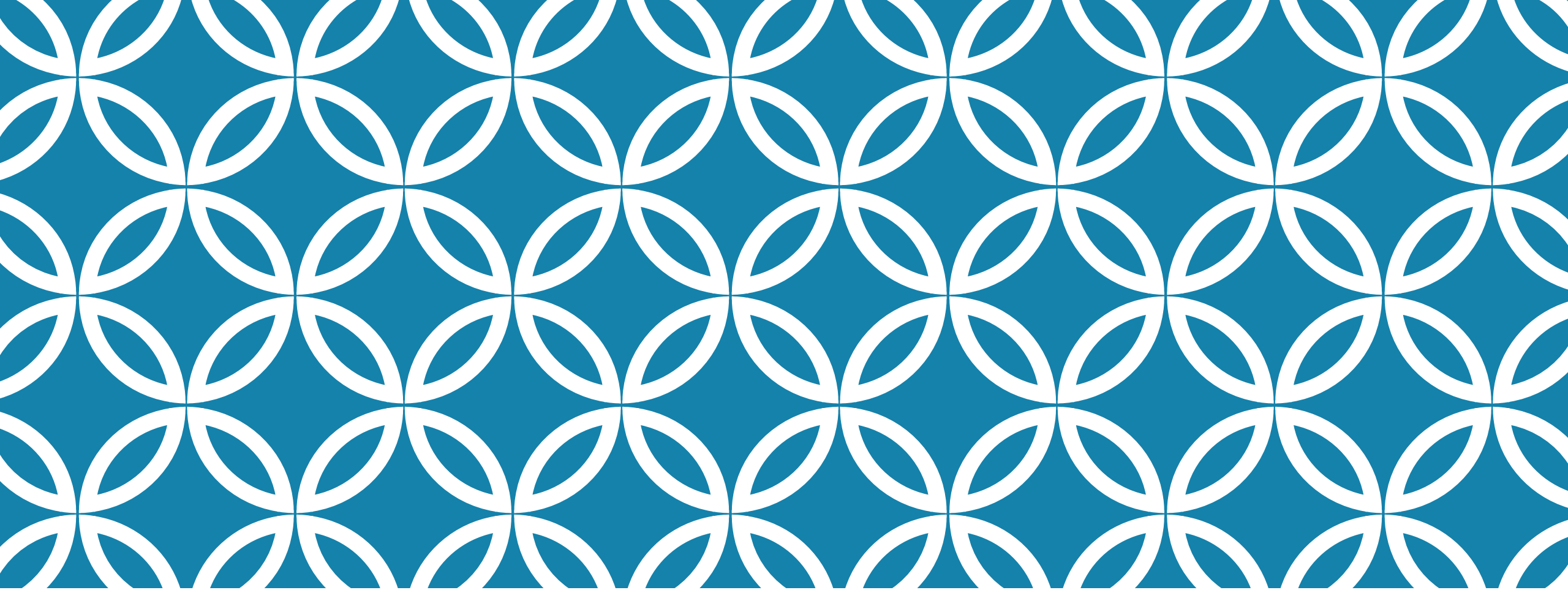


Introduction of C++ Section 3

Dr/Ghada Maher
Eng/ Hossam Medhat

C++ MATH

❖C++ <cmath> Header :

➢ Other functions, such as sqrt (square root), round (rounds a number)

*main.cpp

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      cout << sqrt(64) << endl;
8  cout << round(3.6)<< endl;
9  cout << log(2);
10
11     return 0;
12 }
```

C:\Users\hossam\Desktop\stringg\string\bin\Debug\string.exe

```
8
4
0.693147
Process returned 0 (0x0)   execution time : 0.053 s
Press any key to continue.
```

C++ MATH "CONT"

❖ **C++ <cmath> Header :**

- **C++ has many functions that allows you to perform mathematical tasks on numbers.**

1) **Max (x,y) function :**

- **Max (x,y) function can be used to find the highest value of x and y.**
- **Example :**

main.cpp

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7     cout << max(5, 10);
8        return 0;
9  }
```

C:\Users\hossam\Desktop\stringg\string\bin\Debug\string.exe

```
10
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

C++ MATH "CONT"

❖ **C++ <cmath> Header :**

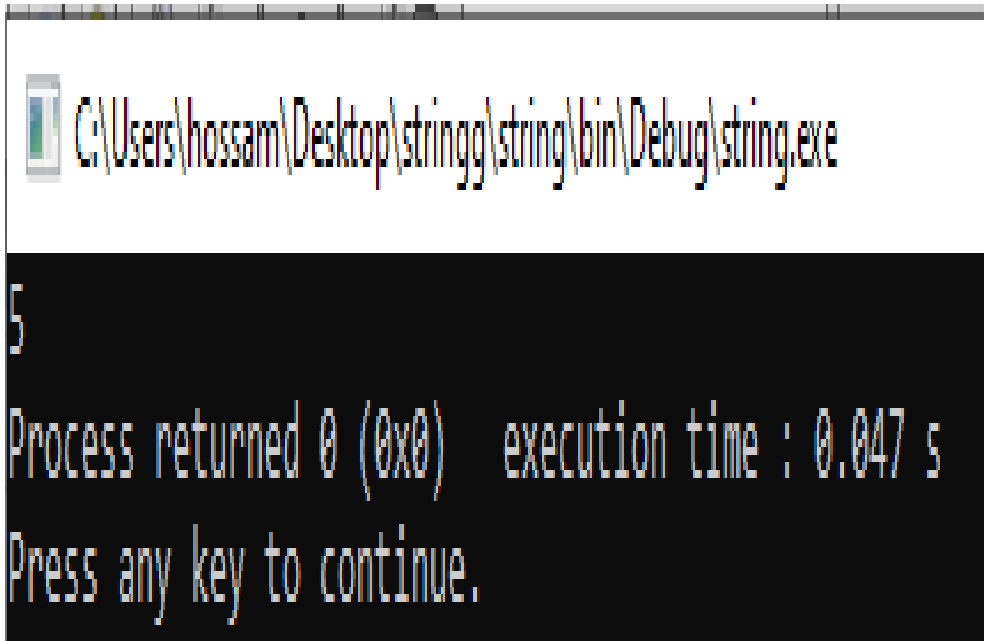
➢ **C++ has many functions that allows you to perform mathematical tasks on numbers.**

2) Min(x,y) function :

➢ **Min(x,y) function can be used to find the lowest value of x and y.**

*main.cpp

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      cout << min(5, 10);
8      return 0;
9  }
```



C++ STRING CONCATENATION

- The + operator can be used between strings to add them together to make a new string. This is called concatenation.

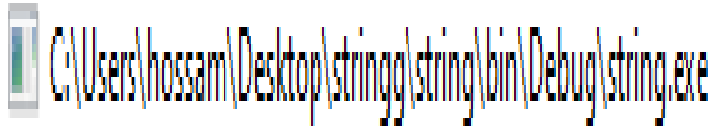
```
17  #include <iostream>
18  using namespace std;
19  int main()
20  {
21      string firstName = "John ";
22  string lastName = "Doe";
23  string fullName = firstName + lastName;
24  cout << fullName;
25      return 0;
26  }
```

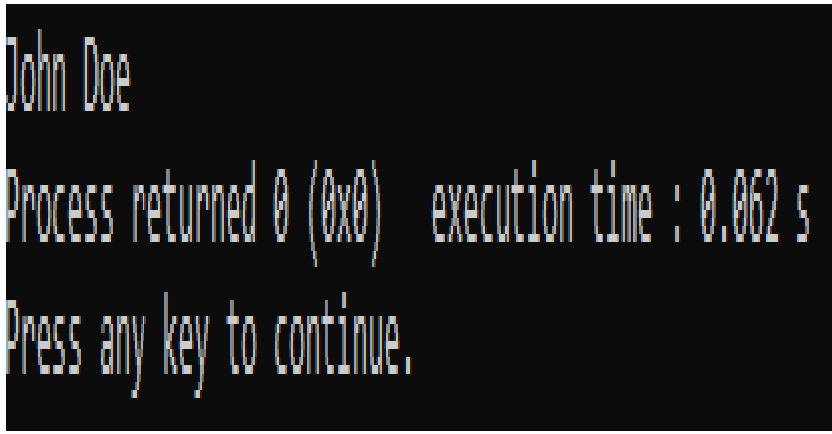

```
John Doe
Process returned 0 (0x0)   execution time : 0.053 s
Press any key to continue.
```

C++ STRING CONCATENATION " CONT"

- **Append() function:**
  - **Append() function used also to concatenate strings .**

```
33  #include <iostream>
34
35  using namespace std;
36
37  int main()
38  {
39      string firstName = "John ";
40  string lastName = "Doe";
41  string fullName = firstName.append(lastName);
42  cout << fullName;
43      return 0;
44  }
```





C++ GETLINE() FUNCTION

❖ **getline() function :**

- **getline() function used to read a line of text.**
- **Syntax :**

```
getline (cin, variable_name);
```

```
47  #include <iostream>
48
49  using namespace std;
50
51  int main()
52  {
53      string fullName;
54      cout << "Type your full name: ";
55      getline (cin, fullName);
56      cout << "Your name is: " << fullName;
57      return 0;
58  }
```

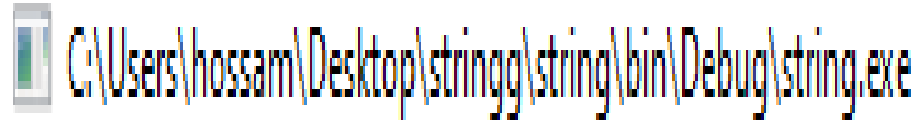
```
C:\Users\hossam\Desktop\stringg\string\bin\Debug\string.exe
Type your full name: hossam
Your name is: hossam
Process returned 0 (0x0)   execution time : 5.351 s
Press any key to continue.
```

C++ ADDING NUMBERS AND STRINGS

- C++ uses the + operator for both addition and concatenation.
- Numbers are added. Strings are concatenated.

1) Add two numbers :

```
63  #include <iostream>
64
65  using namespace std;
66
67  int main()
68  {
69  int x = 10;
70  int y = 20;
71  int z = x + y;
72  cout << z;
73      return 0;
74  }
```




```
30

Process returned 0 (0x0)   execution time : 0.038 s

Press any key to continue.
```

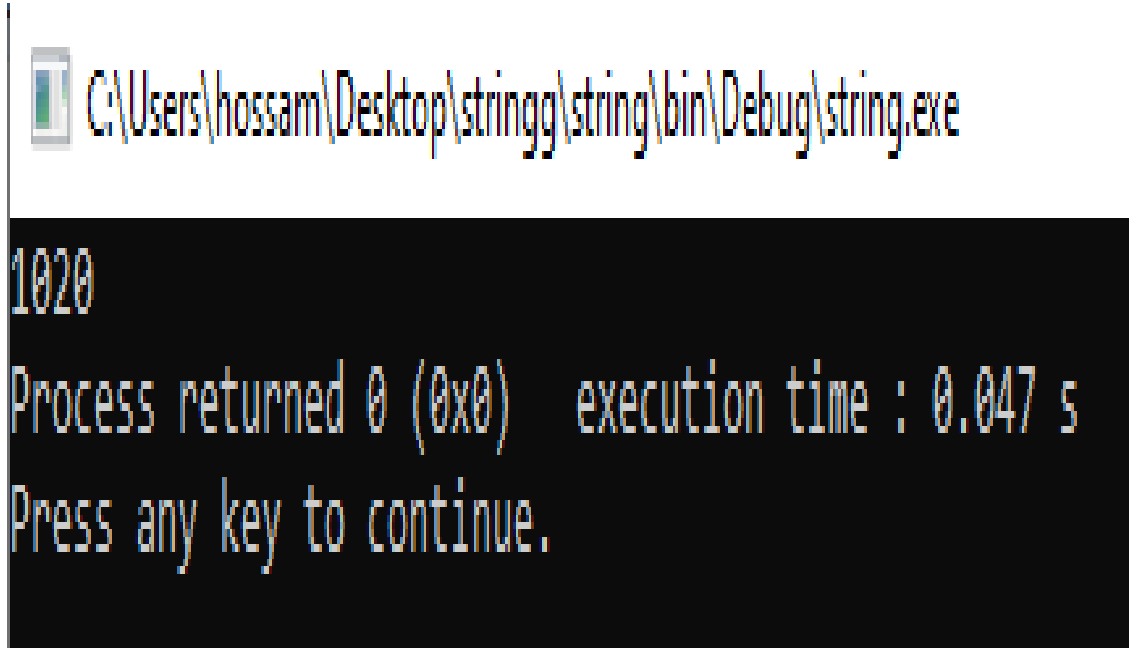
C++ ADDING NUMBERS AND STRINGS " CONT"

❖C++ uses the + operator for both addition and concatenation.

➢ Numbers are added. Strings are concatenated.

1) Add two strings :

```
78  #include <iostream>
79
80  using namespace std;
81
82  int main()
83  {
84  string x = "10";
85  string y = "20";
86  string z = x + y;
87  cout << z;
88      return 0;
89  }
```



C++ ADDING NUMBERS AND STRINGS " CONT"

- **What will happen when add number to string ?**

```
 93  #include <iostream>
 94
 95  using namespace std;
 96
 97  int main()
 98  {
 99  string x = "10";
100  int y = 20;
101  string z = x + y;
102  cout << z;
103      return 0;
104  }
```

```
 93  #include <iostream>
 94
 95  using namespace std;
 96
 97  int main()
 98  {
 99  string x = "10";
100  int y = 20;
101  string z = x + y;
102  cout << z;
103      return 0;
104  }
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++

File	Line	Message
		=== Build: Debug in string (compiler: GNU GCC Compiler) ===
C:\Users\hoss...		In function 'int main()':
C:\Users\hoss...	101	error: no match for 'operator+' (operand types are 'std::__cxx11::string' {aka 'std::_...

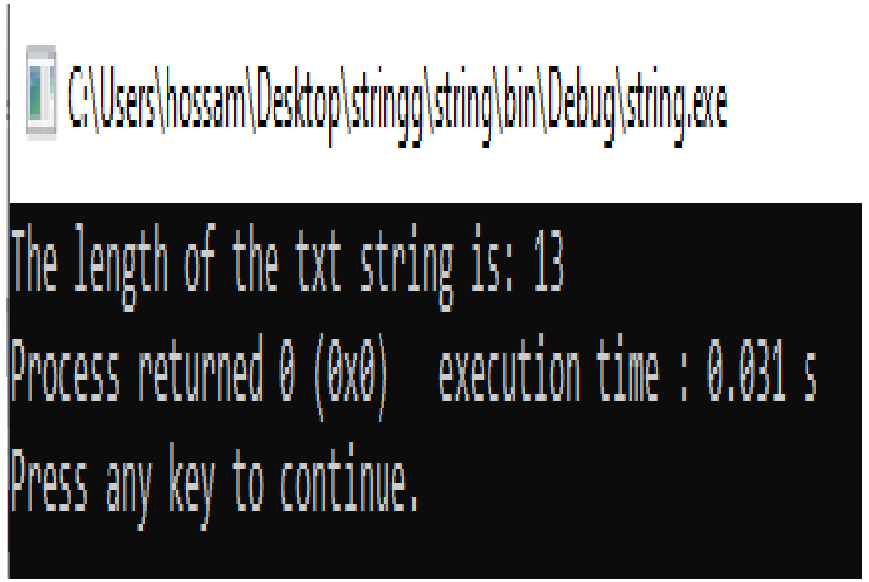
C++ STRING LENGTH() OR SIZE():

- **String Length() or size() is used to get the length or size of a string .**

```cpp
#include <iostream>

using namespace std;

int main()
{
string txt = "Hossam Medhat";
cout << "The length of the txt string is: " << txt.length();
    return 0;
}
```




```cpp
#include <iostream>

using namespace std;

int main()
{
string txt = "Hossam Medhat";
cout << "The length of the txt string is: " << txt.size();
    return 0;
}
```

C++ ACCESS STRINGS

- In C++ You can access the characters in a string by referring to its index number inside square brackets [ ].
- String indexes start with 0: [0] is the first character, [1] is the

```
124  // index of string
125  #include <iostream>
126
127  using namespace std;
128
129  int main()
130  {
131  string myString = "Hello";
132  cout << myString[0]<<endl;
133  cout << myString[1];
134      return 0;
135  }
```

```
C:\Users\hossam\Desktop\stringg\string\bin\Debug\string.exe
H
e
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

C++ ACCESS STRINGS " CONT"

- In C++ if you need To change the value of a specific character in a string, refer to the index number, and use single quotes.

```
125  #include <iostream>
126
127  using namespace std;
128
129  int main()
130  {
131  string myString = "Hello";
132  myString[0] = 'J';
133  cout << myString;
134      return 0;
135  }
136
```

Outputs Jello instead of Hello

IF STATEMENT IN C++

- In C++ programming, if statement is used to test the condition.
- There are various types of if statements in C++ :
  1) if statement
  2) if-else statement
  3) nested if statement

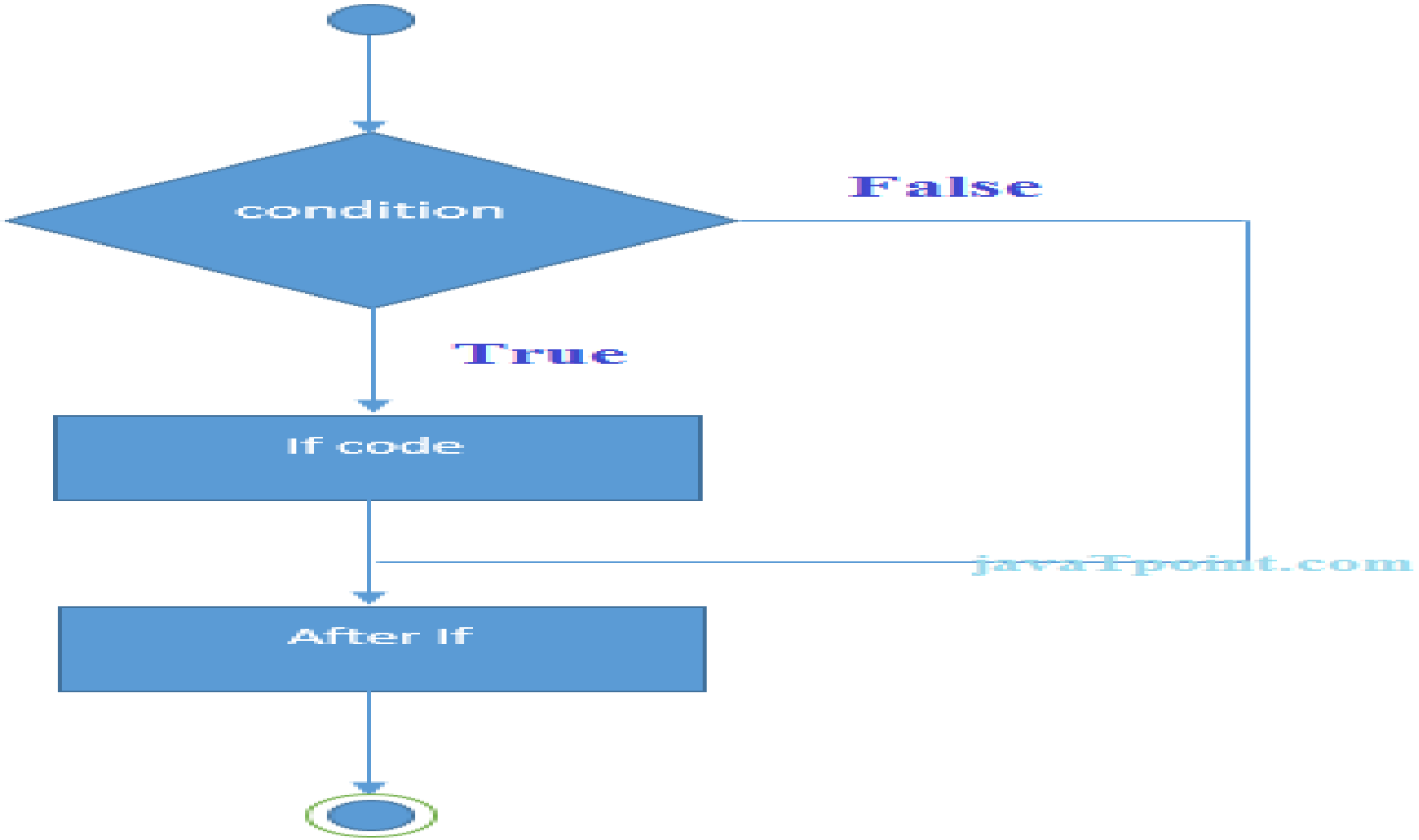
IF STATEMENT IN C++" CONT"

- **Decision Making:**
  - **The if statement is used to execute some code if a condition is true.**
  - **Syntax:**

```
if(condition)
{
//statements
}
```

  - **The condition specifies which expression is to be evaluated. If the condition is true,the statements in curly brackets are executed.**

IF STATEMENT IN C++“ CONT”



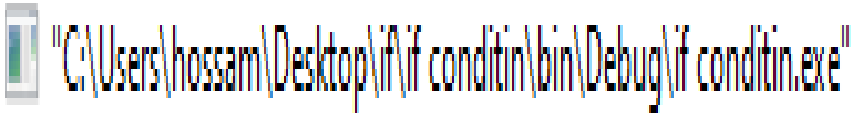
IF STATEMENT IN C++“ CONT”

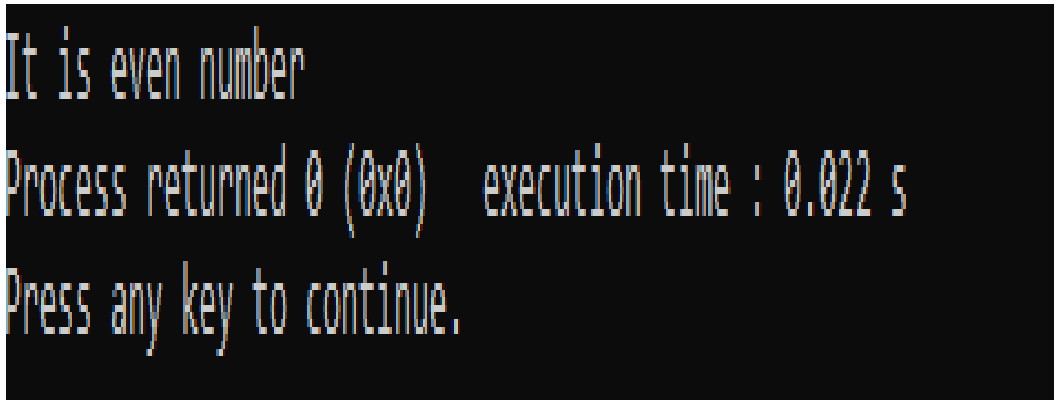
- Use relational operators to evaluate conditions.

Example

main.cpp

```
1
2   //C++ If Example
3   #include <iostream>
4   using namespace std;
5
6   int main () {
7       int num = 10;
8               if (num % 2 == 0)
9               {
10                  cout<<"It is even number";
11              }
12      return 0;
13  }
```





IF ELSE STATEMENT IN C++

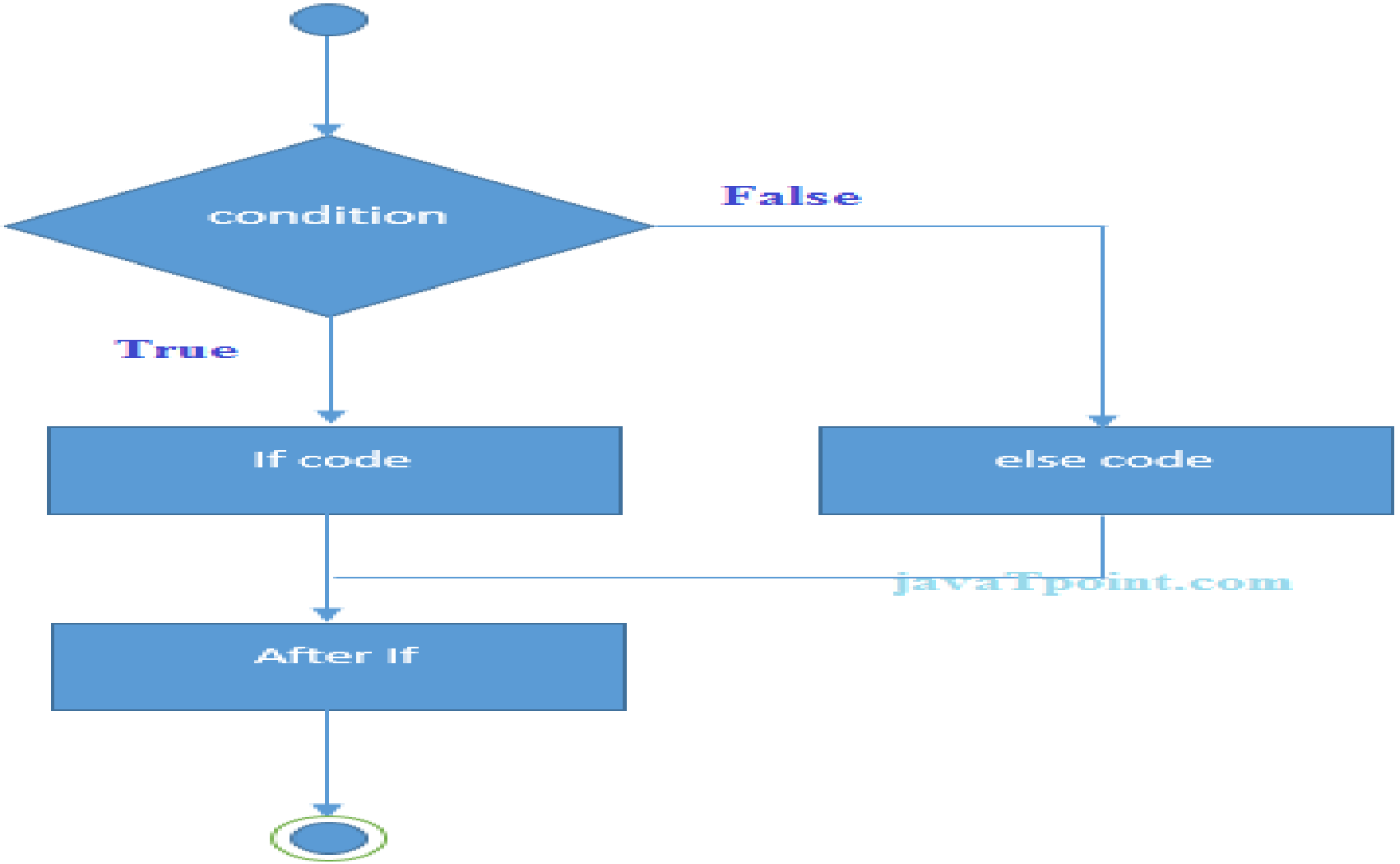
❖ **Else statement:**

- An if statement can be followed by an optional else statement, which executes when the condition is false.
- **Syntax:**

```
if(condition) {
//statements
}
else {
//statements
}
```

- The compiler will test the condition:-
- If it evaluates to true, then the code inside the if statement will be executed
- If it evaluates to false, then the code inside the else statement will be executed

IF ELSE STATEMENT IN C++“ CONT”



IF ELSE STATEMENT IN C++“ CONT”

- Use relational operators to evaluate conditions.

Example

main.cpp

```
16  //C++ If-else Example
17  #include <iostream>
18  using namespace std;
19  int main () {
20      int num = 11;
21              if (num % 2 == 0)
22              {
23                  cout<<"It is even number";
24              }
25              else
26              {
27                  cout<<"It is odd number";
28              }
29      return 0;
30  }
```

"C:\Users\hossam\Desktop\if\if conditin\bin\Debug\if conditin.exe"

```
It is odd number

Process returned 0 (0x0)   execution time : 0.017 s

Press any key to continue.
```

IF ELSE STATEMENT IN C++“ CONT”

Example

```
int mark = 90;
if(mark<33) {
cout<<"You failed"<<endl;
}
else{
cout<<You passed"<<endl;
}
```

Outputs : "You passed"

IF ELSE STATEMENT IN C++“ CONT”

Example

```
int time = 22;
if (time < 10) {
  cout << "Good morning.";
} else if (time < 20) {
  cout << "Good day.";
} else {
  cout << "Good evening.";
}
```

Outputs : "Good evening."

C++ SHORT HAND IF ELSE

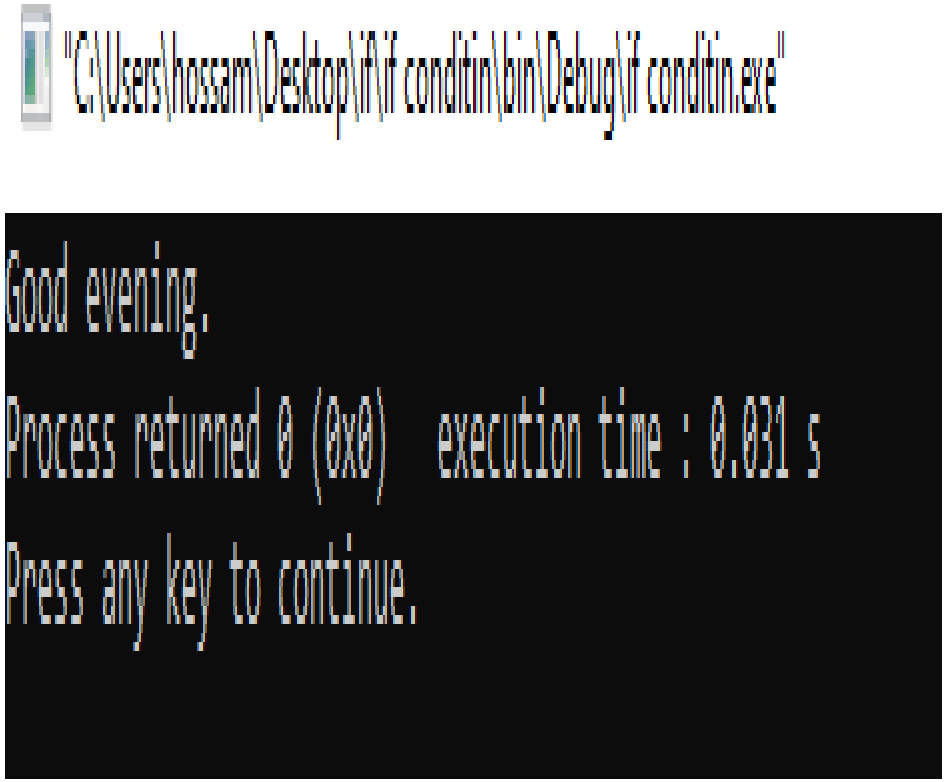
- **Else statement:**
  - a short-hand if else which is known as the ternary operator because it consists of three operands. It can be used to replace multiple lines of code with a single line.
  - It is often used to replace simple if else statements.
  - **Syntax:**

```
variable = (condition) ? expressionTrue :
expressionFalse;
```

C++ SHORT HAND IF ELSE " CONT"

Example

```
33  //example of Short Hand If Else
34  #include <iostream>
35  using namespace std;
36
37  int main () {
38    int time = 20;
39  if (time < 18) {
40    cout << "Good day.";
41  } else {
42    cout << "Good evening.";
43  }
44     return 0;
45  }
```



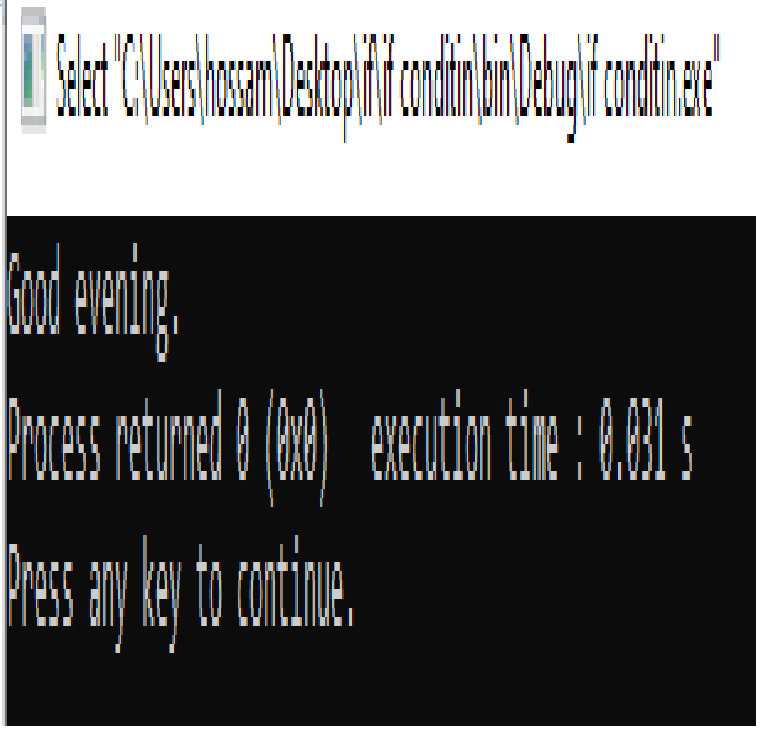
C++ SHORT HAND IF ELSE " CONT"

- **You can simply write :**

Example

main.cpp

```
47  //----------------------
48
49  //You can simply write:
50  #include <iostream>
51  using namespace std;
52
53  int main () {
54  int time = 20;
55  string result = (time < 18) ? "Good day." : "Good evening.";
56  cout << result;
57     return 0;
58  }
```

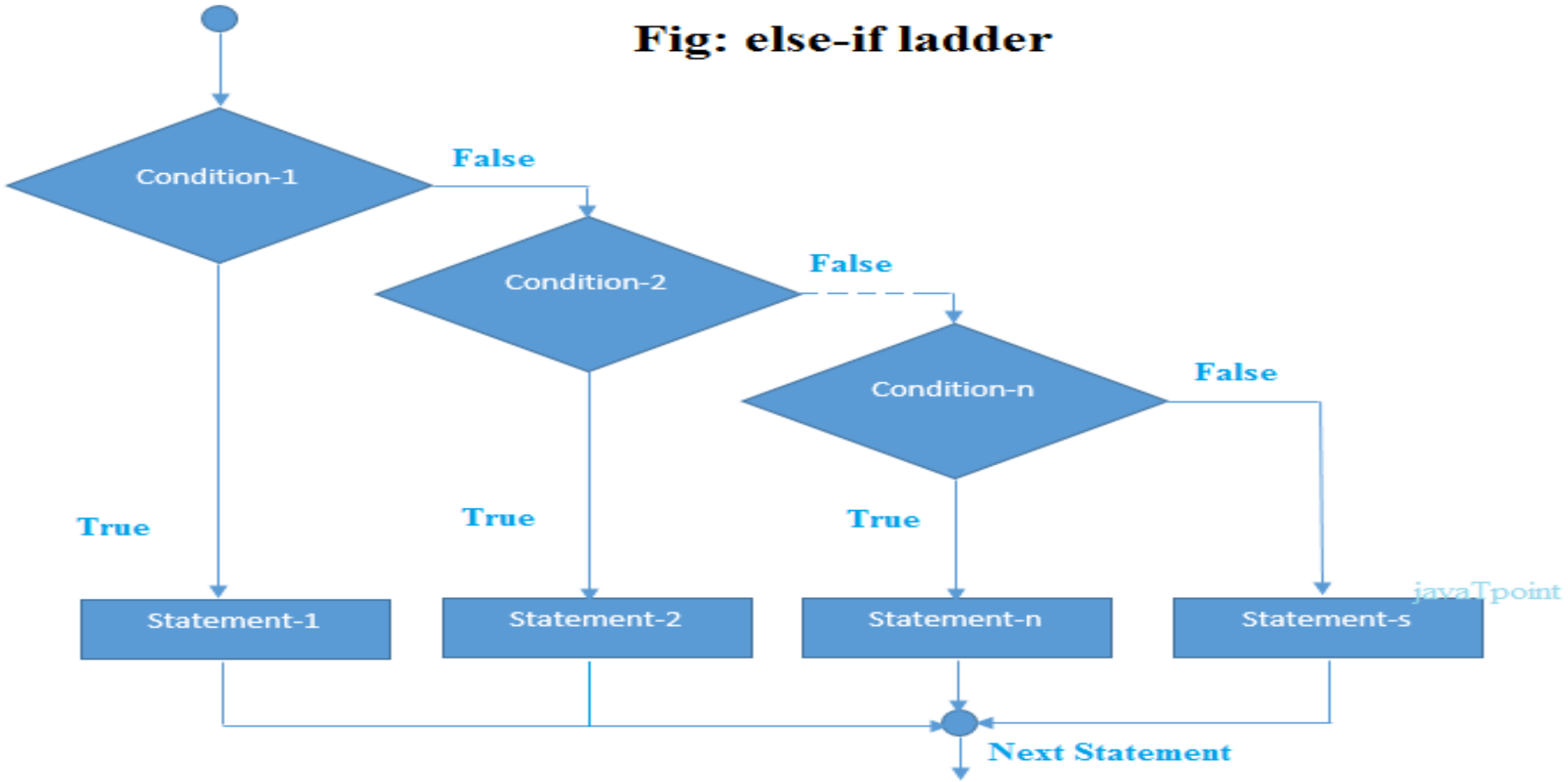


NESTED IF & IF/ELSE STATEMENTS IN C++

- **Nested if statements:** nesting an unlimited number of if/else statements.
- **Syntax:**

```
if(condition1){
//code to be executed if condition1 is true
}else if(condition2){
//code to be executed if condition2 is true
}
else if(condition3){
//code to be executed if condition3 is true
}
else{
//code to be executed if all the conditions are
false
}
```

NESTED IF & IF/ELSE STATEMENTS IN C++“ CONT”

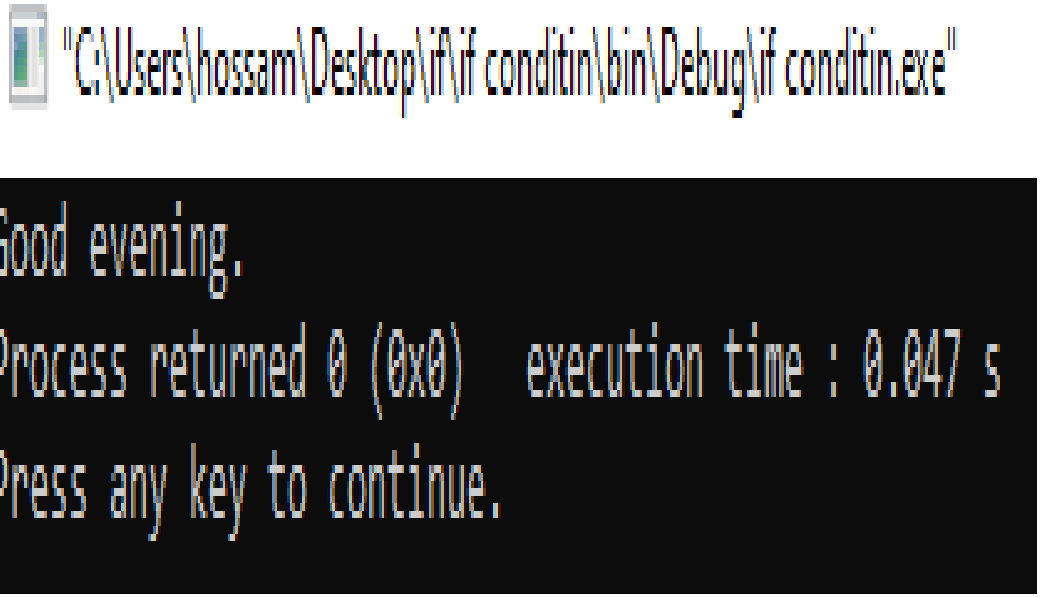


NESTED IF & IF/ELSE STATEMENTS IN C++“ CONT”

Example

*main.cpp

```
81
82  #include <iostream>
83  using namespace std;
84  int main () {
85  int time = 22;
86  if (time < 10) {
87    cout << "Good morning.";
88  } else if (time < 20) {
89    cout << "Good day.";
90  } else {
91    cout << "Good evening.";
92  }
93  return 0;
94  }
```



NESTED IF & IF/ELSE STATEMENTS IN C++“ CONT”

Example

```
int mark = 100;
if (mark >= 50) {
  cout << "You passed." << endl;
  if (mark == 100) {
    cout <<"Perfect!" << endl;  }
} else {
  cout << "You failed." << endl;
}
```

Outputs : “ You passed “

“ Perfect! “

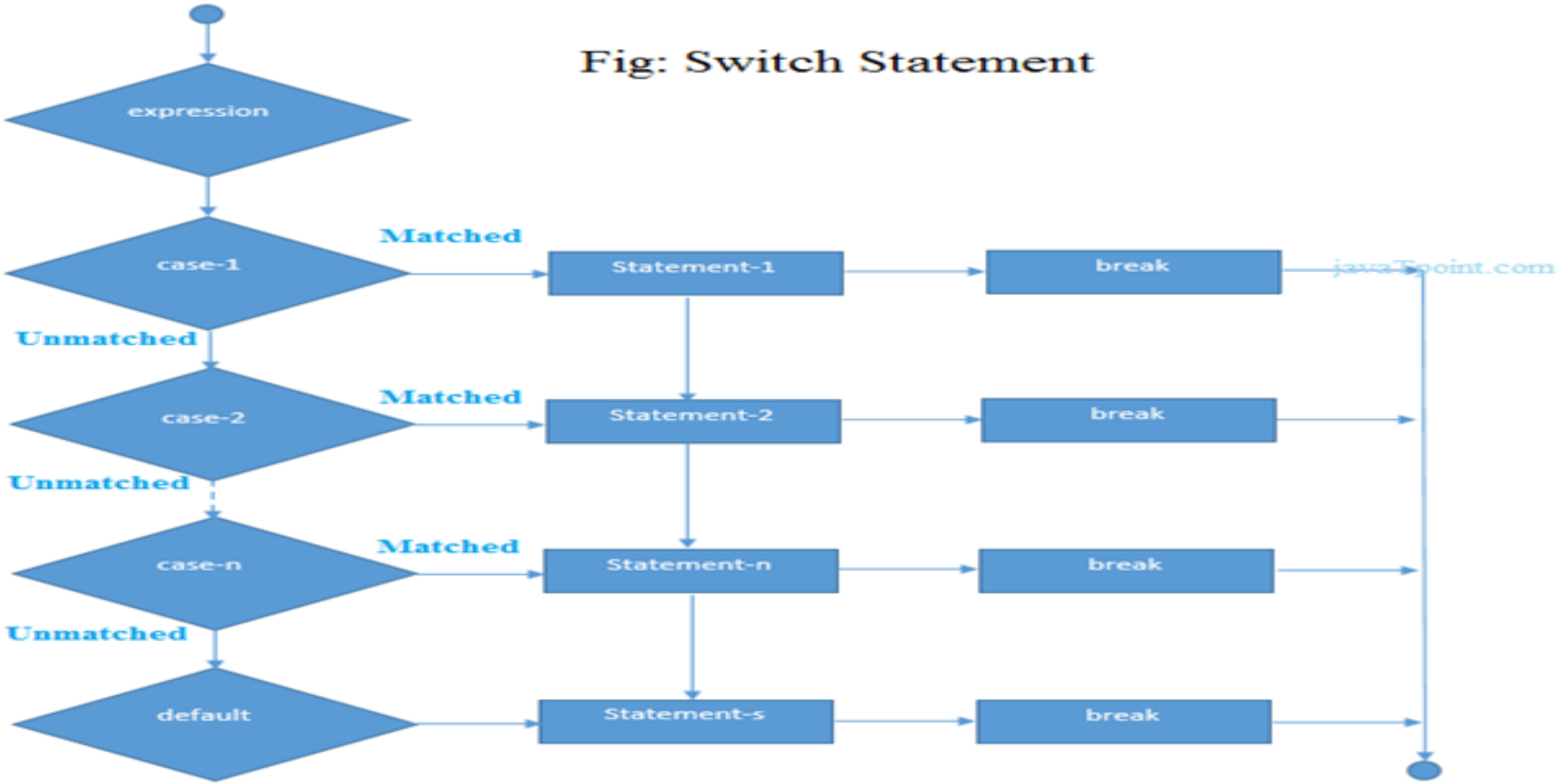
C++ SWITCH STATEMENTS

- C++ switch statement is like nested if statement in C++.
- Use the switch statement to select one of many code blocks to be executed.
  - Syntax:

```
switch(expression) {
  case x:  // code block
break;
  case y:   // code block
    break;
  default:
    // code block
}
```

- The switch expression is evaluated once
- The value of the expression is compared with the values of each case ,
  If there is a match, the associated block of code is executed.

C++ SWITCH STATEMENTS " CONT"



C++ BREAK STATEMENT

- When C++ reaches a break keyword, it breaks out of the switch block.
- A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

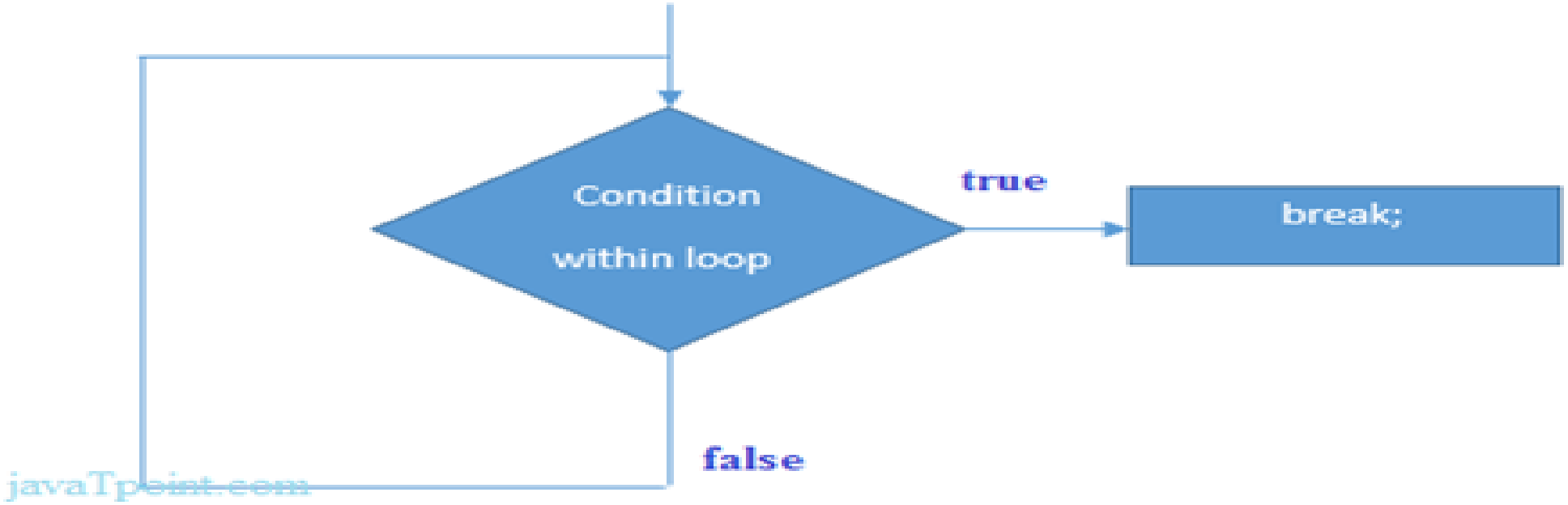
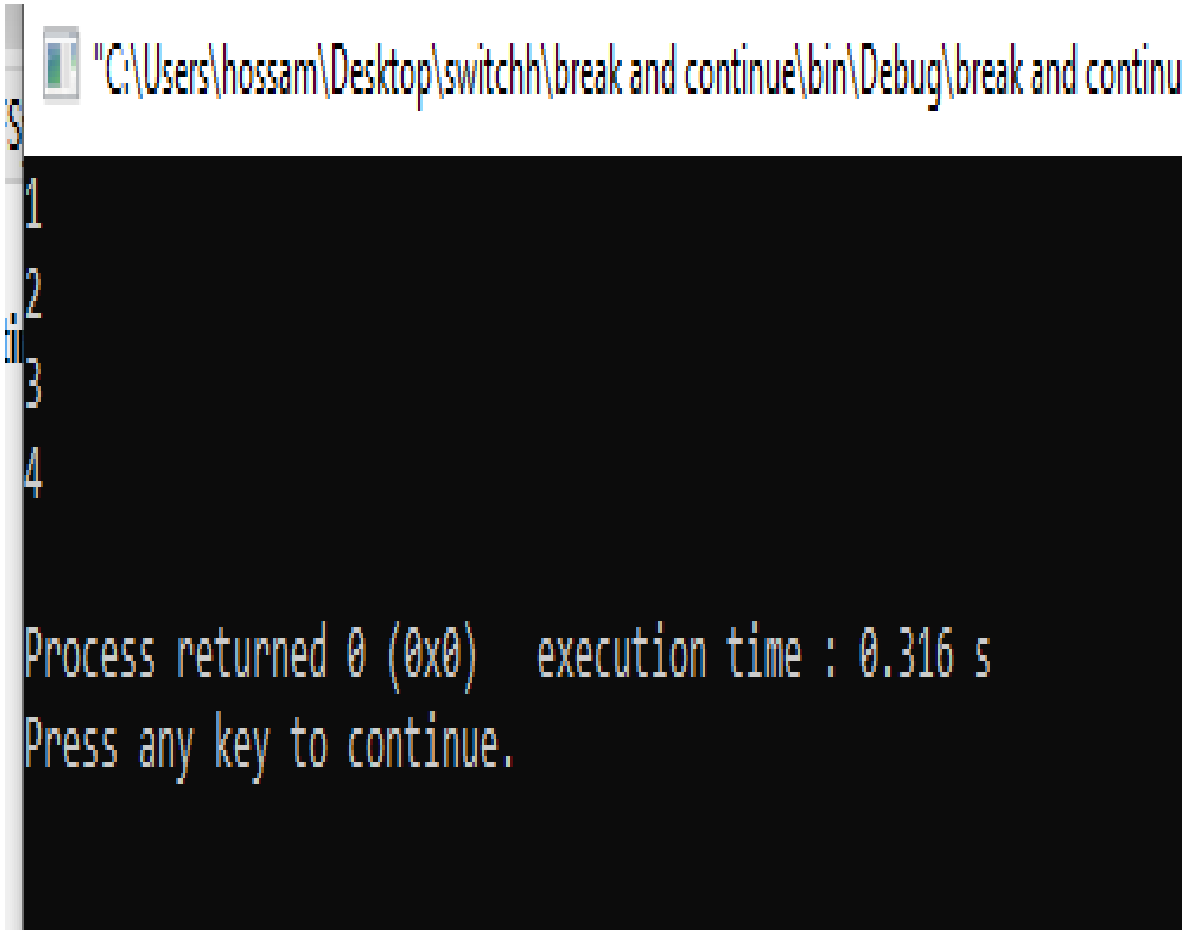


Figure: Flowchart of break statement

C++ BREAK STATEMENT" CONT"

- what is the output of this code ?

```
3   #include <iostream>
4   using namespace std;
5   int main() {
6         for (int i = 1; i <= 10; i++)
7             {
8                 if (i == 5)
9                 {
10                    break;
11                }
12            cout<<i<<"\n";
13            }
14  }
```



C++ CONTINUE STATEMENT

- C++ continue statement is used to continue loop.
- It continues the current flow of the program and skips the remaining code at specified condition.

```
34  //continue examples
35  #include <iostream>
36  using namespace std;
37  int main()
38  {
39      for(int i=1;i<=10;i++){
40          if(i==5){
41              continue;
42          }
43          cout<<i<<"\n";
44      }
45  }
```

```
"C:\Users\hossam\Desktop\switchh\break and continue\bin\Debug\break
1
2
3
4
6
7
8
9
10

Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```

THANKS

Dr/Ghada Maher
Eng/ Hossam Medhat