -- `#include <cstdlib>`
File stream Library -- `#include <fstream>`

```
#include <iostream>
        std::cout
        std::cin
        std::endl
```

```
#include <fstream>
        std::ofstream
        std::ifstream
```

```
#include <string>
        std::memcpy
        std::strcpy
        std::strcat
        std::strlen
```

```
#include <cstdlib>
        std::rand
        std::srand
        std::itoa
```

```
#include <cmath>
        std::cos
        std::exp
        std::pow
```

Library functions

Example Program 1:

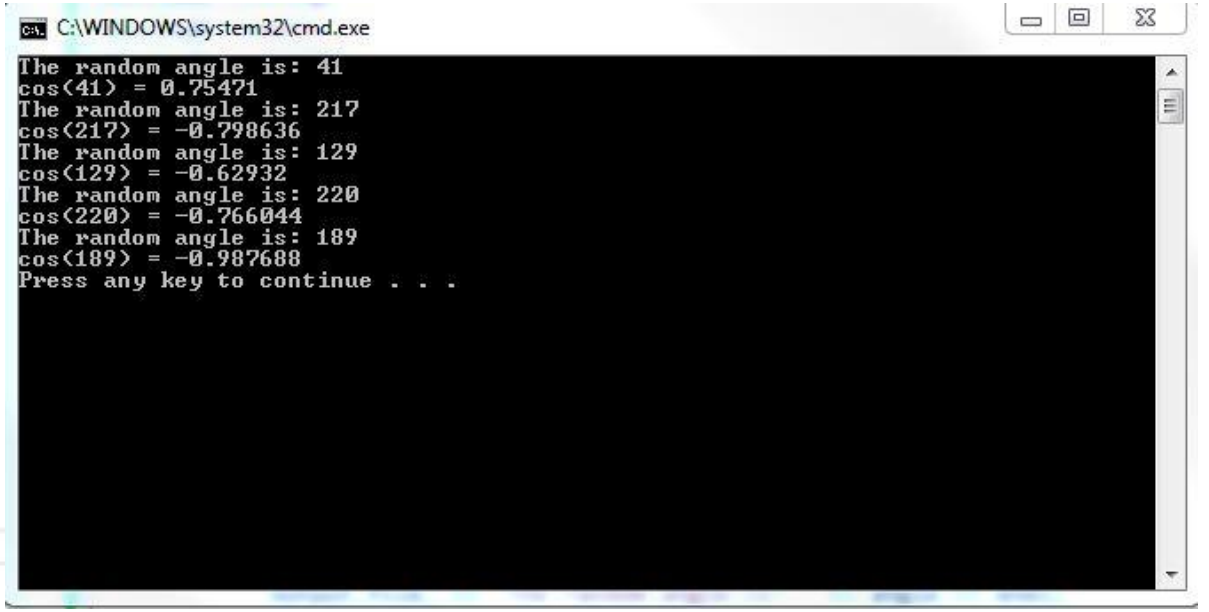
```
// library functions example
#include <iostream>
#include <string>
#include <cmath>
#include <cstdlib>
#include <fstream>

using namespace std;
int main(int argc, char** argv) {

    string apple ("apple");
    string orange;
    string fruit;
    orange = "orange";
    cout << "The first fruit is: " << apple << endl;
    cout << "The second fruit is: " << orange << endl;

    while (fruit.compare(apple) != 0) {
        cout << "Type in my favorite fruit: ";
        getline(cin, fruit);
    }
    cout << "The answer is correct!!!" << endl;

    return 0;
}
```

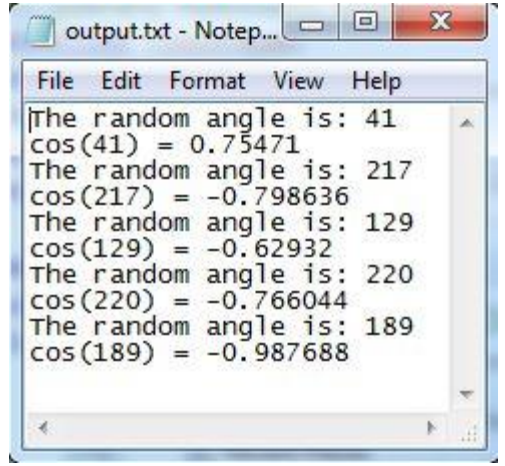
Output:

Library functions

Example Program 2:

```
// library functions example
#include <iostream>
#include <string>
#include <cmath>
#include <cstdlib>
#include <fstream>

#define PI 3.14159265
using namespace std;
int main(int argc, char** argv) {

    double angle;
    ofstream output_file_("output.txt", ios::out);
    if (output_file_.fail()) {
        cout << "The file can't be created!" << endl;
    }
    else {
        for (int i = 0; i < 5; i++) {
            angle = rand() % 365;
            cout << "The random angle is: " << angle << endl;
            output_file_ << "The random angle is: " << angle << endl;
            cout << "cos(" << angle << ") = " << cos(angle / 180 * PI) << endl;
            output_file_ << "cos(" << angle << ") = " << cos(angle / 180 * PI) << endl;
        }
        output_file_.close();
    }

    return 0;
}
```





User-defined functions

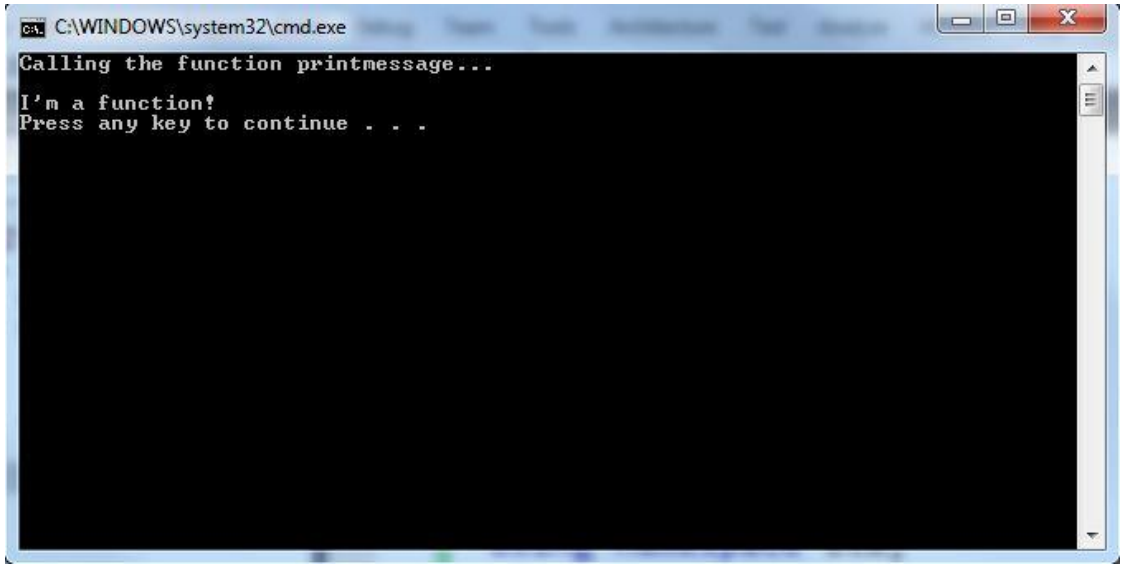
Printmessage.cpp

```cpp
#include <iostream>

using namespace std;
void printmessage(void) {
    cout << "I'm a function!" << endl;
}
```

Source.cpp

```cpp
#include <iostream>

using namespace std;

void printmessage(void);
int main(int argc, char** argv) {

    cout << "Calling the function printmessage...\n" << endl;
    printmessage();

    return 0;
}
```

Output:



Header files

- Put function prototype and variable declaration in header file
- Put function details in cpp file
- Include the header file

Source.cpp

```cpp
#include <iostream>
#include "printmessage.h"
using namespace std;

//void printmessage(void);
int main(int argc, char** argv) {

    cout << "Calling the function printmessage...\n" << endl;
    printmessage();

    return 0;
}
```

printmessage.h

```cpp
//#pragma once

#include <iostream>

void printmessage(void);
```

```cpp
#include <iostream>

using namespace std;

void printmessage(void);
int main(int argc, char** argv) {

    cout << "Calling the function printmessage...\n" << endl;
    printmessage();

    return 0;
}
```

Recursive functions

Source.cpp:

```cpp
#include <iostream>
#include "factorial.h"

using namespace std;
int main(int argc, char** argv) {

    int number = 4;
    cout << "The factorial of " << number
        << " is: " << factorial(number) << endl;

    return 0;
}
```

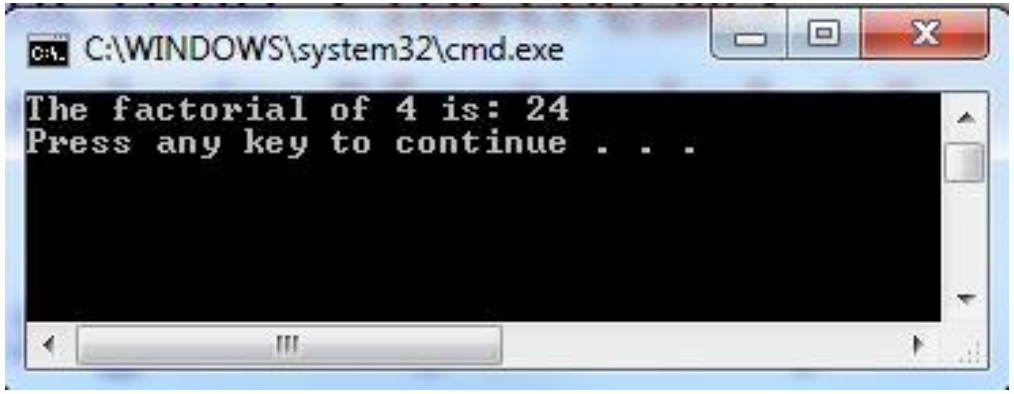
factorial.h:

```cpp
#include <iostream>

unsigned long factorial(unsigned long);
```

factorial.cpp:

```cpp
#include <iostream>

unsigned long factorial(unsigned long number) {
    if (number <= 1) {
        return 1;
    }
    else {
        return number*factorial(number - 1);
    }
}
```

Output:



C++ Programming Tutorial

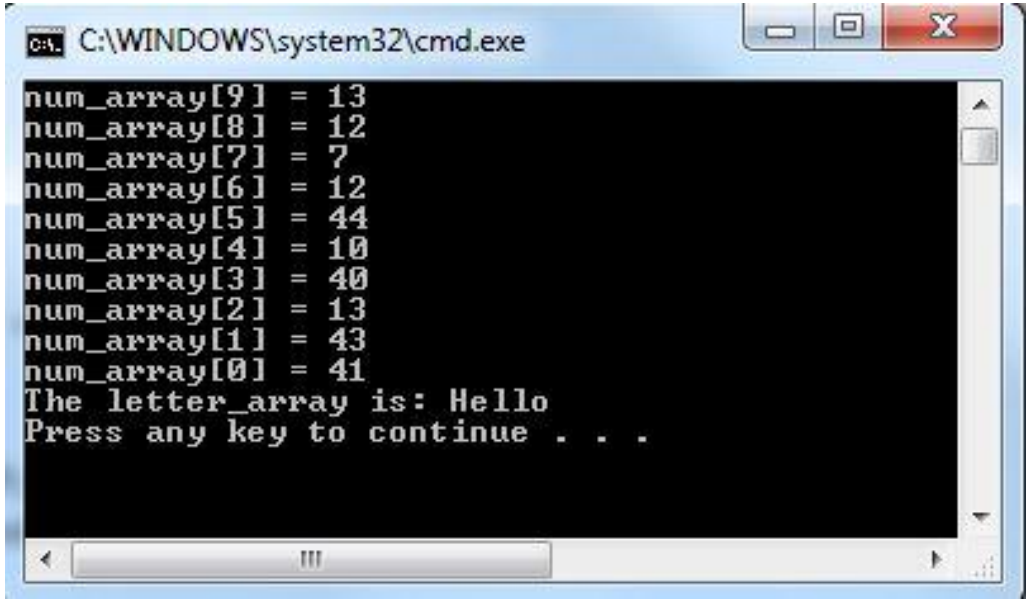
- Arrays, Pointers and Vectors
  - Arrays
  - Pointers
  - vectors

Arrays

Program:

```
// Arrays example
#include <iostream>
#include <cstdlib>

using namespace std;
int main(int argc, char** argv) {
    int num_array[10];

    for (int i = 0; i < 10; i++) {
        num_array[i] = rand() % 49;
    }
    for (int i = 9; i > -1; i--) {
        cout << "num_array[" << i << "] = " << num_array[i] << endl;
    }
    char letter_array[20];
    letter_array[0] = 'H';
    letter_array[1] = 'e';
    letter_array[2] = 'l';
    letter_array[3] = 'l';
    letter_array[4] = 'o';
    letter_array[5] = 0;
    cout << "The letter_array is: " << letter_array << endl;
}
```

Output:

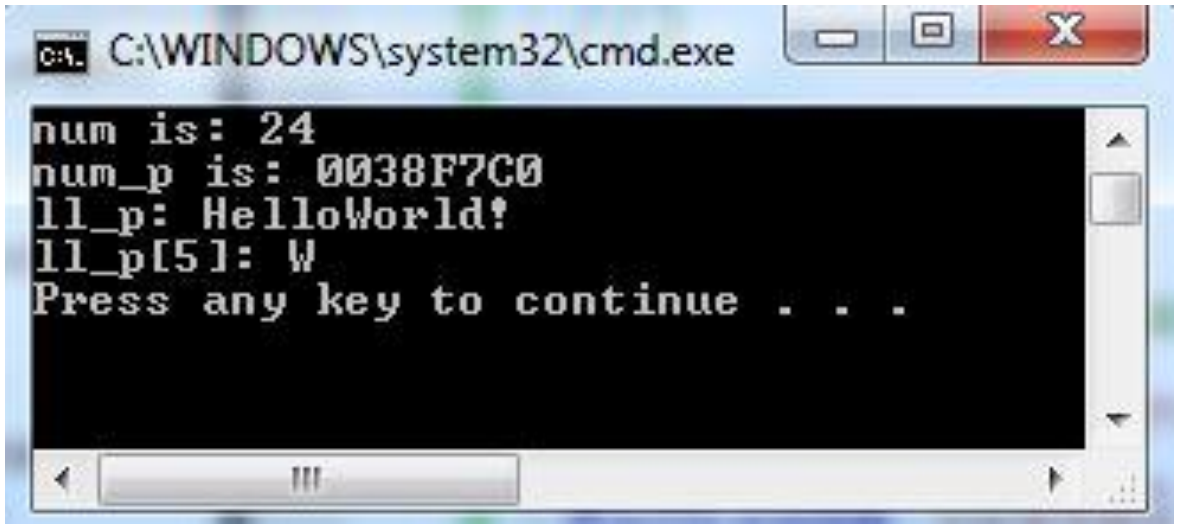


Pointers

Program:

```
// Pointers example
#include <iostream>
#include <cstdlib>

using namespace std;
int main(int argc, char** argv) {
    int num;
    int* num_p = &num;
    *num_p = 24;
    cout << "num is: " << num << endl;
    cout << "num_p is: " << num_p << endl;
    char ll[20] = "HelloWorld!";
    char* ll_p = ll;
    cout << "ll_p: " << ll_p << endl;
    cout << "ll_p[5]: " << ll_p[5] << endl;
}
```

Output:

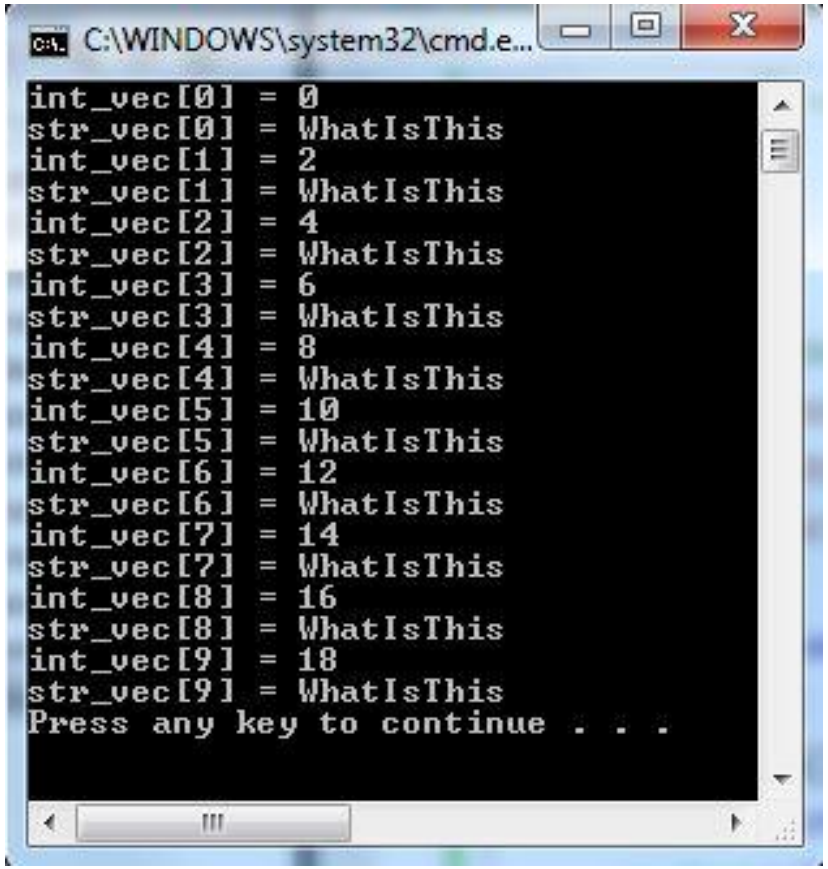


Vectors

Program:

```cpp
// Vectors example
#include <iostream>
#include <cstdlib>
#include <vector>

using namespace std;
int main(int argc, char** argv) {
    char l_a[20] = "WhatIsThis";
    vector<int> int_vec;
    vector<char*> str_vec;
    for (int i = 0; i < 10; i++) {
        int_vec.push_back(i * 2);
        str_vec.push_back(l_a);
    }
    for (int i = 0; i < int_vec.size(); i++) {
        cout << "int_vec[" << i << "] = " << int_vec[i] << endl;
        cout << "str_vec[" << i << "] = " << str_vec[i] << endl;
    }
}
```

Output:



C++ Programming Tutorial

- Classes
  - Define a class
  - Create a object/instance of the class
  - Class in vector
  - Class in header

Define a class

Class definition:

```
class car {
public:
    car(char* name, char* plate);
    char car_plate_[16];    // set in constructor
    unsigned long mileage_; //
    void _set_up(void);     //function
    void _print_report(void);//function

private:
    double weight_;     // _set_up
    int door_num_;      // _set_up
    char color_[64];    // _set_up
    char car_name_[64]; // _init -> _set_name
    void _init(char*);  //function
    void _set_name(char*);//function
};
```

Class functions:

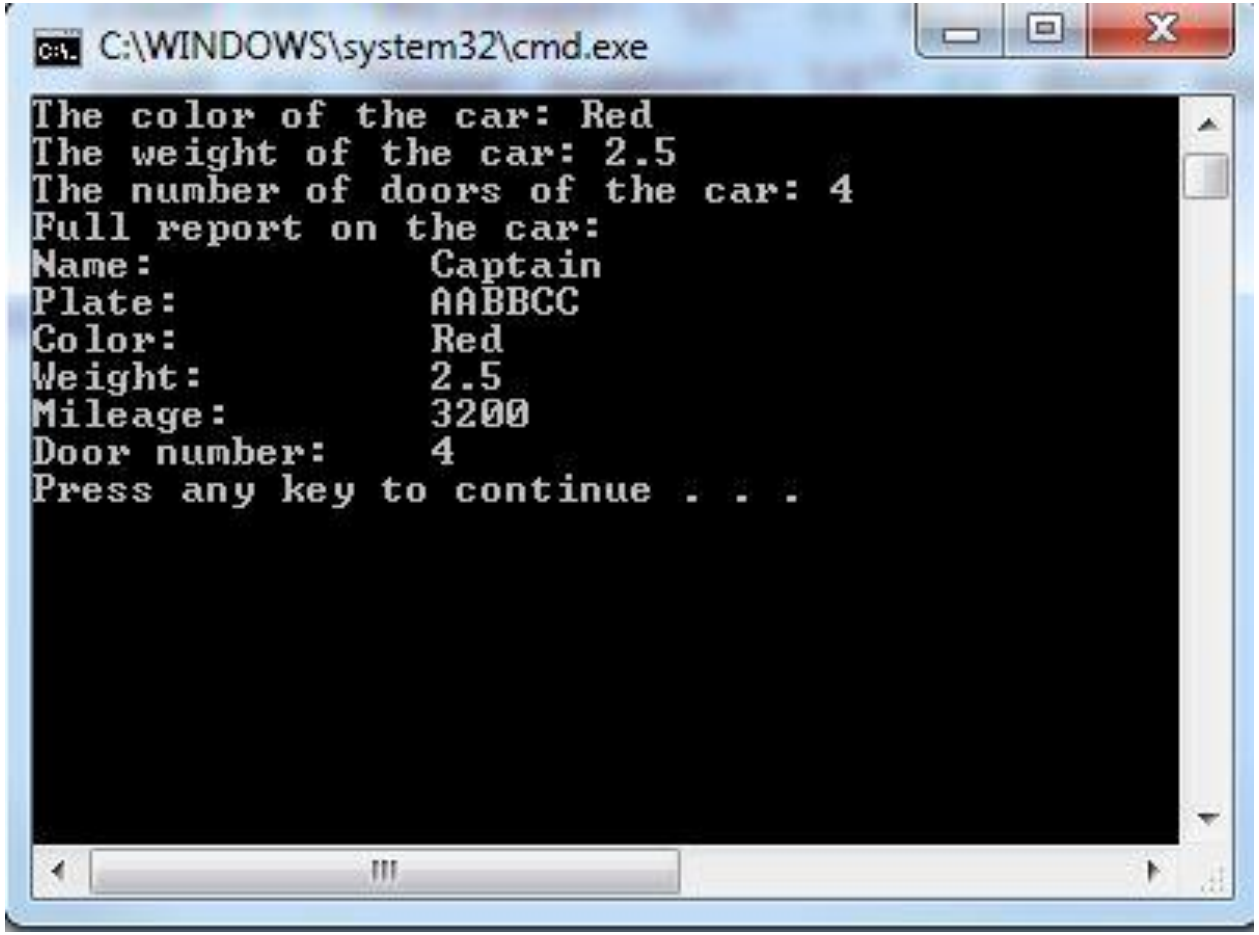
```
car::car(char* name, char* plate) {
    strcpy_s(car_plate_, plate);
    _init(name);
}

void car::_init(char* name) {
    _set_name(name);
}

void car::_set_name(char* name) {
    strcpy_s(car_name_, name);
}

void car::_set_up(void) {
    cout << "The color of the car: ";
    cin >> color_;
    cout << "The weight of the car: ";
    cin >> weight_;
    cout << "The number of doors of the car: ";
    cin >> door_num_;
}

void car::_print_report(void) {
    cout << "Full report on the car: " << endl;
    cout << "Name: \t" << car_name_ << endl;
    cout << "Plate: \t" << car_plate_ << endl;
    cout << "Color: \t" << color_ << endl;
    cout << "Weight: \t" << weight_ << endl;
    cout << "Mileage: \t" << mileage_ << endl;
    cout << "Door number: \t" << door_num_ << endl;
}
```

Create a object/instance of the class

Main function:

```
int main(int argc, char** argv) {

    car Car1("Captain", "AABBCC");
    Car1._set_up();
    Car1.mileage_ = 3200;
    Car1._print_report();

}
```

Output:

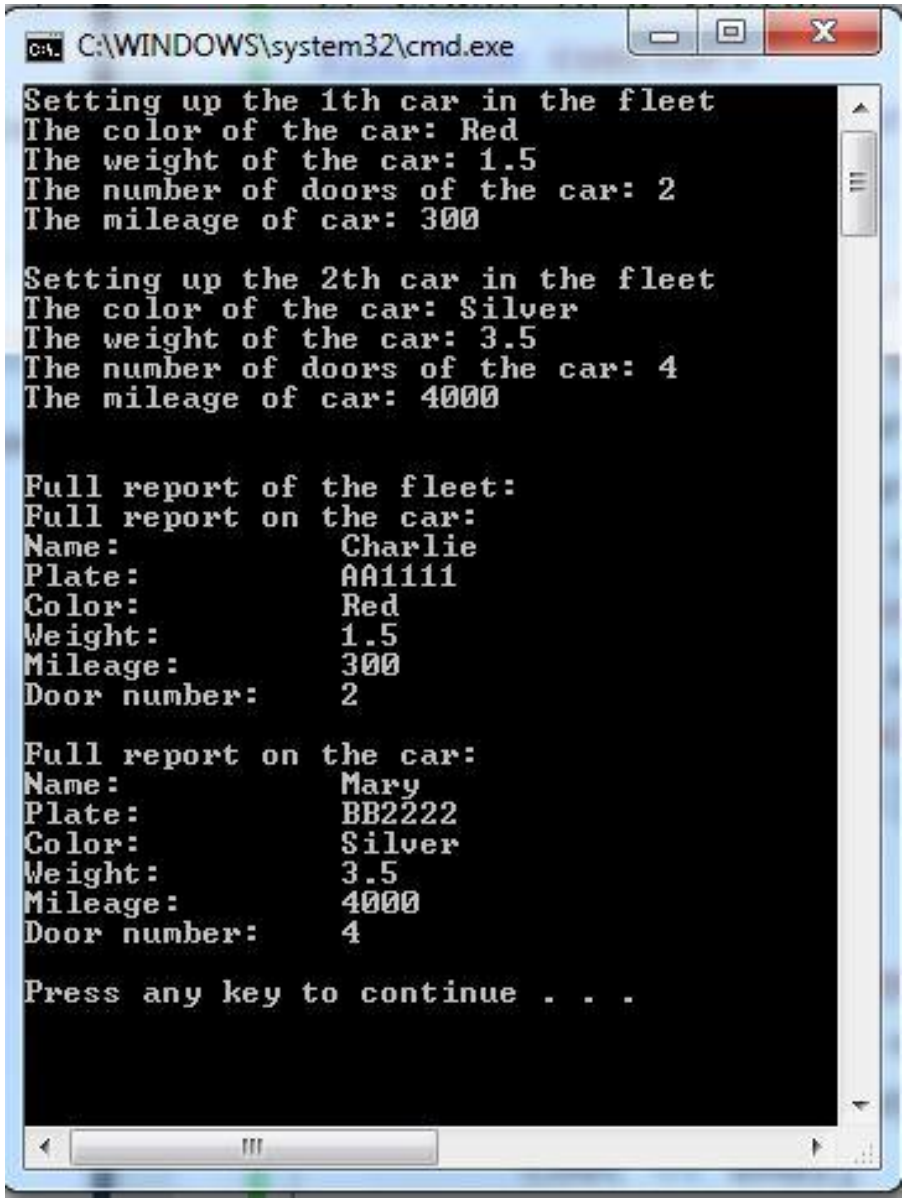


Class in a vector

Main function:

```
// Class in a vector
#include <vector>
int main(int argc, char** argv) {
    vector<car> fleet;

    car Car1_("Charlie", "AA1111");
    car Car2_("Mary","BB2222");
    fleet.push_back(Car1_);
    fleet.push_back(Car2_);
    for (int i = 0; i < fleet.size(); i++) {
        cout << "Setting up the " << i + 1 << "th car in the fleet"<<endl;
        fleet[i]._set_up();
        cout << "The mileage of car: ";
        cin >> fleet[i].mileage_;
        cout << endl;
    }
    cout << "\nFull report of the fleet:" << endl;
    for (int i = 0; i < fleet.size(); i++) {
        fleet[i]._print_report();
        cout << endl;
    }
    return 0;
}
```

Output:



Class in header

Car.h:

```
class car {
public:
    car(char* name, char* plate);
    char car_plate_[16];    // set in constructor
    unsigned long mileage_; //
    void _set_up(void);     //function
    void _print_report(void);//function

private:
    double weight_;     // _set_up
    int door_num_;      // _set_up
    char color_[64];    // _set_up
    char car_name_[64]; // _init -> _set_name
    void _init(char*);  //function
    void _set_name(char*);//function
};
```

Car.cpp:

```
car::car(char* name, char* plate) {
    strcpy_s(car_plate_, plate);
    _init(name);
}

void car::_init(char* name) {
    _set_name(name);
}

void car::_set_name(char* name) {
    strcpy_s(car_name_, name);
}

void car::_set_up(void) {
    cout << "The color of the car: ";
    cin >> color_;
    cout << "The weight of the car: ";
    cin >> weight_;
    cout << "The number of doors of the car: ";
    cin >> door_num_;
}

void car::_print_report(void) {
    cout << "Full report on the car: " << endl;
    cout << "Name: \t" << car_name_ << endl;
    cout << "Plate: \t" << car_plate_ << endl;
    cout << "Color: \t" << color_ << endl;
    cout << "Weight: \t" << weight_ << endl;
    cout << "Mileage: \t" << mileage_ << endl;
    cout << "Door number: \t" << door_num_ << endl;
}
```

- Put class definition in header file
- Put class functions in cpp file
- Include the header file “car.h” in the main cpp file

Class in header

Source.cpp

```cpp
// Class in header example
#include <iostream>
#include <cstring>
#include "Car.h"

int main(int argc, char** argv) {
    car Car1("Captain", "AABBCC");
    Car1._set_up();
    Car1.mileage_ = 3200;
    Car1._print_report();
}
```

Output: