# Microsoft Visual Studio Tutorial

Colorado School of Mines

Department of Electrical Engineering and Computer Science

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.



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).



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

Click on "**Next**" to go to the next screen:

in32 Application Wizard - MyProject1			
Welcom	e to the Win32 Application Wizard		
Overview Application Settings	These are the current project settings: • Console application Click Finish from any window to accept the current settings. After you create the project, see the project's readme.txt file for information about the project features and files that are generated.		
	<pre></pre>		

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



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**.



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:

---

Output:

```
#include <iostream>
```

return 0;

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

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



### 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;</pre>
      }
      else {
_
          cout << "Failed!" << endl;</pre>
      }
```

Output:



return 0;

```
Program:
```

E

```
using namespace std;
⊡int main(int argc, char* argv[]) {
      int month = 8;
      switch (month) {
           case 1:
               cout << "January" << endl;</pre>
               break:
           case 2:
               cout << "February" << endl;</pre>
                break:
           case 3:
                cout << "March" << endl;</pre>
               break;
           case 4:
                cout << "April" << endl;</pre>
               break;
           case 5:
               cout << "May" << endl;</pre>
               break:
           case 6:
                cout << "June" << endl;</pre>
                break;
           case 7:
                cout << "July" << endl;</pre>
               break;
           case 8:
               cout << "August" << endl;</pre>
               break;
           case 9:
               cout << "September" << endl;</pre>
               break;
           case 10:
               cout << "October" << endl;</pre>
               break;
           case 11:
               cout << "November" << endl;</pre>
               break;
           case 12:
               cout << "December" << endl;</pre>
               break:
           default:
                cout << "Invalid month" << endl;</pre>
                break;
```

return 0;

### switch structure - multiple selection



### 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:";</pre>
          cin.getline(password, 1024);
          cout << "The password entered: " << password << endl;</pre>
      }
     cout << "\n" << password << " is the right password!" << endl;</pre>
                                                                                                             _ 0
                                         C:\WINDOWS\system32\cmd.exe
                                         Type in the password:I don't know!
The password entered: I don't know!
     return 0;
                                         Type in the password:GuessOnePassword!
                                         The password entered: GuessOnePassword!
                                         Type in the password:Mines2016
                                         The password entered: Mines2016
                                         Mines2016 is the right password!
                                         Press any key to continue . . .
                    Output:
```

23

Ш

### do/while structure – repetition structure

#### Program:

```
#include <iostream>
#include <cstring>
using namespace std;
pint 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;</pre>
```

return 0;



### for structure – counter-controlled repetition structure

```
Output:
                                                                                                           C:\WINDOWS\system32\cmd.exe
  Program:
                                                         The sum of 1 to 100 is: 5050
                                                         Press any key to continue . . .
F#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++) {</pre>
          sum += i;
      cout << "The sum of 1 to 100 is: " << sum << endl;
      return 0;
```

### 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 <cstdlib>

std::rand
std::srand
std::itoa

#### #include <string>

std::memcpy
std::strcpy
std::strcat
std::strlen

# #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;</pre>
     cout << "The second fruit is: " << orange << endl
     while (fruit.compare(apple) != 0) {
         cout << "Type in my favorite fruit: ";</pre>
         getline(cin, fruit);
     }
     cout << "The answer is correct!!!" << endl;</pre>
     return 0;
```

C:\WINDOWS\system32\cmd.exe		
The first fruit is: apple The second fruit is: orange Type in my favorite fruit: orange Type in my favorite fruit: grape Type in my favorite fruit: apple The answer is correct?!? Press any key to continue	* III	
	-	

### Library functions

cout << "The random angle is: " << angle << endl;</pre>

output file << "The random angle is: " << angle << endl;

cout << "cos(" << angle << ") = " << cos(angle / 180 \* PI) << endl;</pre>

output\_file\_ << "cos(" << angle << ") = " << cos(angle / 180 \* PI) << endl;</pre>

#### Example Program 2:

```
_ 0
 // library functions example
                                                                                                                                                                23
                                                                                    C:\WINDOWS\system32\cmd.exe
#include <iostream>
                                                                                    The random angle is: 41
                                                                                    \cos(41) = 0.75471
 #include <string>
                                                                                    The random angle is: 217
 #include <cmath>
                                                                                    \cos(217) = -0.798636
                                                                                    The random angle is: 129
 #include <cstdlib>
                                                                                    \cos(129) = -0.62932
                                                                                    he random angle is: 220
 #include <fstream>
                                                                                    \cos(220) = -0.766044
                                                                                    The random angle is: 189
                                                                                    \cos(189) = -0.987688
 #define PI 3.14159265
                                                                                    Press any key to continue . . .
 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;</pre>
     }
     else {
         for (int i = 0; i < 5; i++) {
              angle = rand() % 365;
```

File	Edit	Format	View	Help		
The	rand	mang = 0.75	le is: 471	41		*
The	rand	om ang = -0.	le is: 798636	217		
The	rand (129)	om ang = -0.	le is: 62932	129		
The	rand (220)	om ang = -0.	le is: 766044	220		
The cos	r and (189)	$ \begin{array}{r} \text{om ang} \\ = -0. \end{array} $	le is: 987688	189		
						÷
*					ы	

return 0;

output file .close();

### **User-defined functions**

Printmessage.cpp

```
#include <iostream>
            using namespace std;
                                                             C:\WINDOWS\system32\cmd.exe
          void printmessage(void) {
                                                             'm a function!
                cout << "I'm a function!" << endl;</pre>
            }
Source.cpp
          #include <iostream>
          using namespace std;
          void printmessage(void);
         int main(int argc, char** argv) {
               cout << "Calling the function printmessage...\n" << endl;</pre>
               printmessage();
               return 0;
           }
```



### Header files

• Put function prototype and variable declaration in header file

Source.cpp

- Put function details in cpp file
- Include the header file

printmessage.h

//#pragma once

#include <iostream>

```
void printmessage(void);
```

```
F#include <iostream>
  #include "printmessage.h"
  using namespace std:
  //void printmessage(void);

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

      cout << "Calling the function printmessage...\n" << endl;</pre>
      printmessage();
      return 0;
#include <iostream>
using namespace std;
void printmessage(void);
-int main(int argc, char** argv) {
    cout << "Calling the function printmessage...\n" << endl;</pre>
    printmessage();
    return 0;
}
```

### Recursive functions

```
Source.cpp:
```

```
#include "factorial.h"
```

```
using namespace std;
```

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

```
return 0;
```

factorial.cpp:

}

```
#include <iostream>
```

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

factorial.h:

#include <iostream>

unsigned long factorial(unsigned long);



### 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;</pre>
     char letter_array[20];
     letter_array[0] = 'H';
     letter array[1] = 'e';
     letter_array[2] = '1';
     letter array[3] = '1';
     letter_array[4] = 'o';
     letter array[5] = 0;
     cout << "The letter_array is: " << letter_array << endl;</pre>
```

um_arr	ayly] = 13 au[8] = 12	
um_arr	av[7] = 7	1.1
um_arr	ay[6] = 12	
um_arr	•ay[5] = 44	
um_arr	ay[4] = 10	
um_arr	ay[3] = 40	
um_arr	ay[2] = 13	
um_arr	au[0] = 41	
he let	ter arrau is: Hello	
ress a	my key to continue	
	n <b>- </b> na <b>-</b> a successar a la a	
2		-
<	III	

### Pointers

#### Program:

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

```
using namespace std;

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

    int num;

    int* num_p = #

    *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;

}
```



### Vectors

#### Program:

// Vector	rs example
<b>⊟</b> #include	<iostream></iostream>
#include	<cstdlib></cstdlib>
#include	<vector></vector>
using nam	mespace std;
⊡int main	(int argc, char** argv) {
char	<pre>1 a[20] = "WhatIsThis";</pre>
vecto	pr <int> int vec;</int>
vecto	pr <char*> str vec;</char*>
j for (	$(int i = 0; i < 10; i++) \{$
( si	<pre>int_vec.push_back(i * 2);</pre>
1.00	<pre>str_vec.push_back(l_a);</pre>
}	
for (	(int i = 0; i < int_vec.size(); i++) {
	<pre>cout &lt;&lt; "int_vec[" &lt;&lt; i &lt;&lt; "] = " &lt;&lt; int_vec[i] &lt;&lt; endl;</pre>
	<pre>cout &lt;&lt; "str_vec[" &lt;&lt; i &lt;&lt; "] = " &lt;&lt; str_vec[i] &lt;&lt; endl;</pre>
}	
1	

int_vec[0]	)=	Ø	
str_vec[0]	=	WhatIsThis	
int_vec[1]	=	2	E
str_vec[1]	=	WhatIsThis	
int_vec[2]	=	4	
str_vec[2]	=	WhatIsThis	
int_vec[3]	=	6	
str_vec[3]	=	WhatIsThis	
int_vec[4]	=	8	
str_vec[4]	=	WhatIsThis	
int_vec[5]	=	10	
str_vec[5]	=	WhatIsThis	
int_vec[6]	=	12	
str_vec[6]	=	WhatIsThis	
int_vec[7]	=	14	
str_vec[7]	=	WhatIsThis	
int_vec[8]	=	16	
str_vec[8]	Ш	WhatIsThis	
int_vec[9]	-	18	
str_vec[9]	=	WhatIsThis	
Press any	ke	y to continue	

### 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:

```
I 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:

	<pre>double weight_;</pre>	<pre>// _set_up</pre>
	<pre>int door_num_;</pre>	<pre>// _set_up</pre>
	<pre>char color_[64];</pre>	<pre>// _set_up</pre>
	<pre>char car_name_[64];</pre>	<pre>// _init -&gt; _set_name</pre>
	<pre>void _init(char*);</pre>	//function
	<pre>void _set_name(char</pre>	*);//function
};		

```
Class functions: Ecar::car(char* name, char* plate) {
                        strcpy s(car plate , plate);
                        init(name);
                  pvoid car:: init(char* name) {
                        set name(name);
                    }
                  Pvoid car::_set_name(char* name) {
                        strcpy s(car name , name);
                    }
                  □void car:: set up(void) {
                        cout << "The color of the car: ";</pre>
                        cin >> color ;
                        cout << "The weight of the car: ";</pre>
                        cin >> weight ;
                        cout << "The number of doors of the car: ":
                        cin >> door num ;
                  pvoid car::_print_report(void) {
                        cout << "Full report on the car: " << endl;</pre>
                        cout << "Name: \t" << car_name_ << endl;</pre>
                        cout << "Plate: \t" << car_plate_ << endl;</pre>
                        cout << "Color: \t" << color << endl;</pre>
                        cout << "Weight: \t" << weight << endl;</pre>
                        cout << "Mileage: \t" << mileage << endl;</pre>
                        cout << "Door number: \t" << door_num << endl;</pre>
```

### 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();
```



### Class in a vector

Main function:
// Class in a vector
<pre>#include <vector></vector></pre>
<pre>int main(int argc, char** argv) {     vector<car> fleet;</car></pre>
<pre>car Car1_("Charlie", "AA1111"); car Car2 ("Marv" "BB2222");</pre>
fleet nush back(Carl ):
fleet.push_back(Car2);
<pre>for (int i = 0; i &lt; fleet.size(); i++) {     cout &lt;&lt; "Setting up the " &lt;&lt; i + 1 &lt;&lt; "th car in the fleet"&lt;<endl; fleet[i],="" pre="" set="" up();<=""></endl;></pre>
cout << "The mileage of car: ";
<pre>cin &gt;&gt; fleet[i].mileage_; cout &lt;&lt; endl;</pre>
}
<pre>cout &lt;&lt; "\nFull report of the fleet:" &lt;&lt; endl;</pre>
<pre>for (int i = 0; i &lt; fleet.size(); i++) {</pre>
<pre>fleet[i]print_report();</pre>
cout << endl;
}
return 0; }

Setting up the 1th car in the fleet	1
The color of the car: Kea	1
The weight of the car: 1.5	1.22
The number of doors of the car: 2	17
The mileage of car: 300	1
Setting up the 2th car in the fleet	
The color of the car: Silver	
The weight of the car: 3.5	
The number of doors of the car: 4	
The mileage of car: 4000	
Full report of the fleet:	
Full report on the car:	
Name: Charlie	
Plate: AA1111	
Color: Red	
Weight: 1.5	
Mileage: 300	
Door number: 2	
Full report on the car:	
Name: Mary	
Plate: BB2222	
Color: Silver	
Weight: 3.5	
Mileage: 4000	
Door number: 4	
Press any key to continue	
rress any key to continue	
· · · · · · · · · · · · · · · · · · ·	21

### Class in header

Car.h:

```
Car.cpp:
```

```
Eclass 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:

<pre>double weight_;</pre>	<pre>// _set_up</pre>
<pre>int door_num_;</pre>	<pre>// _set_up</pre>
<pre>char color_[64];</pre>	<pre>// _set_up</pre>
<pre>char car_name_[64];</pre>	<pre>// _init -&gt; _set_name</pre>
<pre>void _init(char*);</pre>	//function
<pre>void _set_name(char</pre>	*);//function

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

```
Gcar::car(char* name, char* plate) {
     strcpy_s(car_plate_, plate);
     init(name);
roid 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: ";</pre>
     cin >> color ;
     cout << "The weight of the car: ";</pre>
     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;</pre>
     cout << "Plate: \t" << car_plate_ << endl;</pre>
     cout << "Color: \t" << color_ << endl;</pre>
     cout << "Weight: \t" << weight << endl;</pre>
     cout << "Mileage: \t" << mileage << endl;</pre>
      cout << "Door number: \t" << door_num << endl;</pre>
```

### Class in header

Source.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();
```

