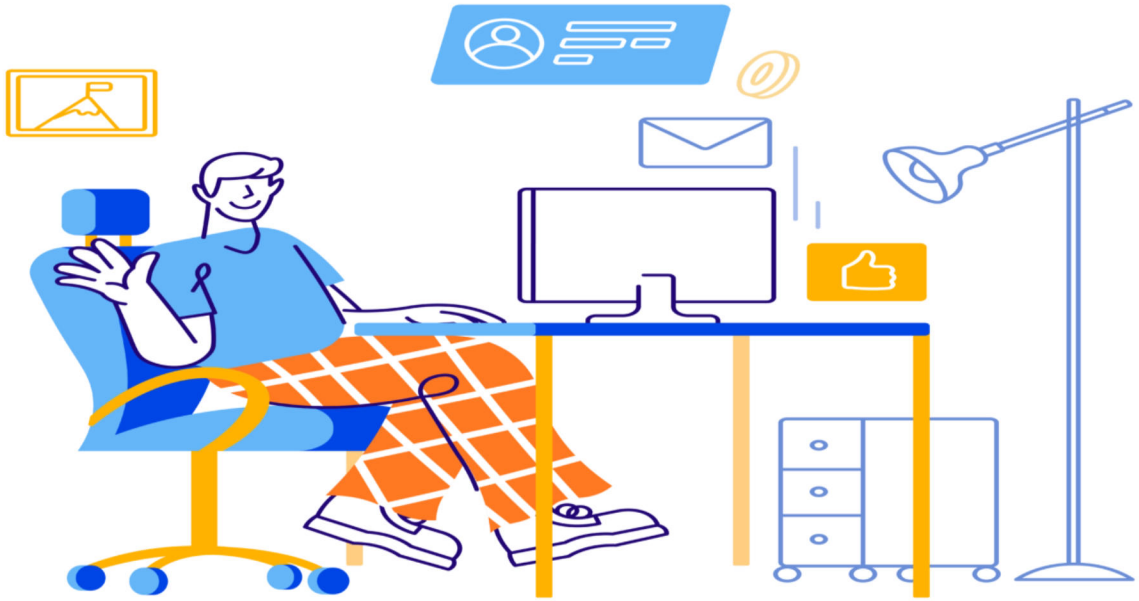
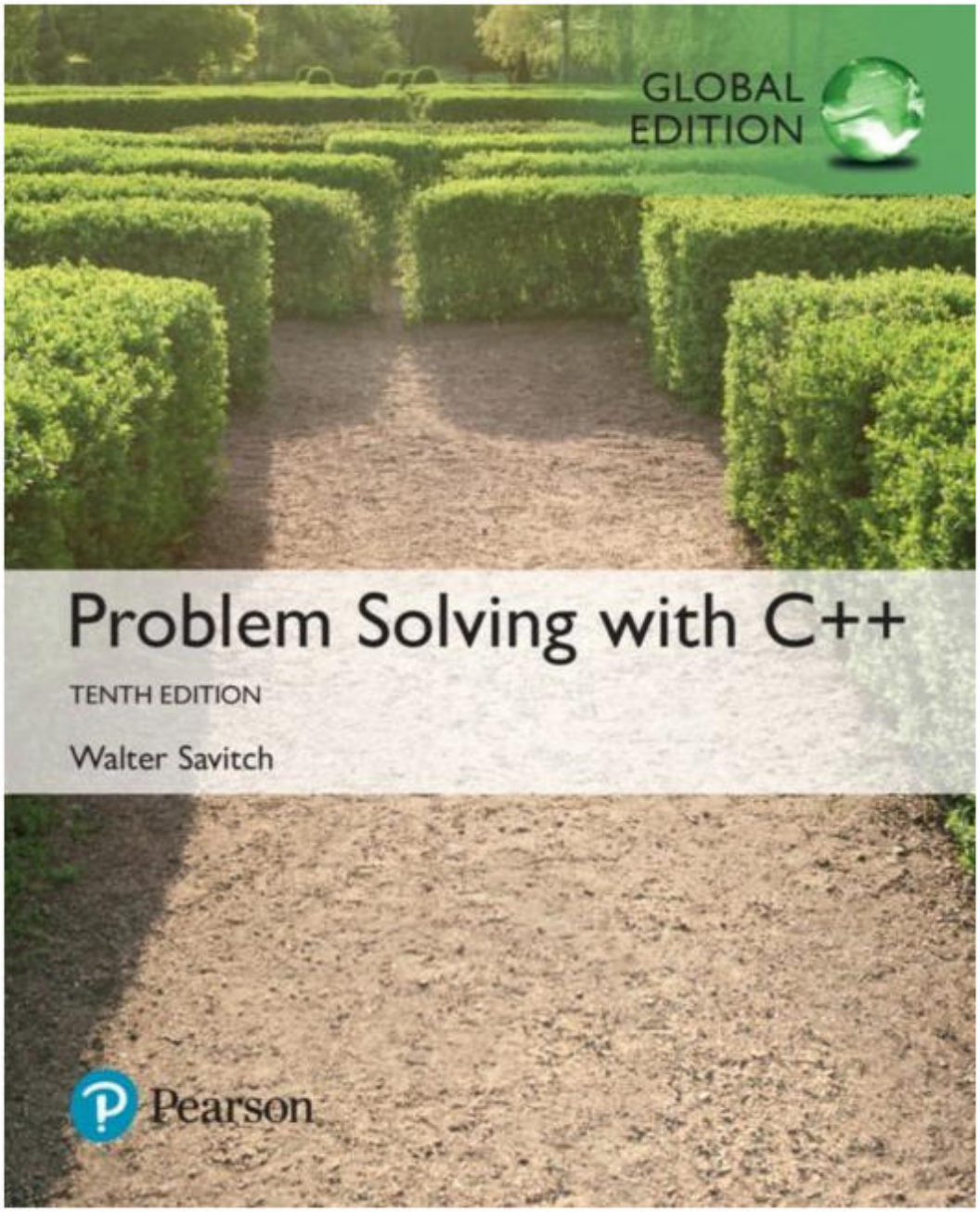
Ahram Canadian University

Fundamental of programming 1



Dr. Marian Wagdy

Lecture 4


GLOBAL
EDITION
Problem Solving with C++
TENTH EDITION
Walter Savitch
Pearson

Chapter 2

2.1 Variables and Assignments

2.2 Input and Output

2.3 Data Types and Expressions

2.4 Simple Flow of Control

2.5 Program Style

2.4

Simple Flow of Control

Simple Flow of Control

- Flow of control
  - The order in which statements are executed
- Branch
  - Lets program choose between two alternatives

Branch Example

- To calculate hourly wages there are two choices
  - Regular time (up to 40 hours)
    - grossPay = rate * hours;
  - Overtime (over 40 hours)
    - grossPay = rate * 40 + 1.5 * rate * (hours - 40);
  - The program must choose which of these expressions to use

Designing the Branch

- Decide if (hours >40) is true
  - If it is true, then use

    grossPay = rate * 40 + 1.5 * rate * (hours - 40);

  - If it is not true, then use

    grossPay = rate * hours;

Implementing the Branch

- if-else statement is used in C++ to perform a branch

```
if (hours > 40)
   grossPay  =  rate * 40 + 1.5 * rate * (hours - 40);
else
   grossPay = rate * hours;
```

Display 2.8

Display 2.9

Boolean Expressions

- Boolean expressions are expressions that are either true or false
- comparison operators such as '>' (greater than) are used to compare variables and/or numbers
  - (hours > 40) Including the parentheses, is the boolean expression from the wages example
  - A few of the comparison operators that use two symbols (No spaces allowed between the symbols!)
    - >= greater than or equal to
    - != not equal or inequality
    - = = equal or equivalent

Display 2.10

if-else Flow Control (1)

- if (boolean expression)
  true statement
  else
  false statement
- When the boolean expression is true
  - Only the true statement is executed
- When the boolean expression is false
  - Only the false statement is executed

if-else Flow Control (2)

- if (boolean expression)
  {
  true statements
  }
  else
  {
  false statements
  }
- When the boolean expression is true
  - Only the true statements enclosed in { } are executed
- When the boolean expression is false
  - Only the false statements enclosed in { } are executed

AND

- Boolean expressions can be combined into more complex expressions with
  - && -- The AND operator
    - True if both expressions are true
- Syntax: (Comparison_1) && (Comparison_2)
- Example: if ((2 < x) && (x < 7))
  - True only if x is between 2 and 7
  - Inside parentheses are optional but enhance meaning

OR

- | | -- The OR operator (no space!)
  - True if either or both expressions are true
- Syntax: (Comparison_1) | | (Comparison_2)
- Example: if ((x = = 1) | | (x = = y))
  - True if x contains 1
  - True if x contains the same value as y
  - True if both comparisons are true

NOT

- ! -- negates any boolean expression
  - !(x < y)
    - True if x is NOT less than y
  - !(x = = y)
    - True if x is NOT equal to y
- ! Operator can make expressions difficult to understand…use only when appropriate

Inequalities

- Be careful translating inequalities to C++
- if x < y < z translates as

```
if ( ( x < y )  && ( y < z ) )
```

NOT

```
if ( x < y < z )
```

Pitfall: Using = or ==

- ' = ' is the assignment operator
  - Used to assign values to variables
  - Example: x = 3;
- '= = ' is the equality operator
  - Used to compare values
  - Example: if (x == 3)
- The compiler will accept this error:

  if (x = 3)

  but stores 3 in x instead of comparing x and 3
  - Since the result is 3 (non-zero), the expression is true

Compound Statements

- A compound statement is more than one statement enclosed in { }
- Branches of if-else statements often need to execute more that one statement

- Example:

```
if (boolean expression)
 {
      true statements
 }
else
 {
     false statements
 }
```

Display 2.11

Branches Conclusion

- Can you
  - Write an if-else statement that outputs the word High if the value of the variable score is greater than 100 and Low if the value of score is at most 100? The variables are of type int.
  - Write an if-else statement that outputs the word Warning provided that either the value of the variable temperature is greater than or equal to 100, or the of the variable pressure is greater than or equal to 200, or both. Otherwise, the if_else sttement outputs the word OK. The variables are of type int.

Simple Loops

- When an action must be repeated, a loop is used
- C++ includes several ways to create loops
- We start with the while-loop

- Example:

```
while (countDown > 0)
  {
      cout << "Hello ";
       countDown  -= 1;
  }
```

- Output: Hello Hello Hello
  when countDown starts at 3

Display 2.12

While Loop Operation

- First, the boolean expression is evaluated
  - If false, the program skips to the line following the while loop
  - If true, the body of the loop is executed
    - During execution, some item from the boolean expression is changed
  - After executing the loop body, the boolean expression is checked again repeating the process until the expression becomes false

- A while loop might not execute at all if the boolean expression is false on the first check

while Loop Syntax

- while (boolean expression is true)

```
{
        statements to repeat
}
```

  - Semi-colons are used only to end the statements within the loop

- while (boolean expression is true)

```
        statement to repeat
```

Display 2.13

do-while loop

- A variation of the while loop.
- A do-while loop is always executed at least once
  - The body of the loop is first executed
  - The boolean expression is checked after the body has been executed

Display 2.14

Display 2.15

- Syntax:

```
do
{
    statements to repeat

} while (boolean_expression);
```

Increment/Decrement

- Unary operators require only one operand
  - + in front of a number such as +5
  - - in front of a number such as -5
- ++ increment operator
  - Adds 1 to the value of a variable

    x ++;

    is equivalent to x = x + 1;
- -- decrement operator
  - Subtracts 1 from the value of a variable

    x --;

    is equivalent to x = x – 1;

Sample Program

- Bank charge card balance of $50
- 2% per month interest
- How many months without payments before your balance exceeds $100
- After 1 month: $50 + 2% of $50 = $51
- After 2 months: $51 + 2% of $51 = $52.02
- After 3 months: $52.02 + 2% of $52.02 …

Display 2.16

Lec2.cpp | Untitled-1

Lec2.cpp > main()

```cpp
#include <iostream>
using namespace std;

int main()
{
    float balance = 0;

    cout << "Balance = ";
    cin >> balance;
    for (int i = 1; i <= 12; i++)
    {
        cout << "After " << i << " Month:\n";
        balance += (balance * 0.02);
        cout << "               Balance = " << balance;
        cout << "\n";
    }

    return 0;
}
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | Code

```
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
emester 2\Fundamentals of Programming 1\" ; if ($?) { g++ Lec2.
 ; if ($?) { .\Lec2 }
Balance = 50
After 1 Month:
               Balance = 51
After 2 Month:
               Balance = 52.02
After 3 Month:
               Balance = 53.0604
After 4 Month:
               Balance = 54.1216
After 5 Month:
               Balance = 55.204
After 6 Month:
               Balance = 56.3081
After 7 Month:
               Balance = 57.4343
After 8 Month:
               Balance = 58.583
After 9 Month:
               Balance = 59.7546
After 10 Month:
               Balance = 60.9497
After 11 Month:
               Balance = 62.1687
After 12 Month:
               Balance = 63.4121
PS E:\Education\Semester 2\Fundamentals of Programming 1>
```

Infinite Loops

- Loops that never stop are infinite loops
- The loop body should contain a line that will eventually cause the boolean expression to become false
- Example: Print the odd numbers less than 12

```
x = 1;
while (x != 12)
    {
       cout << x << endl;
       x = x + 2;
     }
```

- Better to use this comparison: while (x < 12)

Section 2.4 Conclusion

- Can you
  - Show the output of this code if x is of type int?

```
x = 10;
while ( x > 0)
  {
      cout << x << endl;
      x = x – 3;
   }
```

  - Show the output of the previous code using the comparison x < 0 instead of x > 0?

2.5

Program Style

Program Style

- A program written with attention to style
  - is easier to read
  - easier to correct
  - easier to change

Program Style - Indenting

- Items considered a group should look like a group
  - Skip lines between logical groups of statements
  - Indent statements within statements

```
if (x = = 0)
    statement;
```

- Braces {} create groups
  - Indent within braces to make the group clear
  - Braces placed on separate lines are easier to locate

Program Style - Comments

- // is the symbol for a single line comment
  - Comments are explanatory notes for the programmer
  - All text on the line following // is ignored by the compiler
  - Example: //calculate regular wages
    grossPay = rate * hours;
- /* and */ enclose multiple line comments
  - Example: /* This is a comment that spans
    multiple lines without a
    comment symbol on the middle line
    */

Program Style - Constants

- Number constants have no mnemonic value
- Number constants used throughout a program are difficult to find and change when needed
- Constants
  - Allow us to name number constants so they have meaning
  - Allow us to change all occurrences simply by changing the value of the constant

Constants

- const is the keyword to declare a constant
- Example:

```
const int WINDOW_COUNT = 10;
```

declares a constant named WINDOW_COUNT

  - Its value cannot be changed by the program like a variable
  - It is common to name constants with all capitals

Display 2.17

Section 2.5 Conclusion

- Can you
  - Create a named constant of type double?
  - Determine if a program can modify the value of a constant?
  - Describe the benefits of comments?
  - Explain why indenting is important in a program?
  - Explain why blank lines are important in a program?

Display 2.8

DISPLAY 2.8 An if-else Statement

```
1    #include <iostream>
2    using namespace  std;
3    int main( )
4    {
5        int hours;
6        double grossPay, rate;
7        cout << "Enter the hourly rate of pay: $";
8        cin >> rate;
9        cout << "Enter the number of hours worked,\n"
10            << "rounded to a whole number of hours: ";
11       cin >> hours;
12       if (hours > 40)
13          grossPay = rate * 40 + 1.5 * rate * (hours - 40);
14       else
15          grossPay = rate * hours;

16       cout.setf(ios::fixed);
17       cout.setf(ios::showpoint);
18       cout.precision(2);
19       cout << "Hours = " << hours << endl;
20       cout << "Hourly pay rate = $" << rate << endl;
21       cout << "Gross pay = $" << grossPay << endl;
22       return  0;
23   }
```

Sample Dialogue 1

```
Enter the hourly rate of pay: $20.00
Enter the number of hours worked,
rounded to a whole number of hours: 30
Hours = 30
Hourly pay rate = $20.00
Gross pay = $600.00
```

Sample Dialogue 2

```
Enter the hourly rate of pay: $10.00
Enter the number of hours worked,
rounded to a whole number of hours: 41
Hours = 41
Hourly pay rate = $10.00
Gross pay = $415.00
```

Display 2.9

Syntax for an *if-else* Statement

A Single Statement for Each Alternative:

```
if (Boolean_Expression)
    Yes_Statement
else
    No_Statement
```

A Sequence of Statements for Each Alternative:

```
if (Boolean_Expression)
{
    Yes_Statement_1
    Yes_Statement_2
      . . .
    Yes_Statement_Last
}
else
{
    No_Statement_1
    No_Statement_2
      . . .
    No_Statement_Last
}
```

Display 2.10

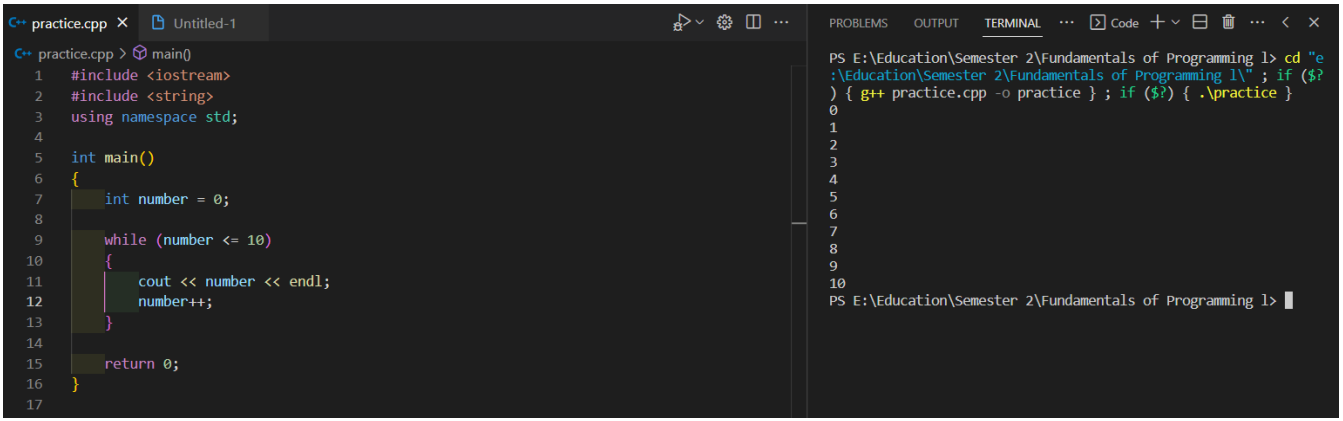
Comparison Operators

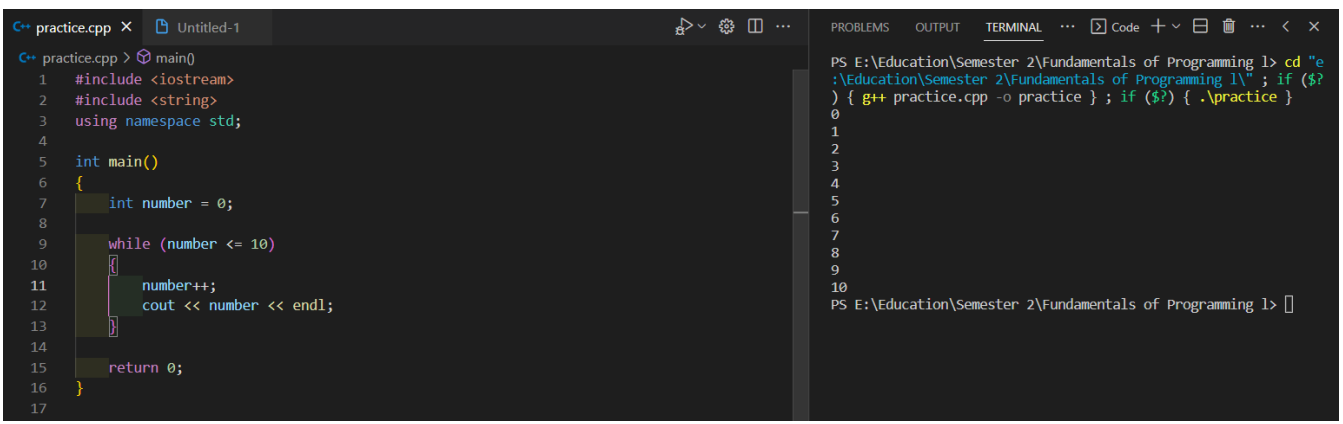
Math Symbol	English	C++ Notation	C++ Sample	Math Equivalent
=	equal to	==	`x + 7 == 2*y`	$x + 7 = 2y$
≠	not equal to	!=	`ans != 'n'`	ans ≠ 'n'
<	less than	<	`count < m + 3`	count < m + 3
≤	less than or equal to	<=	`time <= limit`	time ≤ limit
>	greater than	>	`time > limit`	time > limit
≥	greater than or equal to	>=	`age >= 21`	age ≥ 21

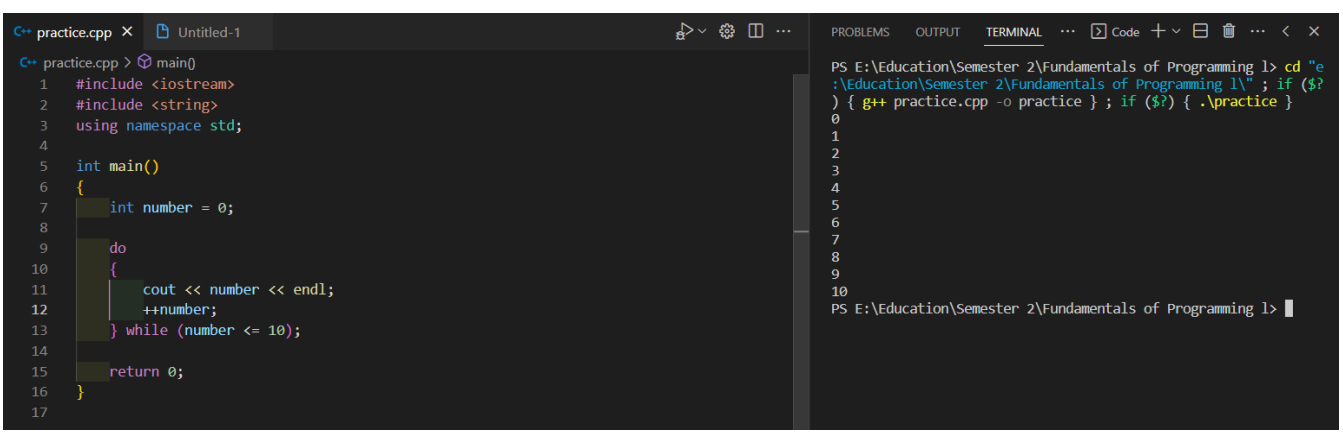
Display 2.11

Compound Statements Used with *if-else*

```
if (my_score > your_score)
{
    cout << "I win!\n";
    wager = wager + 100;
}
else
{
    cout << "I wish these were golf scores.\n";
    wager = 0;
}
```







Do not forget to increase the variable used in the condition, otherwise the loop will never end!

Nested Loops

practice.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    int number = 0;

    for (int i = 1; i <= 2; ++i)
    {
        cout << "Outer: " << i << "\n";
        for (int j = 1; j <= 3; ++j)
        {
            cout << " Inner: " << j << "\n";
        }
    }

    return 0;
}
```

```
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?
) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
Outer: 1
 Inner: 1
 Inner: 2
 Inner: 3
Outer: 2
 Inner: 1
 Inner: 2
 Inner: 3
PS E:\Education\Semester 2\Fundamentals of Programming 1>
```

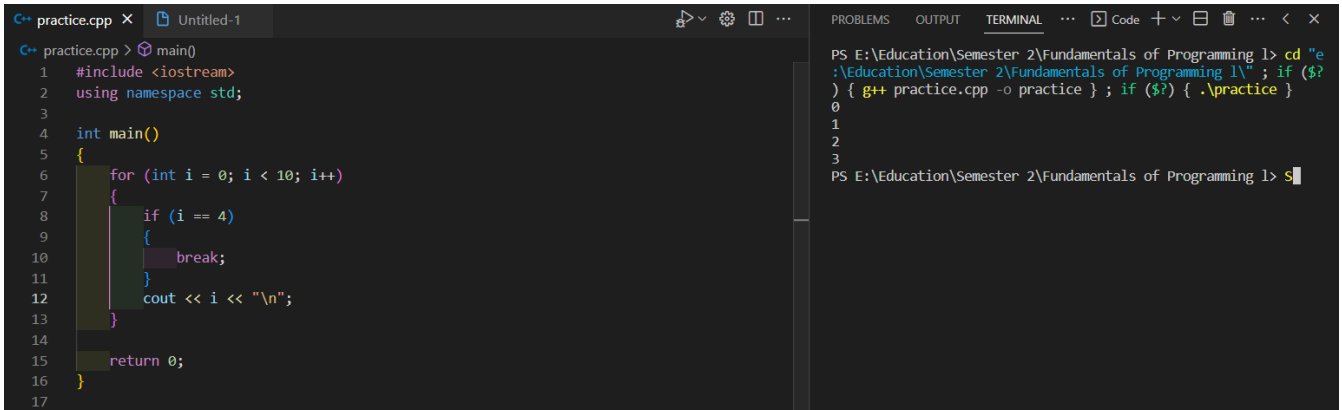
for-each loop in an array

practice.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    // for-each loop in an array
    int myNumbers[5] = {10, 20, 30, 40, 50};
    for (int i : myNumbers)
    {
        cout << i << "\n";
    }

    return 0;
}
```

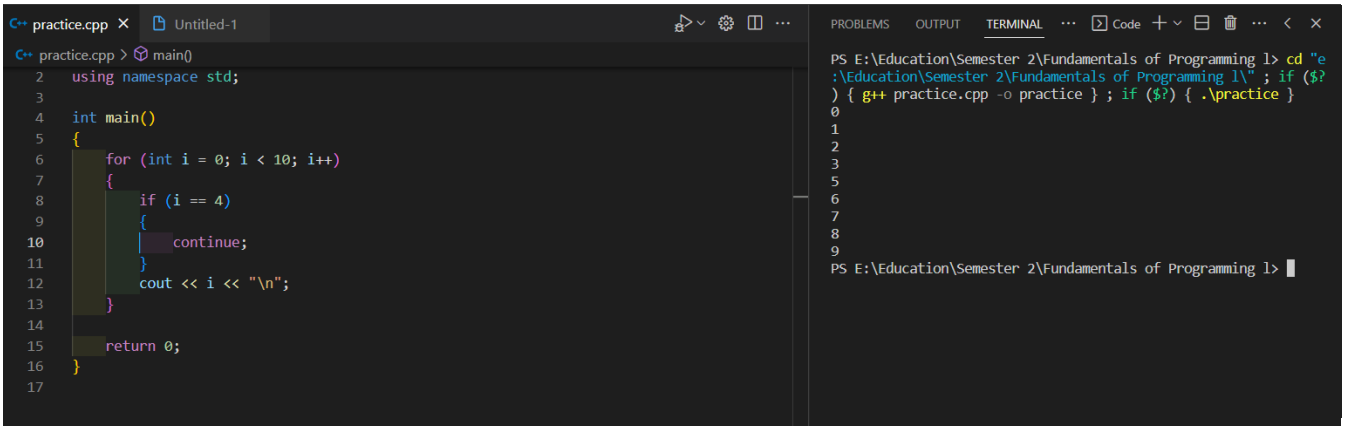
```
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?
) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
10
20
30
40
50
PS E:\Education\Semester 2\Fundamentals of Programming 1>
```

break



The break statement can also be used to jump out of a loop.

The continue



The continue statement breaks one iteration (in the loop)

```cpp
#include <iostream>
using namespace std;

int main() {
  int i = 0;
  while (i < 10) {
    cout << i << "\n";
    i++;
    if (i == 4) {
      break;
    }
  }
  return 0;
}
```

```
0
1
2
3
```

```cpp
#include <iostream>
using namespace std;

int main() {
  int i = 0;
  while (i < 10) {
    if (i == 4) {
      i++;
      continue;
    }
    cout << i << "\n";
    i++;
  }
  return 0;
}
```

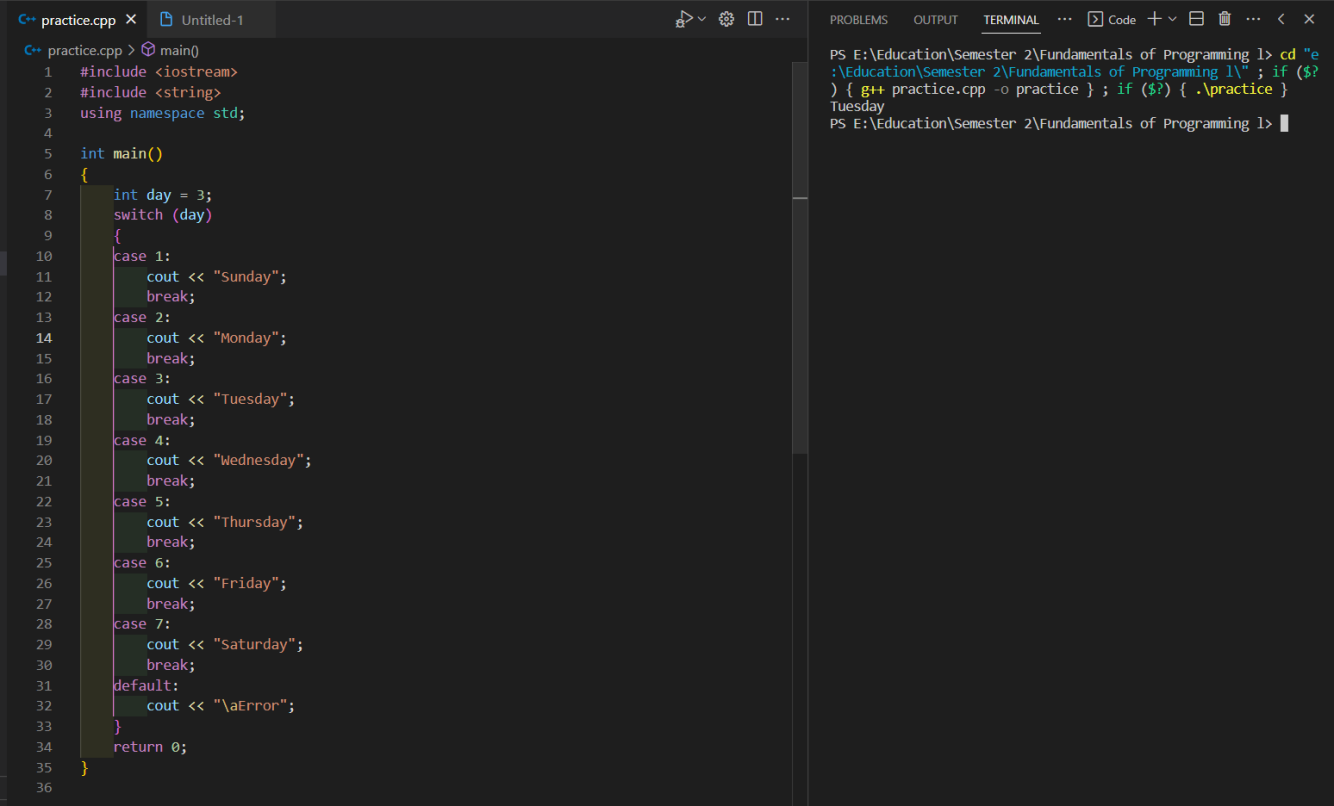
```
0
1
2
3
5
6
7
8
9
```

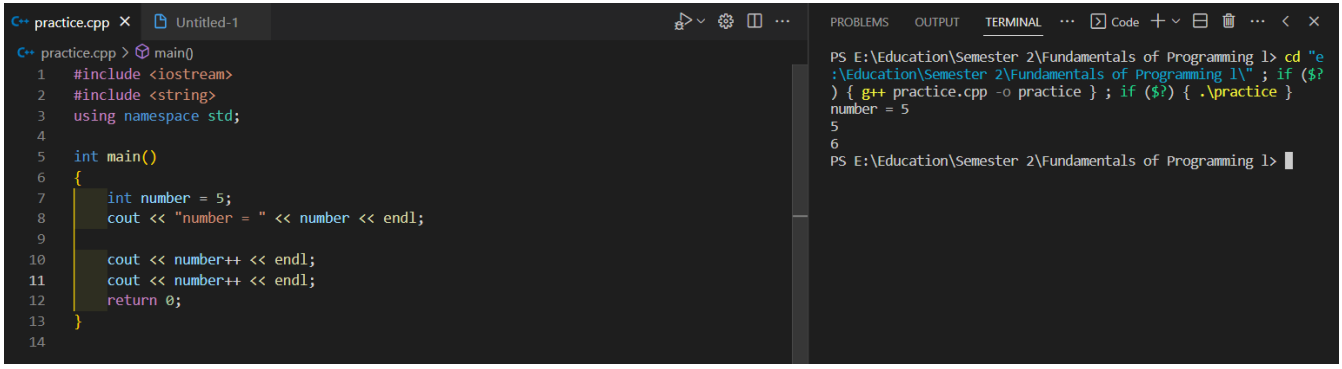
```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
  int time = 20;
  string result = (time < 18) ? "Good day." : "Good evening.";
  cout << result;
  return 0;
}
```

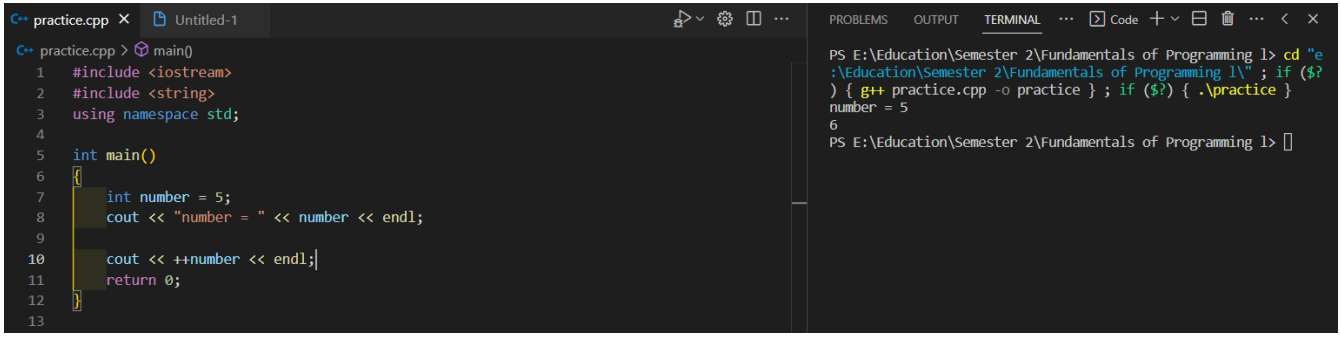
```
Good evening.
```

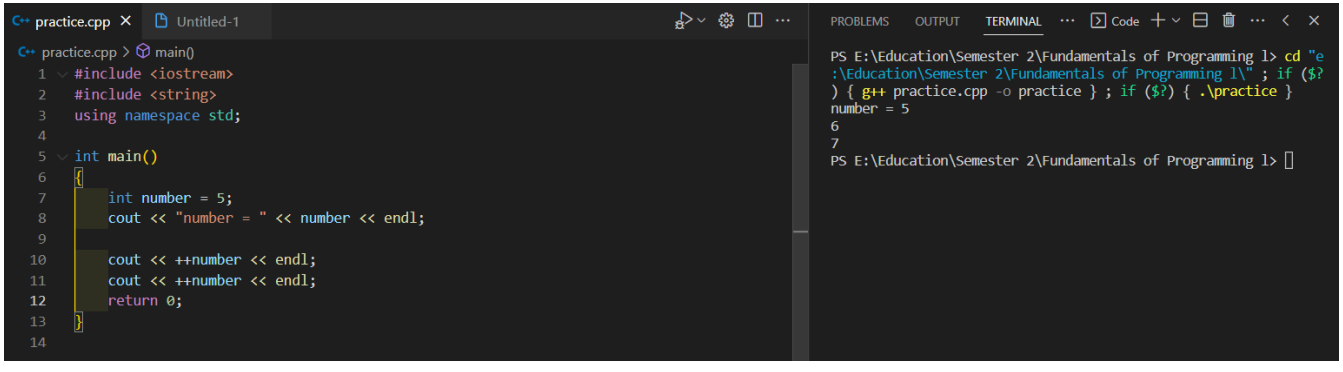
```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    int time = 15;
    cout << "Time = " << time << endl;
    int result = (time < 12) ? time : time - 12;
    cout << "result = " << result << endl;
    return 0;
}
```

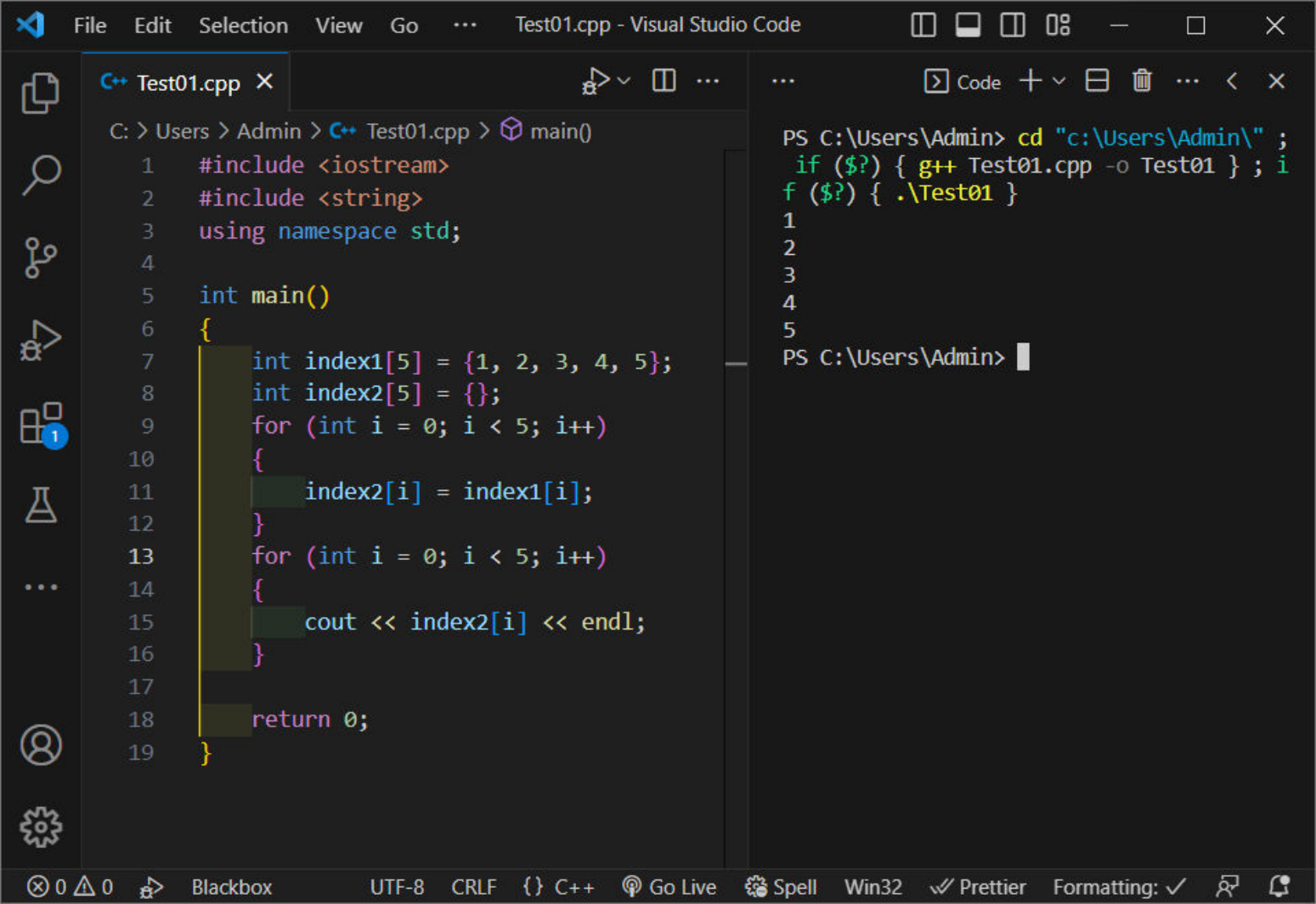
```
Time = 15
result = 3
```

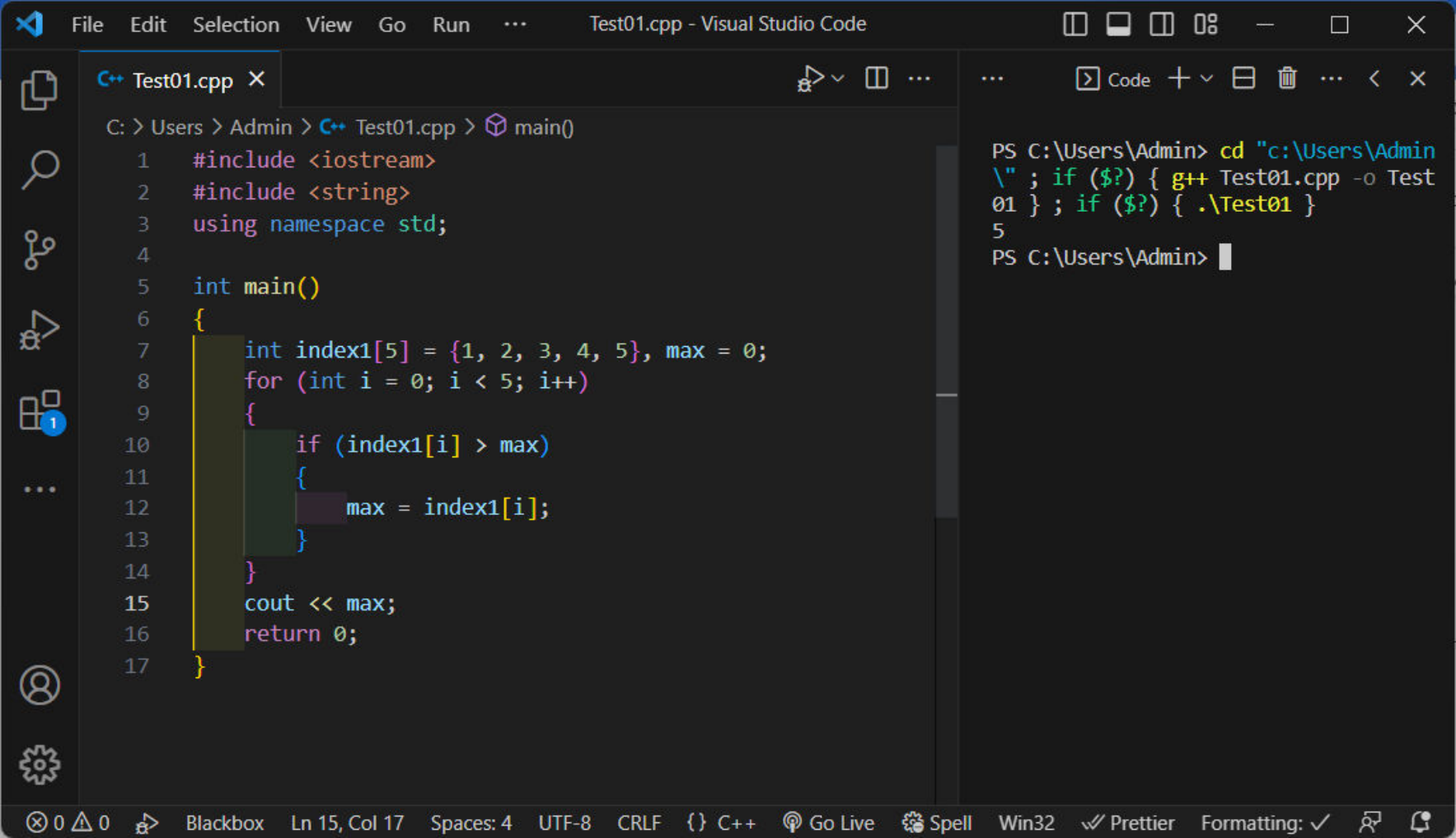

practice.cpp
Untitled-1
practice.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int day = 3;
switch (day)
{
case 1:
cout << "Sunday";
break;
case 2:
cout << "Monday";
break;
case 3:
cout << "Tuesday";
break;
case 4:
cout << "Wednesday";
break;
case 5:
cout << "Thursday";
break;
case 6:
cout << "Friday";
break;
case 7:
cout << "Saturday";
break;
default:
cout << "\aError";
}
return 0;
}
PROBLEMS
OUTPUT
TERMINAL
Code
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
Tuesday
PS E:\Education\Semester 2\Fundamentals of Programming 1>

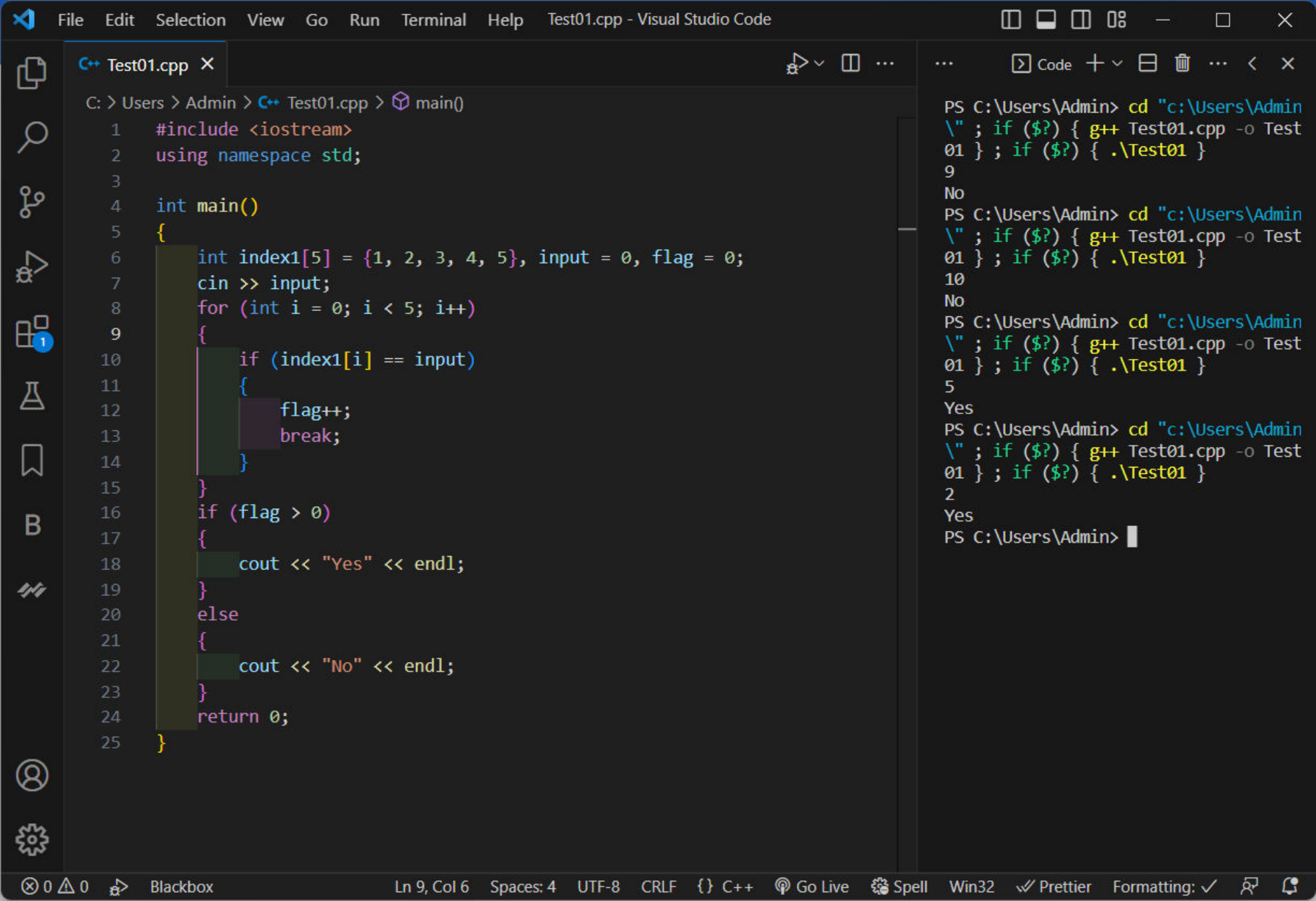

practice.cpp
Untitled-1
practice.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int number = 5;
cout << "number = " << number << endl;
cout << number++ << endl;
cout << number++ << endl;
return 0;
}
PROBLEMS
OUTPUT
TERMINAL
Code
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?
) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
number = 5
5
6
PS E:\Education\Semester 2\Fundamentals of Programming 1>


practice.cpp
Untitled-1
practice.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int number = 5;
cout << "number = " << number << endl;
cout << ++number << endl;
return 0;
}
PROBLEMS
OUTPUT
TERMINAL
Code
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?
) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
number = 5
6
PS E:\Education\Semester 2\Fundamentals of Programming 1>


practice.cpp
Untitled-1
practice.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int number = 5;
cout << "number = " << number << endl;
cout << ++number << endl;
cout << ++number << endl;
return 0;
}
PROBLEMS
OUTPUT
TERMINAL
Code
PS E:\Education\Semester 2\Fundamentals of Programming 1> cd "e
:\Education\Semester 2\Fundamentals of Programming 1\" ; if ($?
) { g++ practice.cpp -o practice } ; if ($?) { .\practice }
number = 5
6
7
PS E:\Education\Semester 2\Fundamentals of Programming 1>


File Edit Selection View Go ··· Test01.cpp - Visual Studio Code
Test01.cpp
C: > Users > Admin > Test01.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int index1[5] = {1, 2, 3, 4, 5};
int index2[5] = {};
for (int i = 0; i < 5; i++)
{
index2[i] = index1[i];
}
for (int i = 0; i < 5; i++)
{
cout << index2[i] << endl;
}
return 0;
}
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
1
2
3
4
5
PS C:\Users\Admin>
0 0 Blackbox UTF-8 CRLF {} C++ Go Live Spell Win32 Prettier Formatting:


File Edit Selection View Go Run ···
Test01.cpp - Visual Studio Code
Test01.cpp
C: > Users > Admin > Test01.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int index1[5] = {1, 2, 3, 4, 5}, max = 0;
for (int i = 0; i < 5; i++)
{
if (index1[i] > max)
{
max = index1[i];
}
}
cout << max;
return 0;
}
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
5
PS C:\Users\Admin>
0 0 Blackbox Ln 15, Col 17 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Prettier Formatting:


File Edit Selection View Go Run Terminal Help Test01.cpp - Visual Studio Code
Test01.cpp
C: > Users > Admin > Test01.cpp > main()
#include <iostream>
using namespace std;
int main()
{
int index1[5] = {1, 2, 3, 4, 5}, input = 0, flag = 0;
cin >> input;
for (int i = 0; i < 5; i++)
{
if (index1[i] == input)
{
flag++;
break;
}
}
if (flag > 0)
{
cout << "Yes" << endl;
}
else
{
cout << "No" << endl;
}
return 0;
}
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
9
No
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
10
No
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
5
Yes
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
2
Yes
PS C:\Users\Admin>
0 0 Blackbox Ln 9, Col 6 Spaces: 4 UTF-8 CRLF C++ Go Live Spell Win32 Prettier Formatting:

File Edit Selection View Go Run Terminal Help Test01.cpp - Visual Studio Code

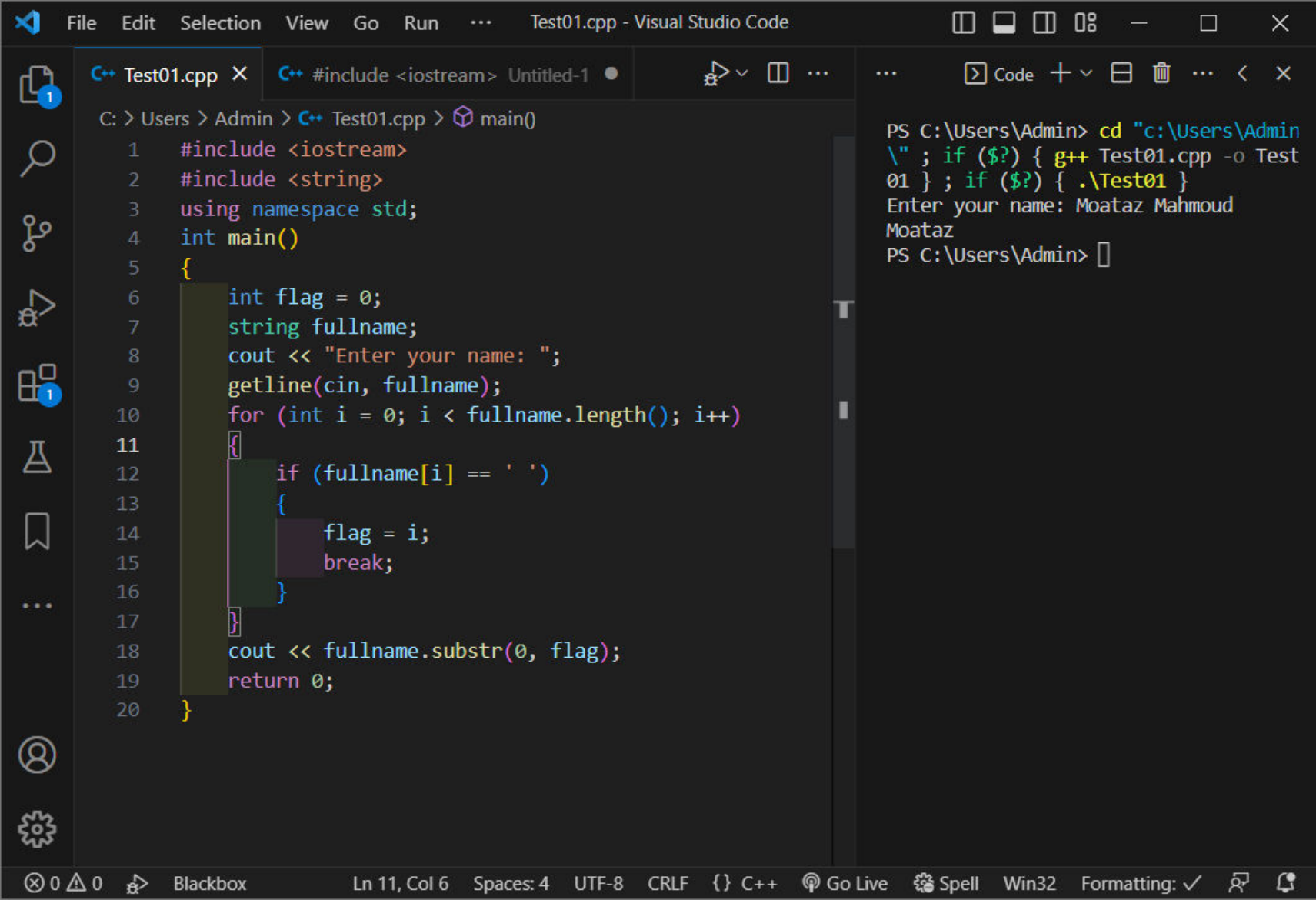
Test01.cpp | #include <iostream> Untitled-1

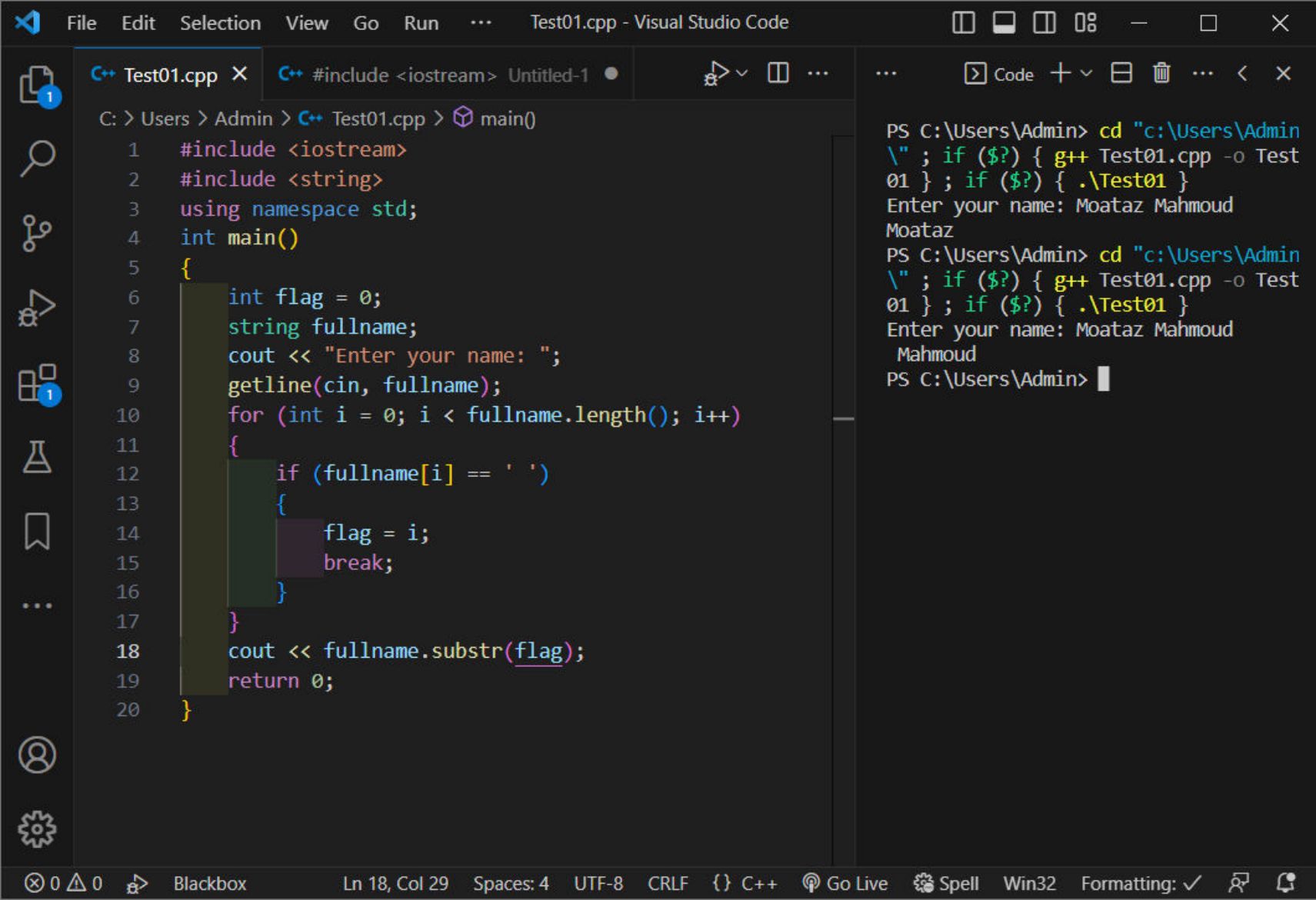
C: > Users > Admin > Test01.cpp > main()

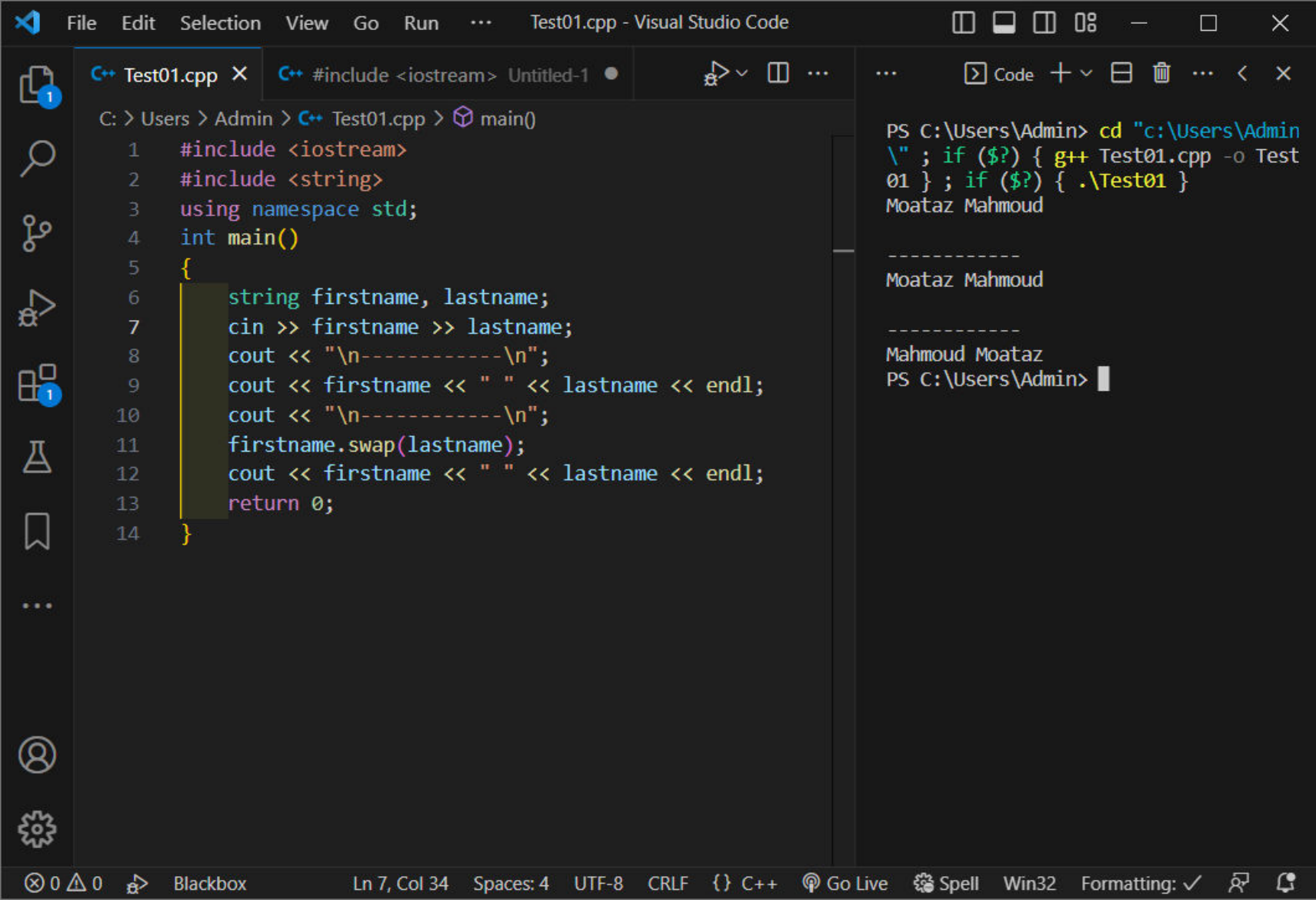
```cpp
#include <iostream>
using namespace std;
int main()
{
    int t;
    int x[5] = {6, 5, 4, 1, 2};
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (x[j] > x[j + 1])
            {
                t = x[j];
                x[j] = x[j + 1];
                x[j + 1] = t;
            }
        }
    }
    for (int r = 0; r < 5; r++)
    {
        cout << x[r] << endl;
    }

    return 0;
}
```

```
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12345
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12366
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12245
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12345
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12377
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
0
1
2
4
5
PS C:\Users\Admin>
```

0 0 Blackbox Ln 9, Col 36 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Formatting:

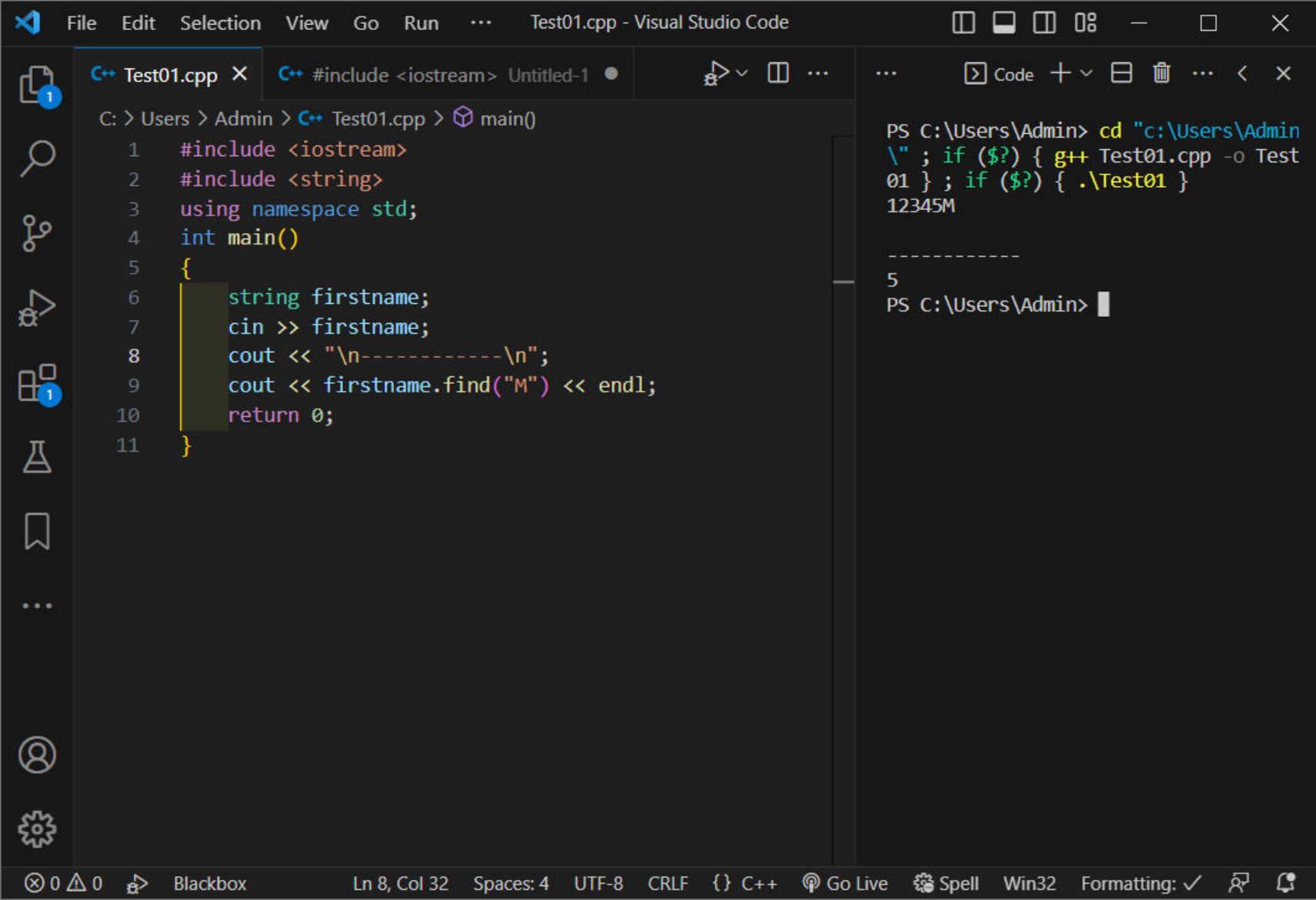

File Edit Selection View Go Run ··· Test01.cpp - Visual Studio Code
Test01.cpp
#include <iostream> Untitled-1
C: > Users > Admin > Test01.cpp > main()
#include <iostream>
#include <string>
using namespace std;
int main()
{
int flag = 0;
string fullname;
cout << "Enter your name: ";
getline(cin, fullname);
for (int i = 0; i < fullname.length(); i++)
{
if (fullname[i] == ' ')
{
flag = i;
break;
}
}
cout << fullname.substr(0, flag);
return 0;
}
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
Enter your name: Moataz Mahmoud
Moataz
PS C:\Users\Admin>
0 0 Blackbox Ln 11, Col 6 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Formatting:


File Edit Selection View Go Run ··· Test01.cpp - Visual Studio Code
Test01.cpp
#include <iostream> Untitled-1
C: > Users > Admin > Test01.cpp > main()
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main()
5 {
6 int flag = 0;
7 string fullname;
8 cout << "Enter your name: ";
9 getline(cin, fullname);
10 for (int i = 0; i < fullname.length(); i++)
11 {
12 if (fullname[i] == ' ')
13 {
14 flag = i;
15 break;
16 }
17 }
18 cout << fullname.substr(flag);
19 return 0;
20 }
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
Enter your name: Moataz Mahmoud
Moataz
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
Enter your name: Moataz Mahmoud
Mahmoud
PS C:\Users\Admin>
0 0 Blackbox Ln 18, Col 29 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Formatting:


File Edit Selection View Go Run ··· Test01.cpp - Visual Studio Code
Test01.cpp
#include <iostream> Untitled-1
C: > Users > Admin > Test01.cpp > main()
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main()
5 {
6 string firstname, lastname;
7 cin >> firstname >> lastname;
8 cout << "\n------------\n";
9 cout << firstname << " " << lastname << endl;
10 cout << "\n------------\n";
11 firstname.swap(lastname);
12 cout << firstname << " " << lastname << endl;
13 return 0;
14 }
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
Moataz Mahmoud

Moataz Mahmoud

Mahmoud Moataz
PS C:\Users\Admin>
0 0 Blackbox Ln 7, Col 34 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Formatting:


File Edit Selection View Go Run ··· Test01.cpp - Visual Studio Code
Test01.cpp
#include <iostream> Untitled-1
C: > Users > Admin > Test01.cpp > main()
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int main()
5 {
6 string firstname;
7 cin >> firstname;
8 cout << "\n------------\n";
9 cout << firstname.find("M") << endl;
10 return 0;
11 }
Code
PS C:\Users\Admin> cd "c:\Users\Admin\" ; if ($?) { g++ Test01.cpp -o Test01 } ; if ($?) { .\Test01 }
12345M

5
PS C:\Users\Admin>
0 0 Blackbox Ln 8, Col 32 Spaces: 4 UTF-8 CRLF {} C++ Go Live Spell Win32 Formatting: