C++

Prepared by: Mohamed Ayman

facebook.com/sw.eng.MohamedAyman

sw.eng.MohamedAyman@gmail.com

wuzzuf.net/me/engMohamedAyman

codeforces.com/profile/Mohamed_Ayman

Conditions

Outline

1) Decision making Definition
2) IF Statement
3) IF , ElSE Statements
4) IF , ELSE IF , ELSE Statements
5) Nested IF Statements
6) Single Statement Suites

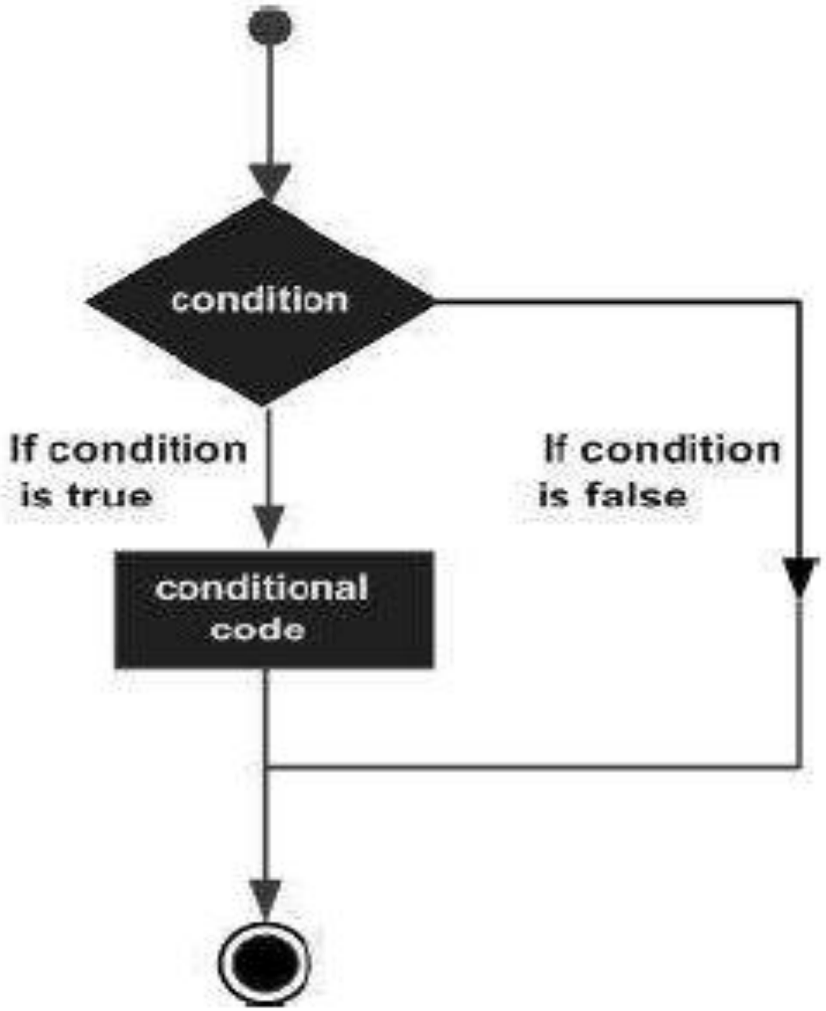
Decision Making Definition

- Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.

- Decision making structure evaluate multiple expressions, which produce TRUE or FALSE as the outcome. You need to determine which action to take and which statement to execute if the outcome is TRUE or FALSE otherwise.

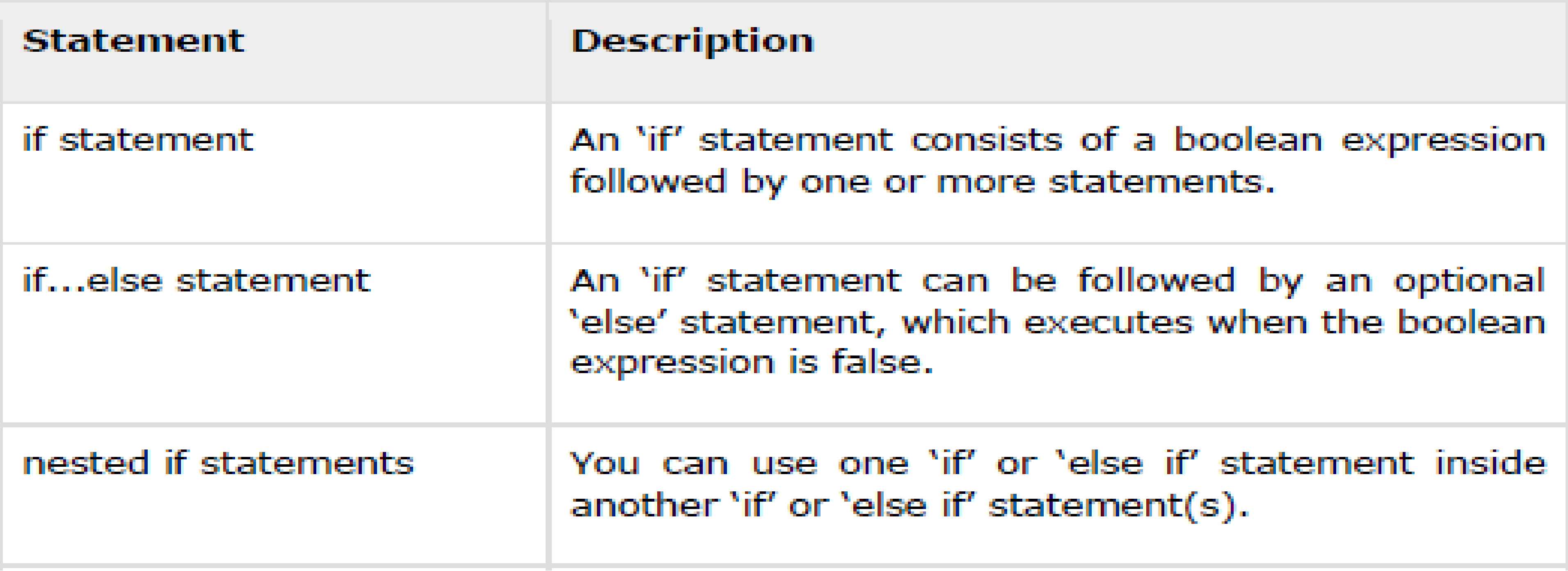
Decision Making Definition

- The general form of a typical decision making structure found in most of the programming language.



Decision Making Definition

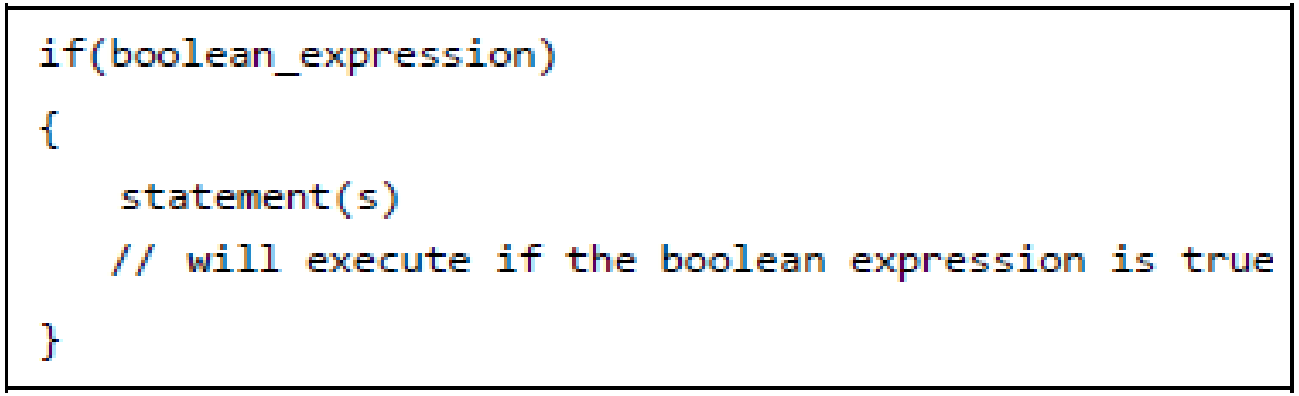
- C++ programming language assume any non-zero and non-null values as TRUE, and any zero or null values as FALSE values.
- C++ programming language provides the following types of decision-making statement



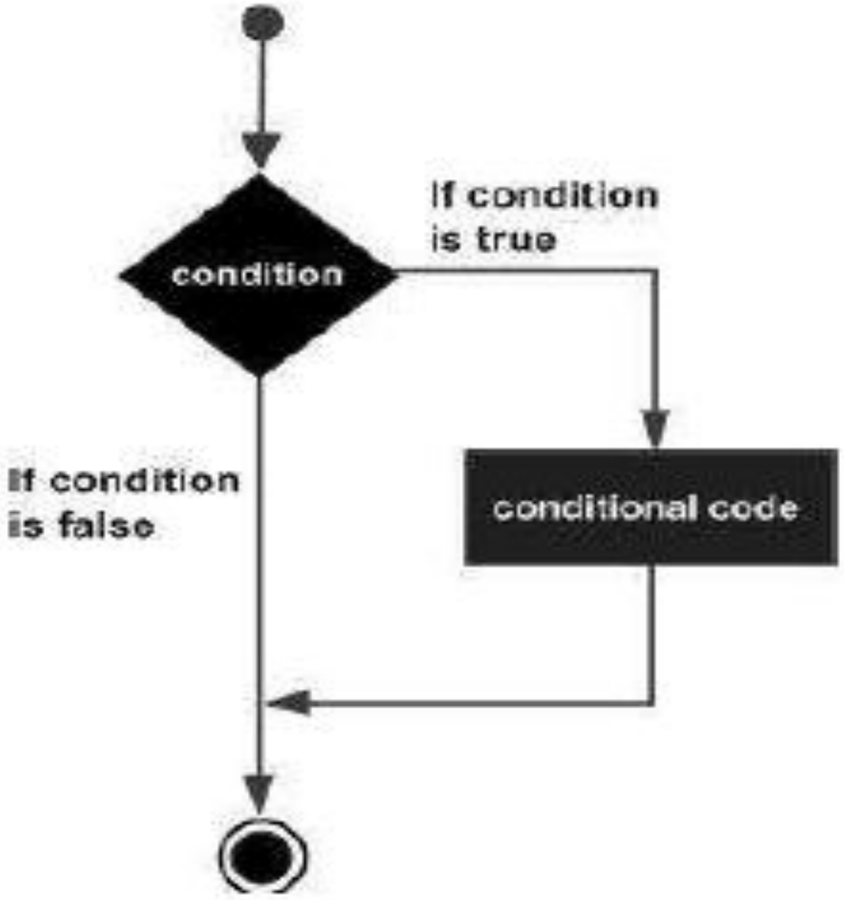
Statement	Description
if statement	An 'if' statement consists of a boolean expression followed by one or more statements.
if...else statement	An 'if' statement can be followed by an optional 'else' statement, which executes when the boolean expression is false.
nested if statements	You can use one 'if' or 'else if' statement inside another 'if' or 'else if' statement(s).

IF Statement

- The IF statement is similar to that of other language. The if statement contains a logical expression using which the data is compared and a decision is made based on the result of the comparison.

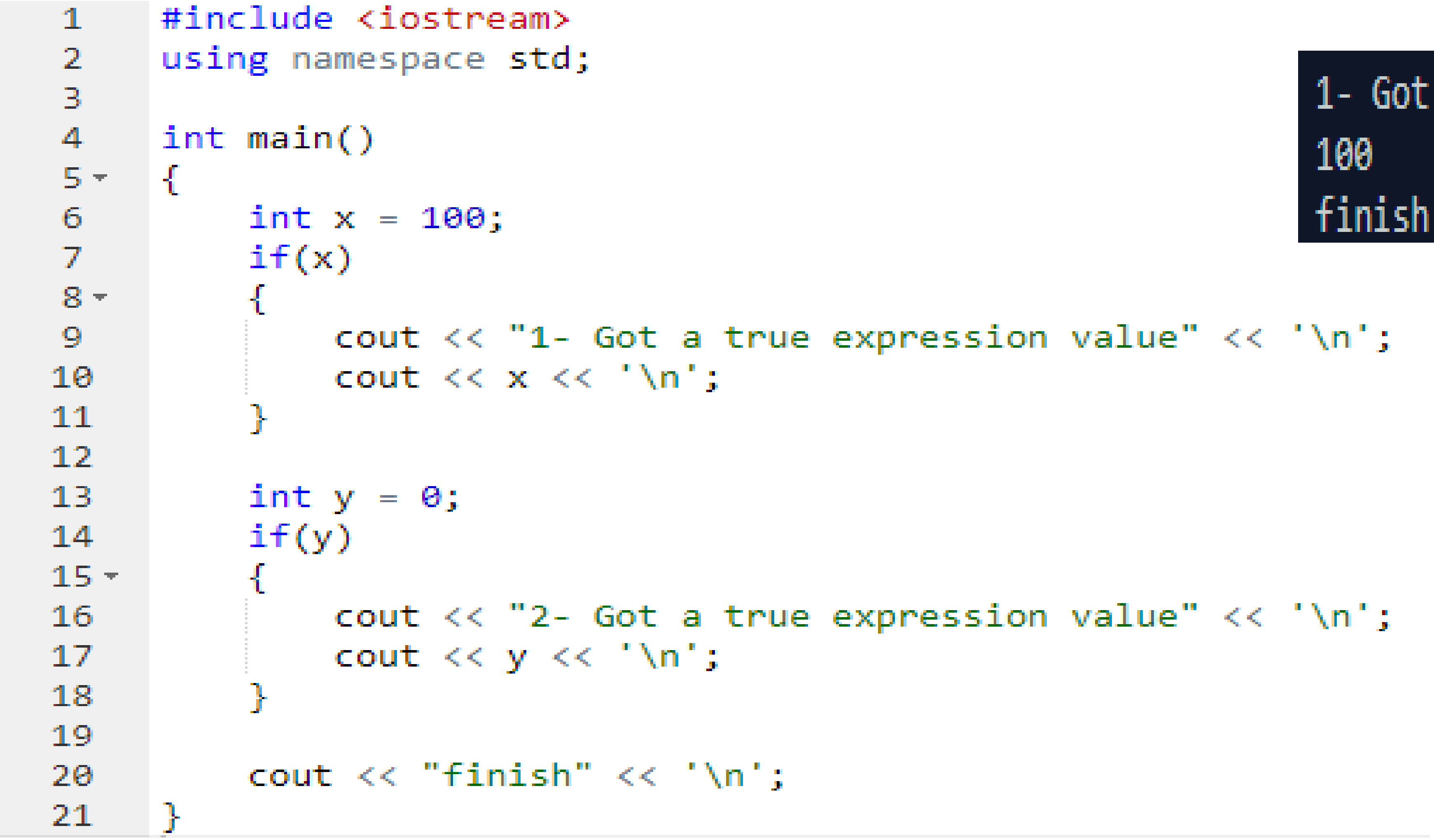


```
if(boolean_expression)
{
    statement(s)
   // will execute if the boolean expression is true
}
```

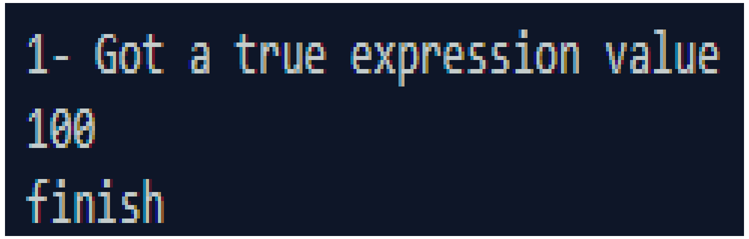


IF Statement Example

- Source code: https://repl.it/repls/GreenyellowCumbersomeAmurminnow



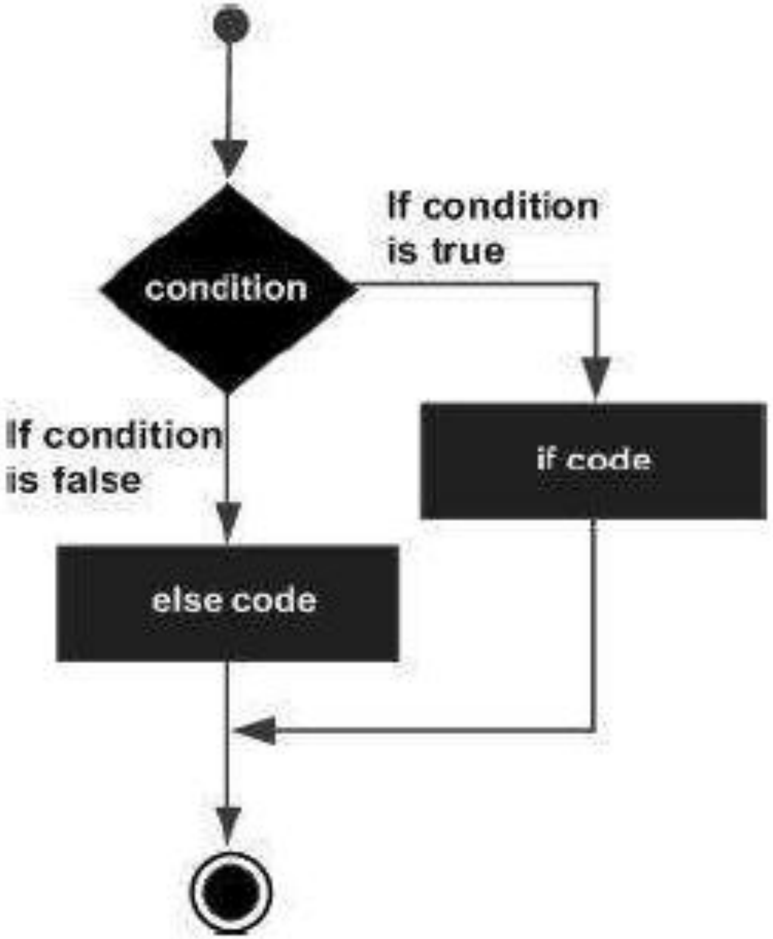
```cpp
#include <iostream>
using namespace std;

int main()
{
    int x = 100;
    if(x)
    {
        cout << "1- Got a true expression value" << '\n';
        cout << x << '\n';
    }

    int y = 0;
    if(y)
    {
        cout << "2- Got a true expression value" << '\n';
        cout << y << '\n';
    }

    cout << "finish" << '\n';
}
```



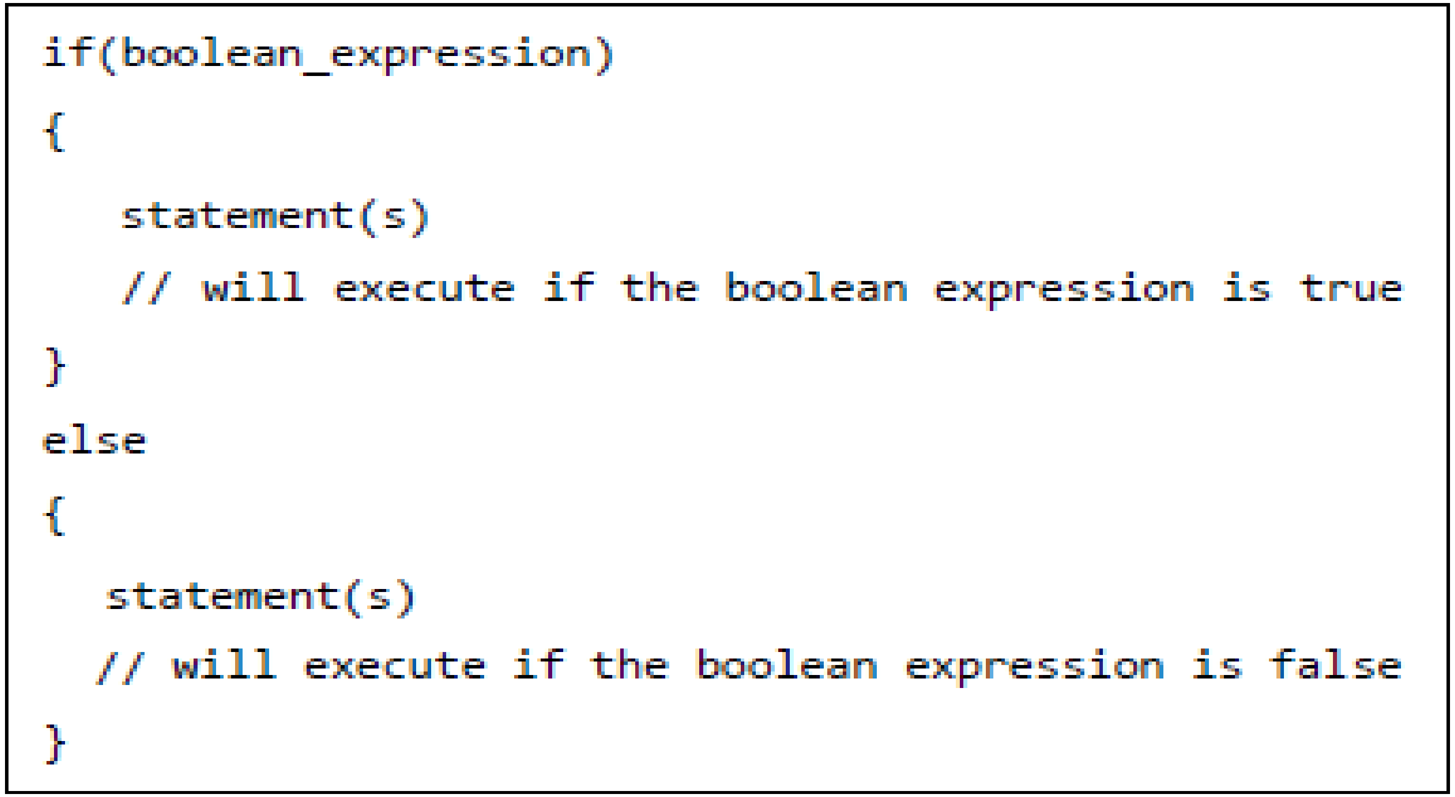
```
1- Got a true expression value
100
finish
```

IF , ELSE Statements

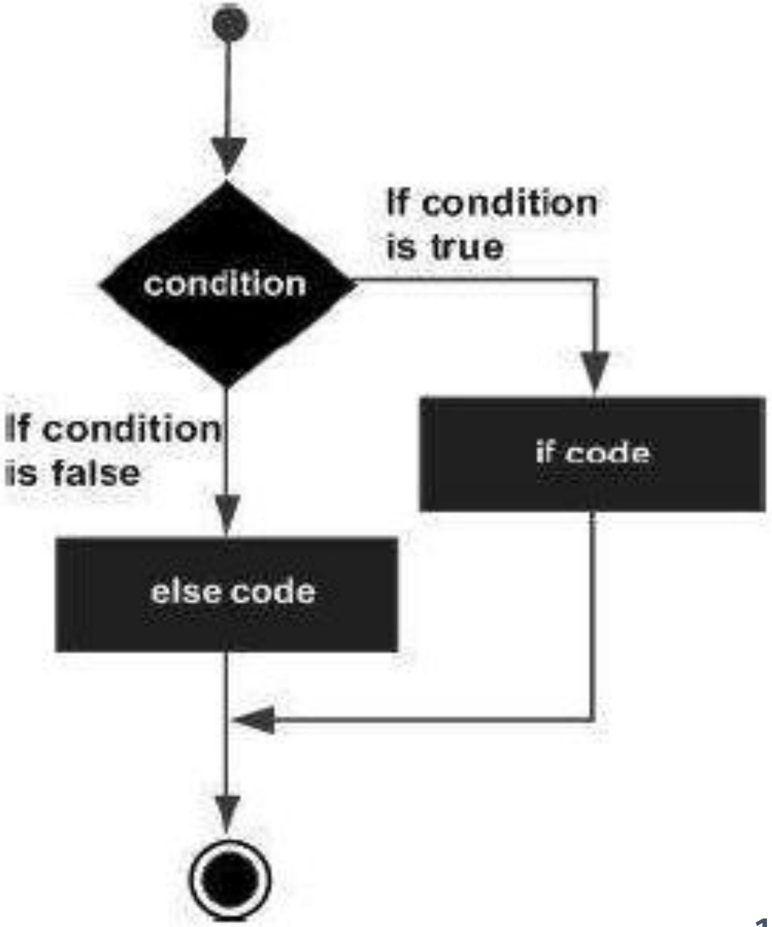
- An else statement can be combined with an if statement. An else statement contains a blocks of code that executes if the conditional expression in the if statement resolve to 0 or a FALSE value.
- The else statement is an optional statement and there could be at most only one else statement following if.



IF , ELSE Statements

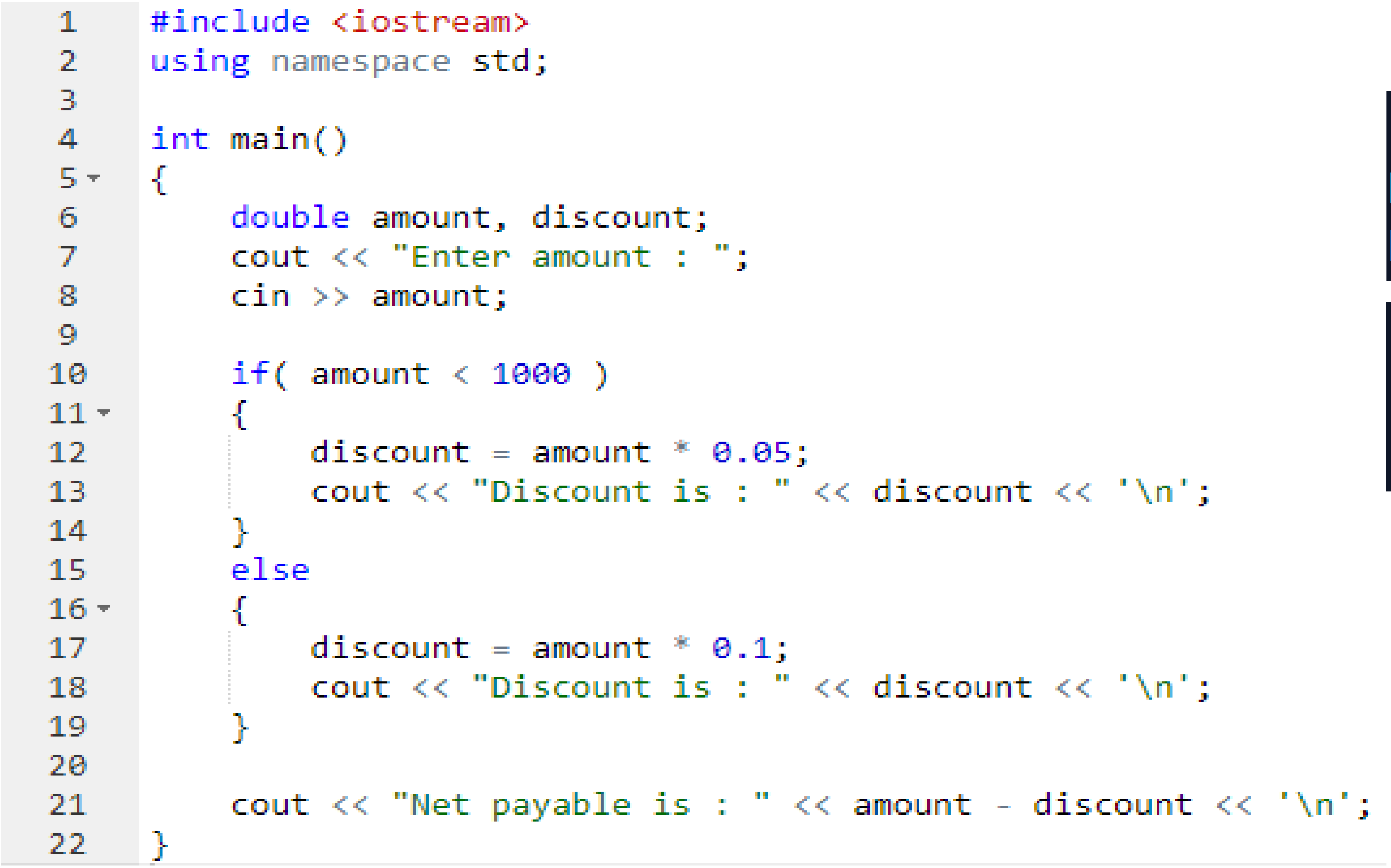


```
if(boolean_expression)
{
    statement(s)
    // will execute if the boolean expression is true
}
else
{
   statement(s)
  // will execute if the boolean expression is false
}
```

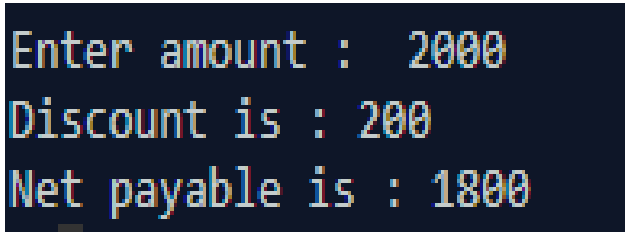


IF , ELSE Statements Example

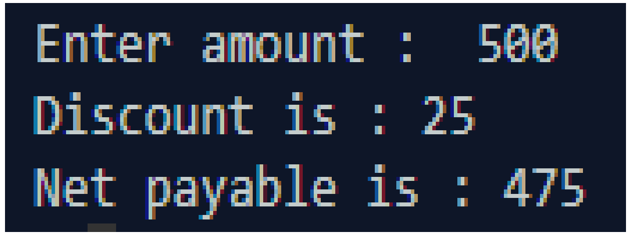
- Source code: https://repl.it/repls/PrudentNotedSloth



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

int main()
{
    double amount, discount;
    cout << "Enter amount : ";
    cin >> amount;

    if( amount < 1000 )
    {
        discount = amount * 0.05;
        cout << "Discount is : " << discount << '\n';
    }
    else
    {
        discount = amount * 0.1;
        cout << "Discount is : " << discount << '\n';
    }

    cout << "Net payable is : " << amount - discount << '\n';
}
```



```
Enter amount :  2000
Discount is : 200
Net payable is : 1800
```



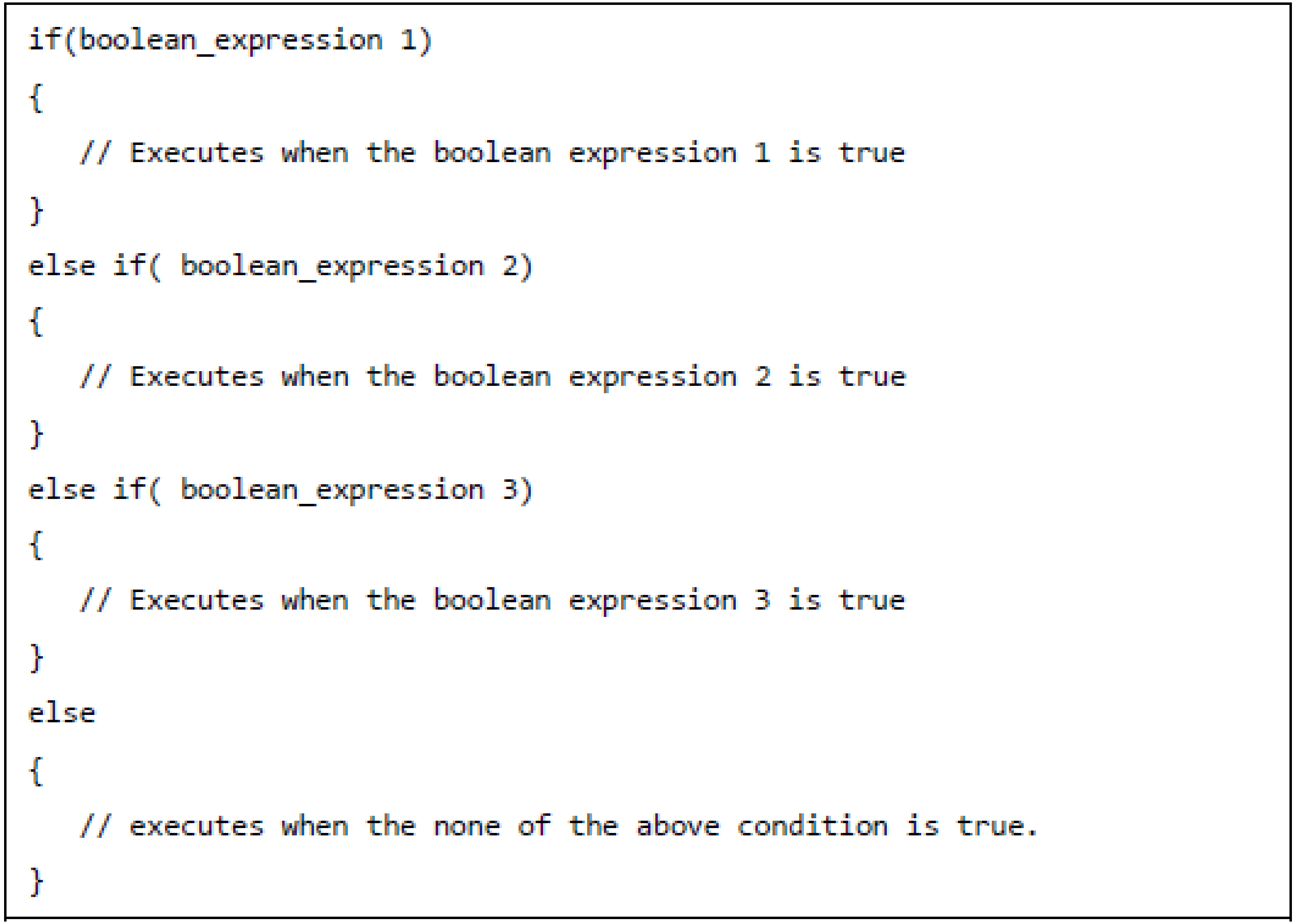
```
Enter amount :  500
Discount is : 25
Net payable is : 475
```

IF , ELIF , ELSE Statements

- The else if statement allows you to check multiple expression for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE

- Similar to the else, the else if statement is optional. However, unlike else, for which there can be at the most one statement, there can be an arbitrary number of else if statements following an if.

IF , ELIF , ELSE Statements

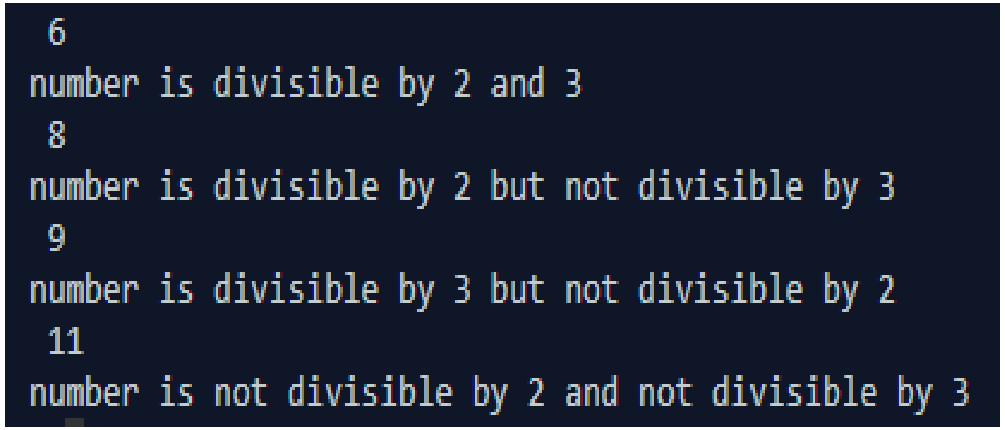


```
if(boolean_expression 1)
{
   // Executes when the boolean expression 1 is true
}
else if( boolean_expression 2)
{
   // Executes when the boolean expression 2 is true
}
else if( boolean_expression 3)
{
   // Executes when the boolean expression 3 is true
}
else
{
   // executes when the none of the above condition is true.
}
```

IF , ELIF , ELSE Statements Example

- Source code: https://repl.it/repls/HiddenLawngreenPolecat

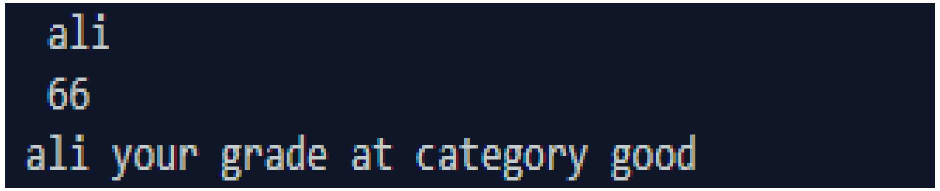
```cpp
#include <iostream>
using namespace std;

int main()
{
    int num;
    cin >> num;

    if( num % 2 == 0 && num % 3 == 0 )
    {
        cout << "number is divisible by 2 and 3\n";
    }
    else if( num % 2 == 0 )
    {
        cout << "number is divisible by 2 but not divisible by 3\n";
    }
    else if( num % 3 == 0 )
    {
        cout << "number is divisible by 3 but not divisible by 2\n";
    }
    else
    {
        cout << "number is not divisible by 2 and not divisible by 3\n";
    }
}
```



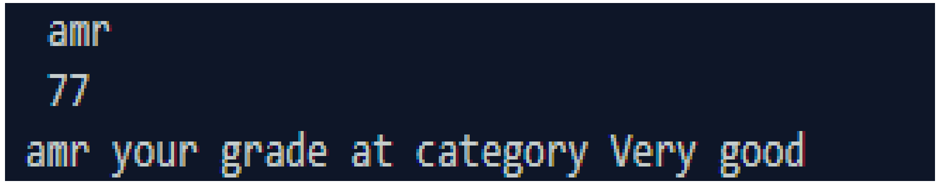
```
 6
number is divisible by 2 and 3
 8
number is divisible by 2 but not divisible by 3
 9
number is divisible by 3 but not divisible by 2
 11
number is not divisible by 2 and not divisible by 3
```

Practice

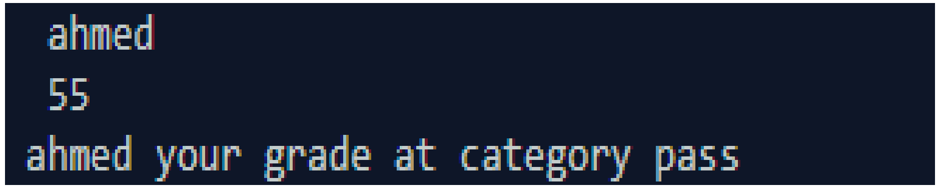
- Take as an input name and x which represent his/her grade and determine which category this grade belong to.
    - Greater than or equal 85 is Excellent
    - Greater than or equal 75 is very good
    - Greater than or equal 65 is good
    - Greater than or equal 50 is pass
    - Less than 50 is fail
- Test Cases:



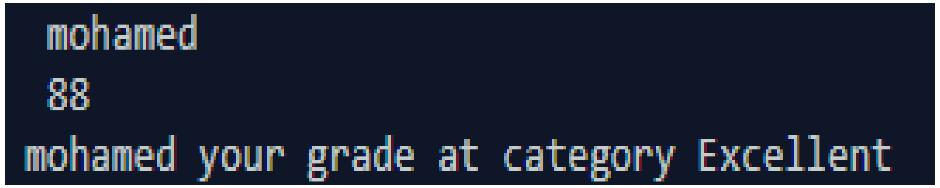
```
 ali
 66
ali your grade at category good
```



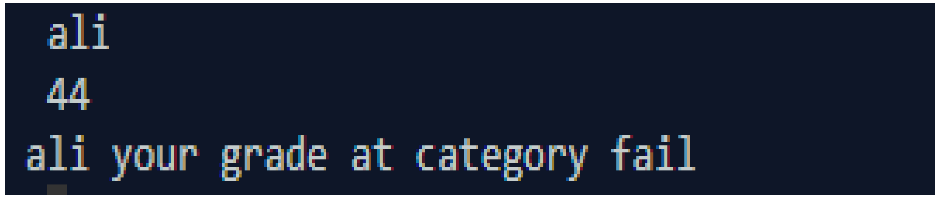
```
 amr
 77
amr your grade at category Very good
```



```
 ahmed
 55
ahmed your grade at category pass
```



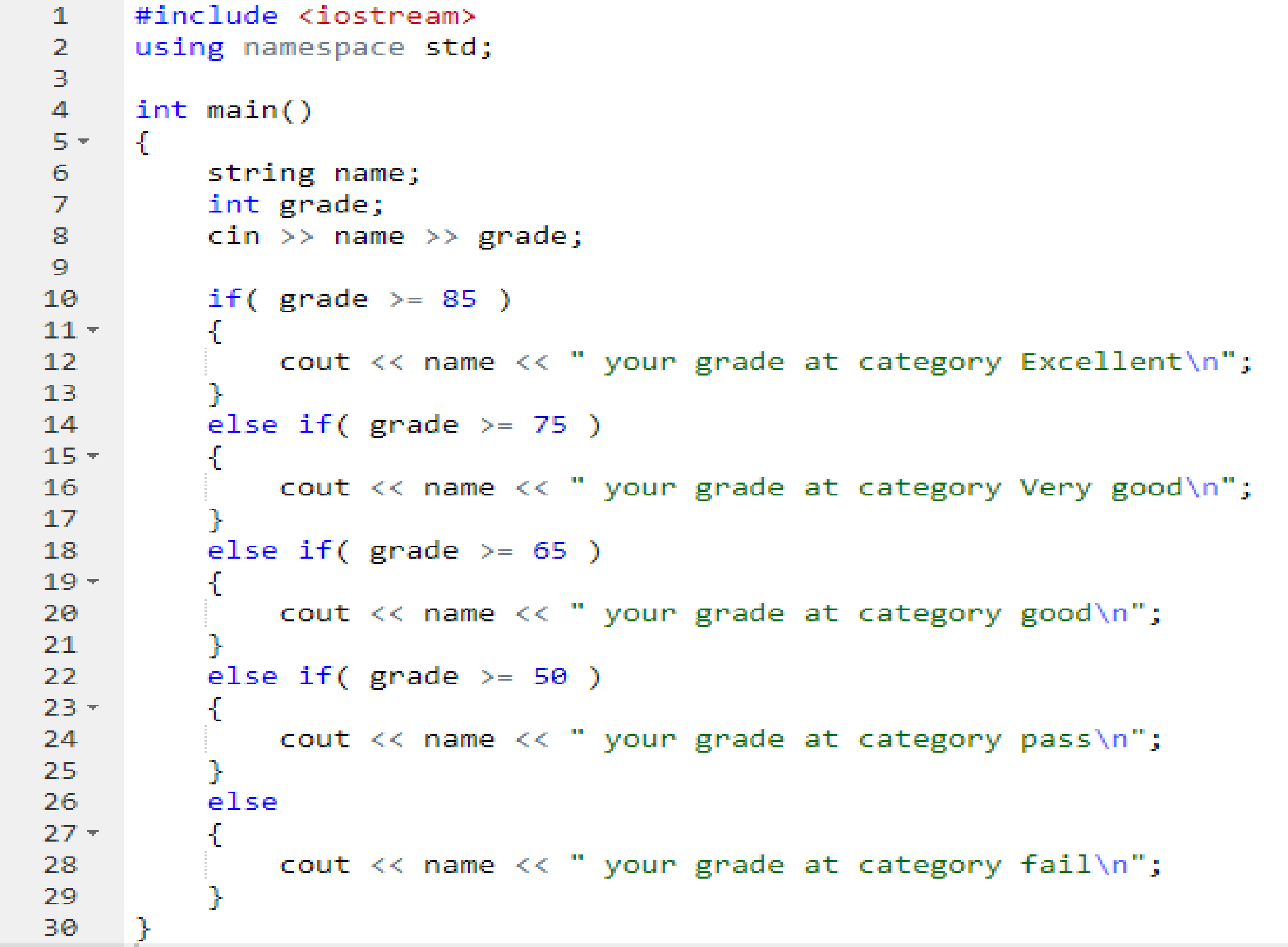
```
 mohamed
 88
mohamed your grade at category Excellent
```



```
 ali
 44
ali your grade at category fail
```

Practice Solution

- Source code: https://repl.it/repls/ClearcutClearGnu



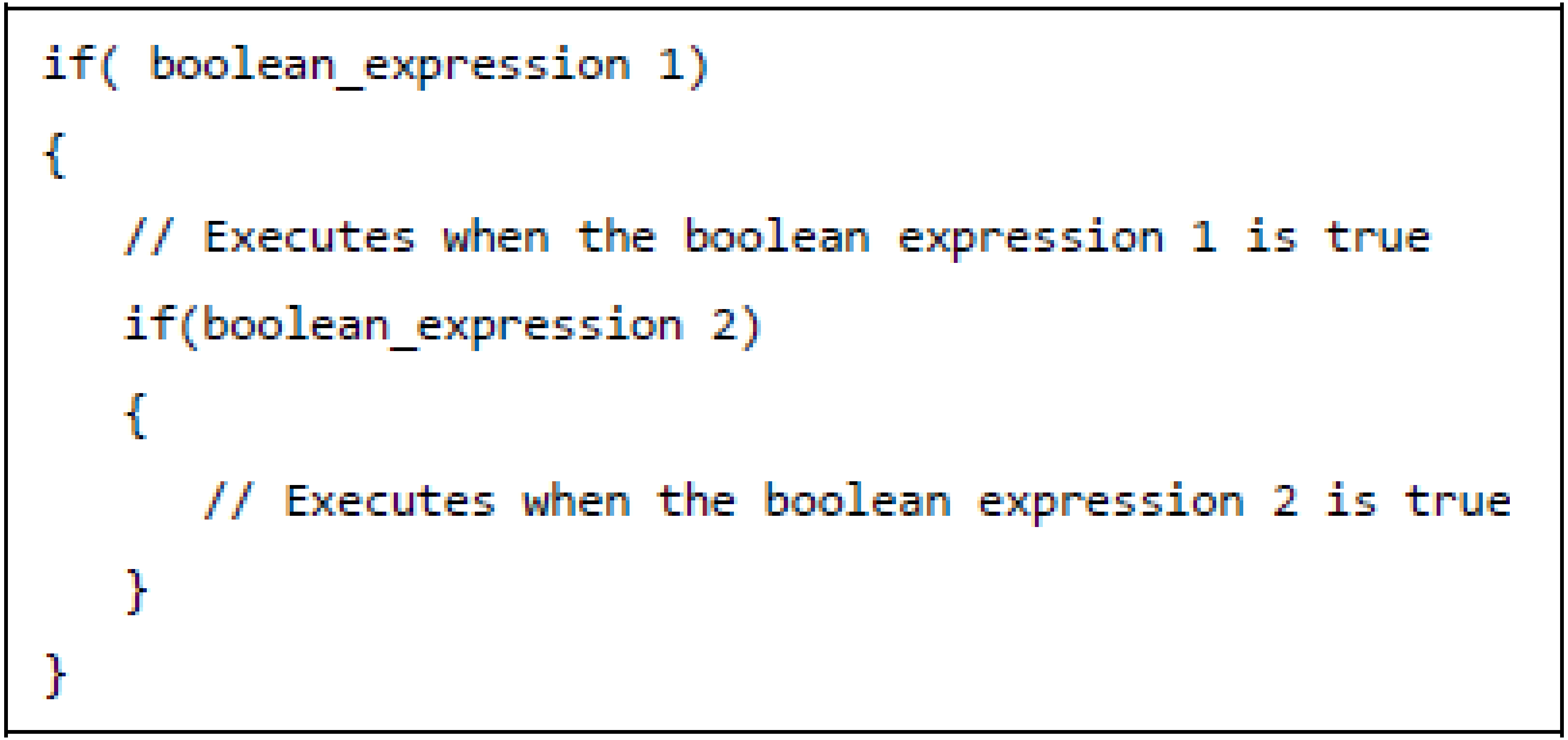
```cpp
#include <iostream>
using namespace std;

int main()
{
    string name;
    int grade;
    cin >> name >> grade;

    if( grade >= 85 )
    {
        cout << name << " your grade at category Excellent\n";
    }
    else if( grade >= 75 )
    {
        cout << name << " your grade at category Very good\n";
    }
    else if( grade >= 65 )
    {
        cout << name << " your grade at category good\n";
    }
    else if( grade >= 50 )
    {
        cout << name << " your grade at category pass\n";
    }
    else
    {
        cout << name << " your grade at category fail\n";
    }
}
```

Nested IF Statements

- There may be a situation when you want to check for another condition after a condition resolve to true. In such a situation, you can use the nested if construct.

- In a nested if construct, you can have an if ... else if ... else construct inside another if ... else if ... else construct.

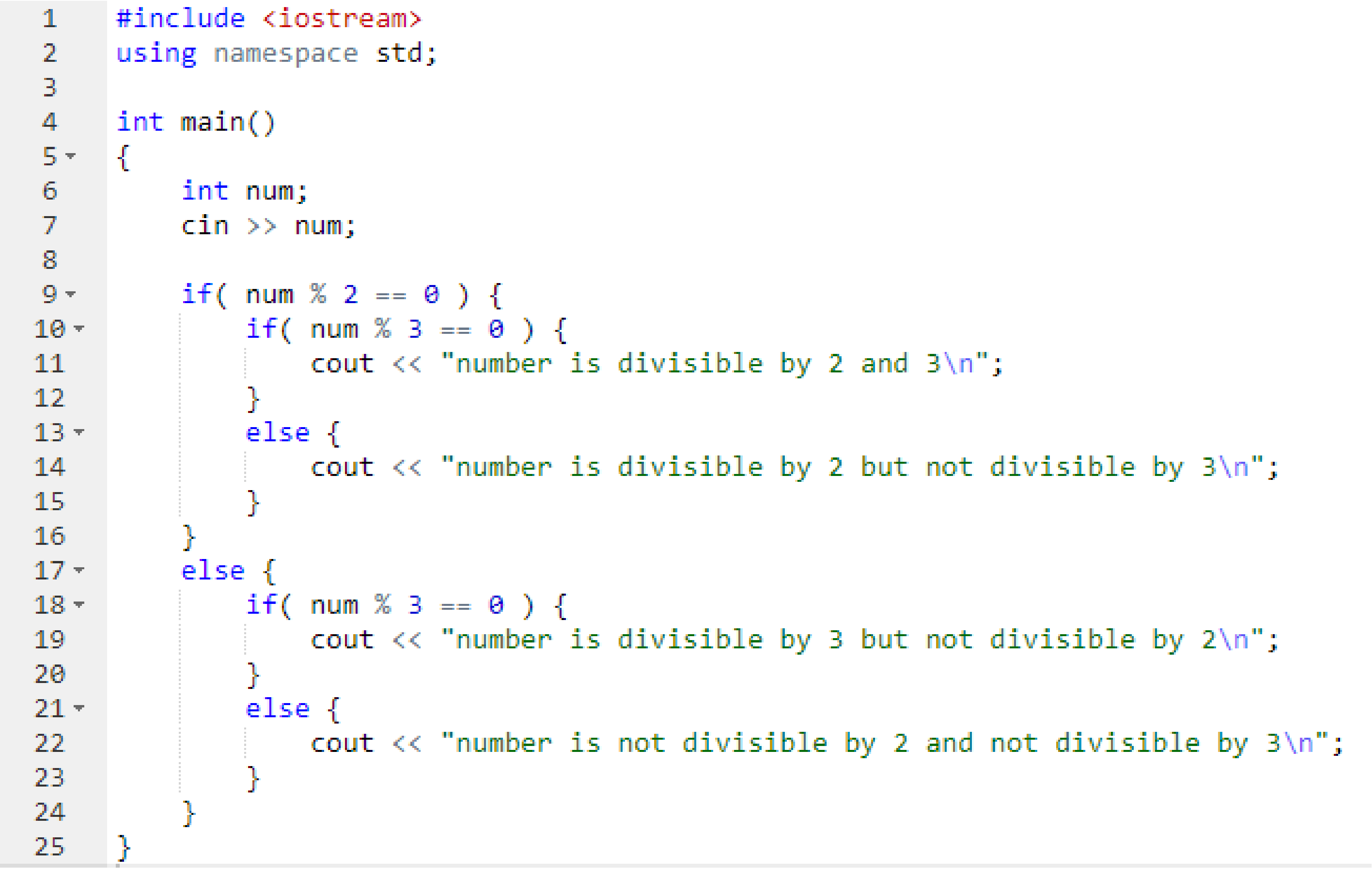
Nested IF Statements



```
if( boolean_expression 1)
{
   // Executes when the boolean expression 1 is true
   if(boolean_expression 2)
   {
      // Executes when the boolean expression 2 is true
   }
}
```

Nested IF Statements Example

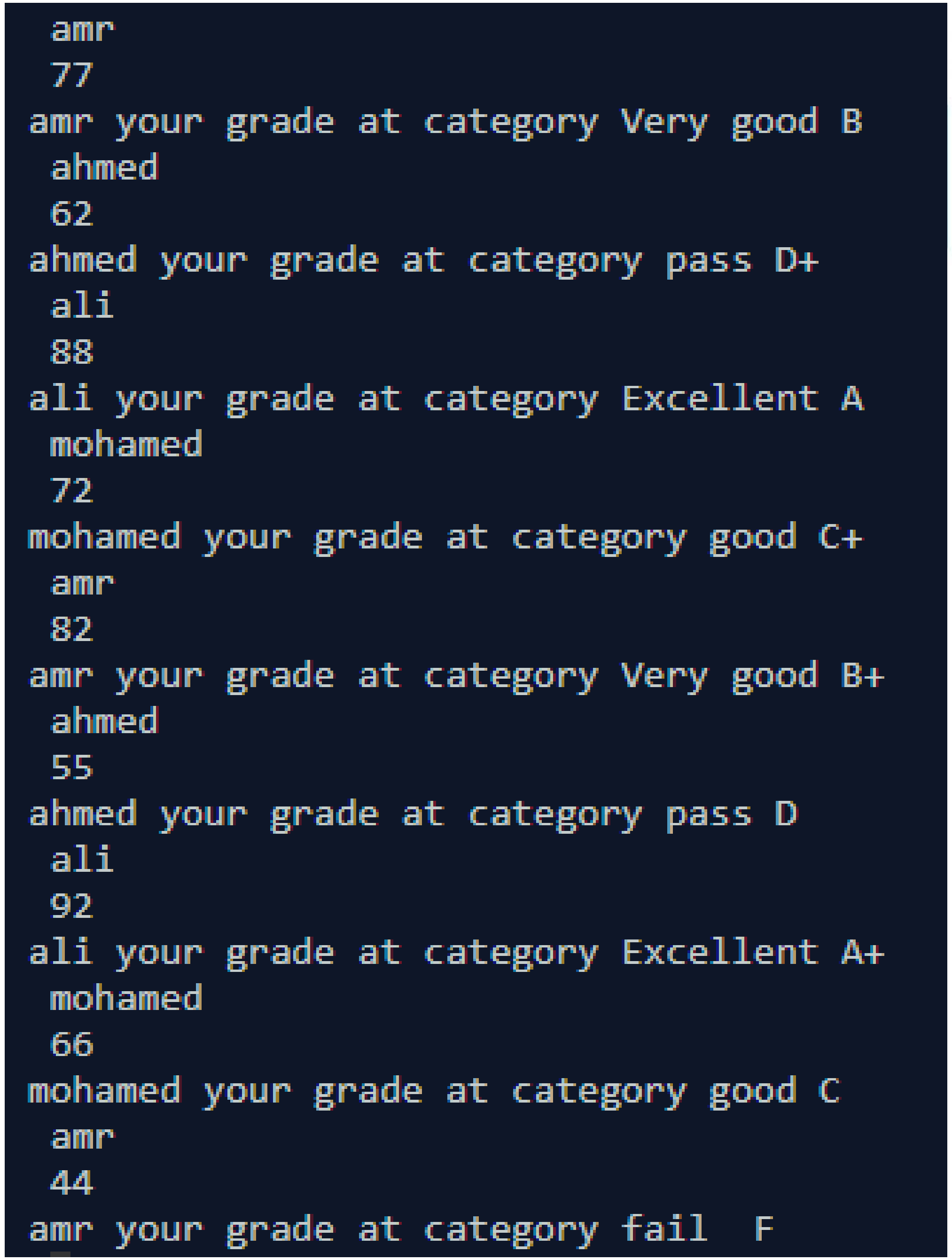
- Source code: https://repl.it/repls/SecondHumiliatingWuerhosaurus



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

int main()
{
    int num;
    cin >> num;

    if( num % 2 == 0 ) {
        if( num % 3 == 0 ) {
            cout << "number is divisible by 2 and 3\n";
        }
        else {
            cout << "number is divisible by 2 but not divisible by 3\n";
        }
    }
    else {
        if( num % 3 == 0 ) {
            cout << "number is divisible by 3 but not divisible by 2\n";
        }
        else {
            cout << "number is not divisible by 2 and not divisible by 3\n";
        }
    }
}
```

Practice

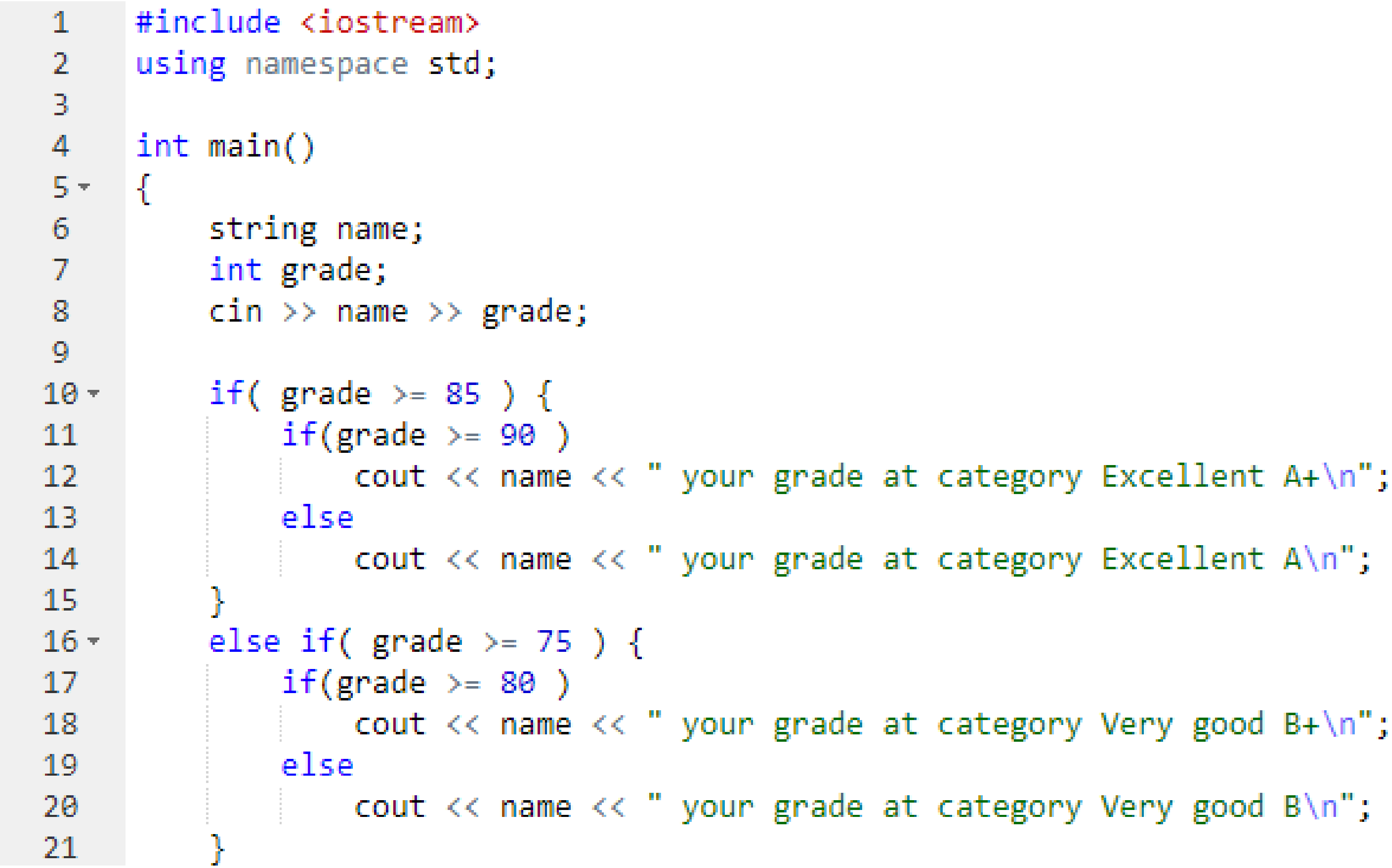
- Take as an input name and x which represent his/her grade and determine which category this grade belong to and his/her subcategory.
  - Greater than or equal 85 is Excellent
    - A+ from 90,   A from 85
  - Greater than or equal 75 is very good
    - B+ from 80,   B from 75
  - Greater than or equal 65 is good
    - C+ from 70,   C from 65
  - Greater than or equal 50 is pass
    - D+ from 60,   D from 50
  - Less than 50 is fail   => F
- Test Cases:



```
 amr
 77
amr your grade at category Very good B
 ahmed
 62
ahmed your grade at category pass D+
 ali
 88
ali your grade at category Excellent A
 mohamed
 72
mohamed your grade at category good C+
 amr
 82
amr your grade at category Very good B+
 ahmed
 55
ahmed your grade at category pass D
 ali
 92
ali your grade at category Excellent A+
 mohamed
 66
mohamed your grade at category good C
 amr
 44
amr your grade at category fail  F
```

Practice Solution

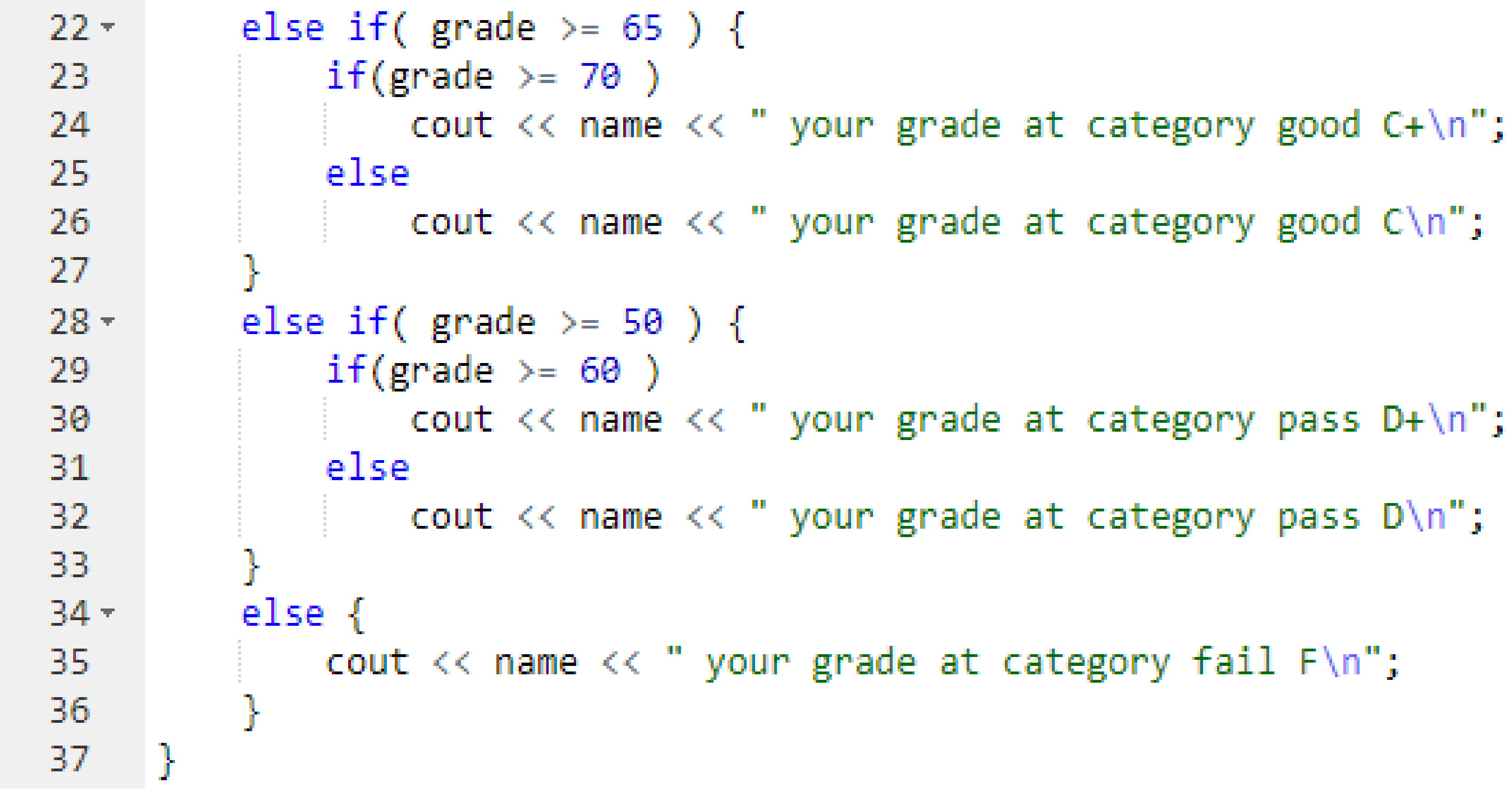
- Source code: https://repl.it/repls/ThistleBogusIcefish



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

int main()
{
    string name;
    int grade;
    cin >> name >> grade;

    if( grade >= 85 ) {
        if(grade >= 90 )
            cout << name << " your grade at category Excellent A+\n";
        else
            cout << name << " your grade at category Excellent A\n";
    }
    else if( grade >= 75 ) {
        if(grade >= 80 )
            cout << name << " your grade at category Very good B+\n";
        else
            cout << name << " your grade at category Very good B\n";
    }
```

Practice Solution

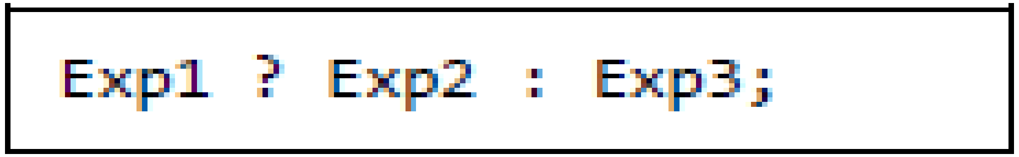
- Source code: https://repl.it/repls/ThistleBogusIcefish



```cpp
    else if( grade >= 65 ) {
        if(grade >= 70 )
            cout << name << " your grade at category good C+\n";
        else
            cout << name << " your grade at category good C\n";
    }
    else if( grade >= 50 ) {
        if(grade >= 60 )
            cout << name << " your grade at category pass D+\n";
        else
            cout << name << " your grade at category pass D\n";
    }
    else {
        cout << name << " your grade at category fail F\n";
    }
}
```

The ? : Operator

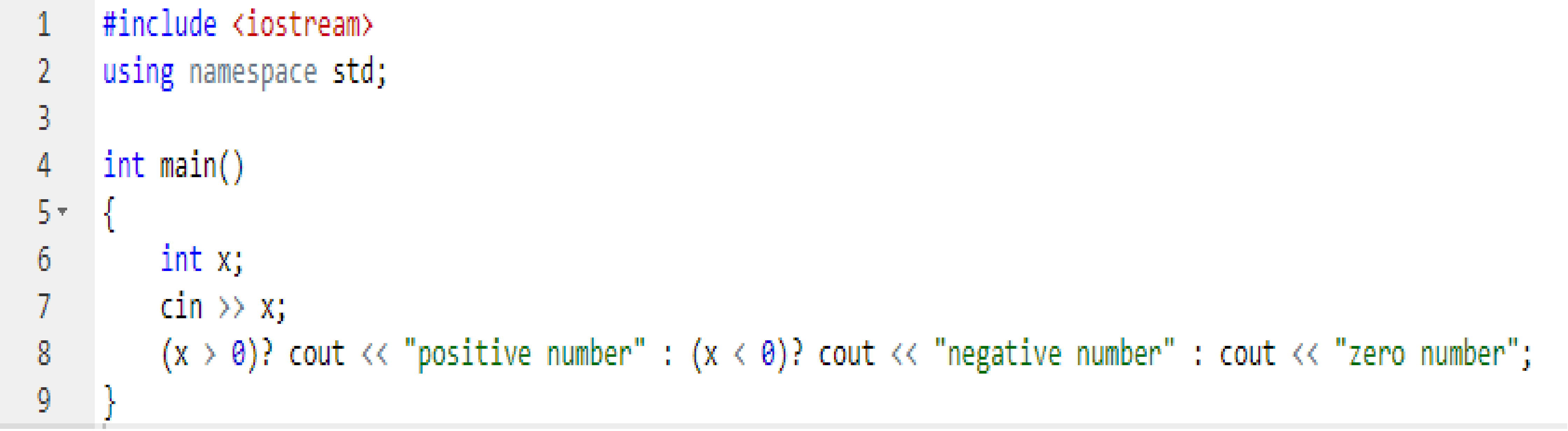
- The value of a ‘?’ expression is determined like this: Exp1 is evaluated. If it is true, then Exp2 is evaluated and becomes the value of the entire ‘?’ expression. If Exp1 is false, then Exp3 is evaluated and its value becomes the value of the expression.



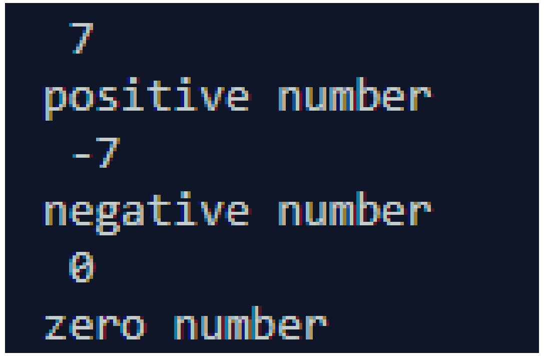
```
Exp1 ? Exp2 : Exp3;
```

The ? : Operator Example

- Source code: https://repl.it/repls/IntelligentGlitteringXeme



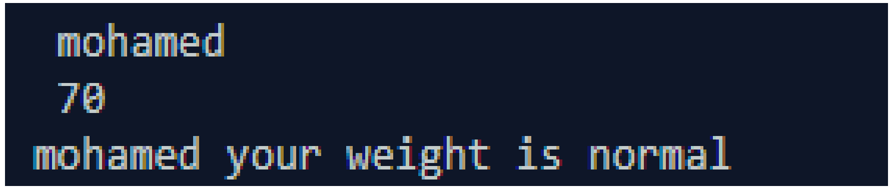
```cpp
#include <iostream>
using namespace std;

int main()
{
    int x;
    cin >> x;
    (x > 0)? cout << "positive number" : (x < 0)? cout << "negative number" : cout << "zero number";
}
```



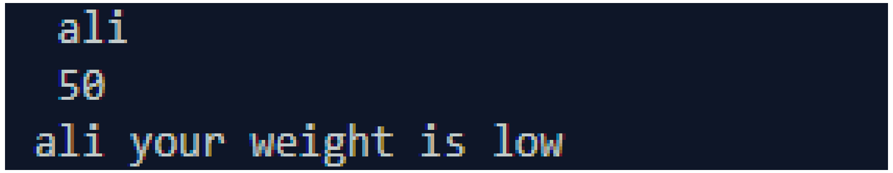
```
 7
positive number
 -7
negative number
 0
zero number
```

Practice

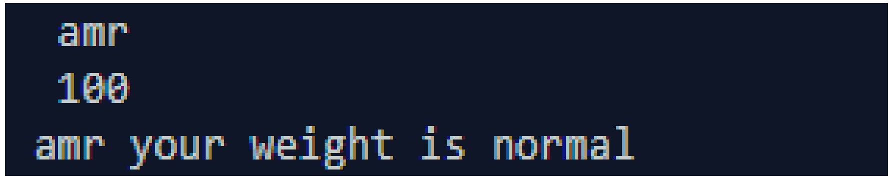
- Take as an input name and x which represent his/her weight and determine this weight high or low or normal
- When weight greater than 100 this mean high and when weight less than or equal 50 this mean low else of that mean normal
- Test Cases:



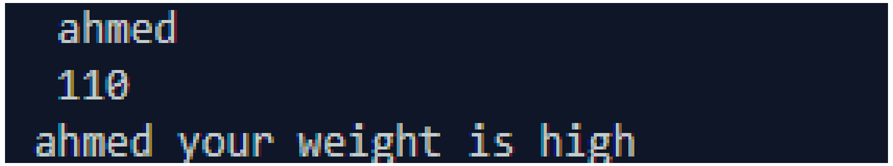
```
 mohamed
 70
mohamed your weight is normal
```



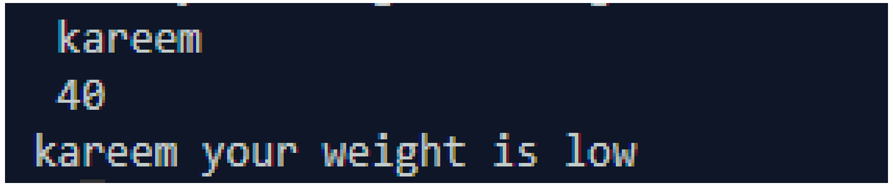
```
 ali
 50
ali your weight is low
```



```
 amr
 100
amr your weight is normal
```



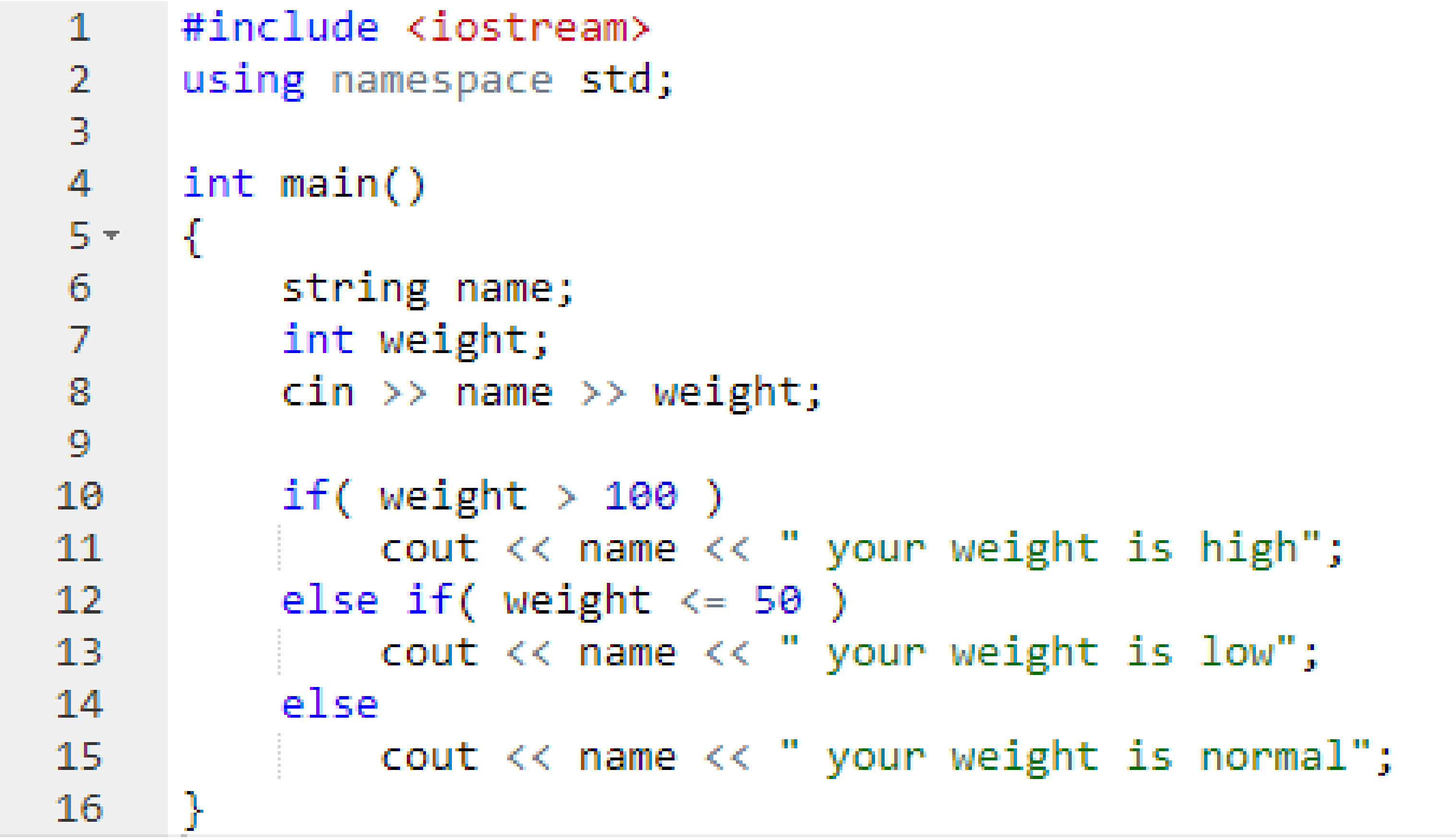
```
 ahmed
 110
ahmed your weight is high
```



```
 kareem
 40
kareem your weight is low
```

Practice Solution

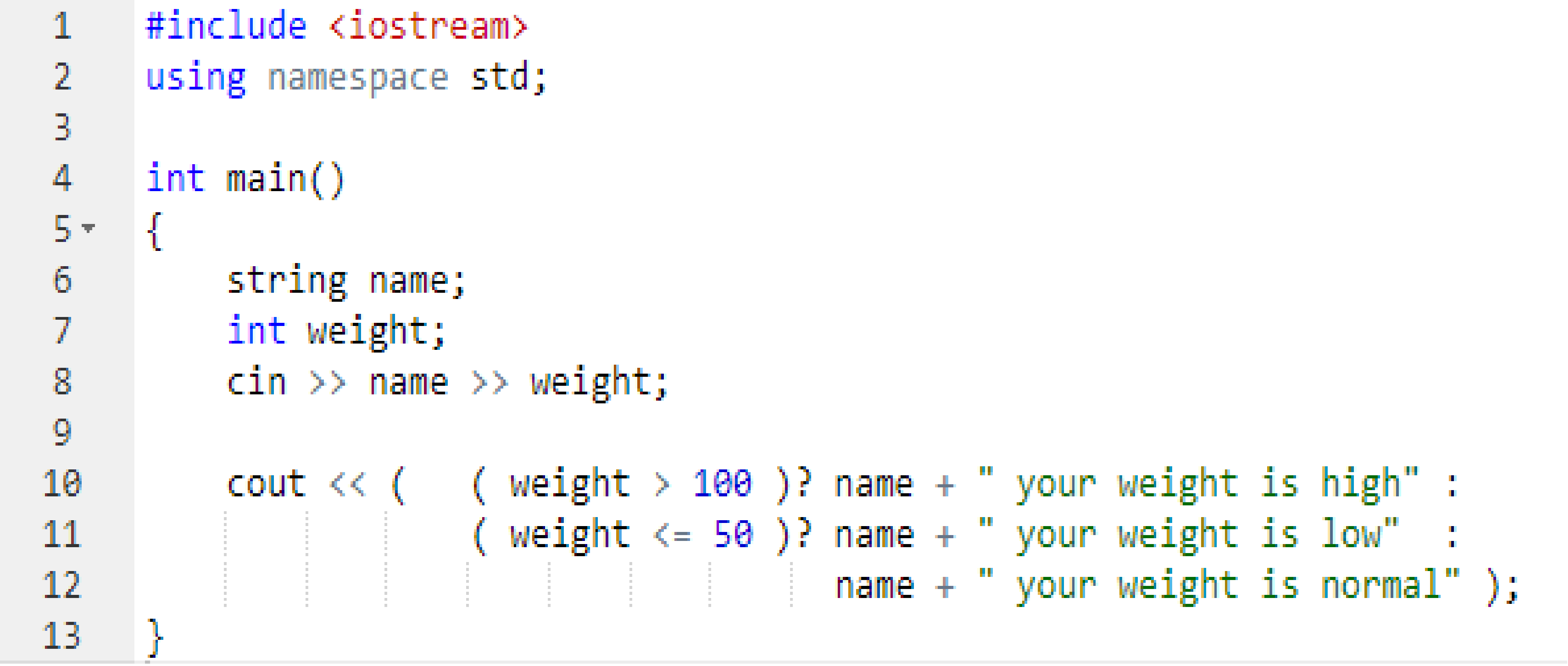
- Source code: https://repl.it/repls/GiganticGiftedBaldeagle



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

int main()
{
    string name;
    int weight;
    cin >> name >> weight;

    if( weight > 100 )
        cout << name << " your weight is high";
    else if( weight <= 50 )
        cout << name << " your weight is low";
    else
        cout << name << " your weight is normal";
}
```

Practice Solution

- Source code: https://repl.it/repls/ClumsyGreenyellowTick



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

int main()
{
    string name;
    int weight;
    cin >> name >> weight;

    cout << (   ( weight > 100 )? name + " your weight is high" :
                ( weight <= 50 )? name + " your weight is low"  :
                                  name + " your weight is normal" );
}
```

Questions ?

References

Online Courses YouTube playlists:	http://bit.ly/2kAPL5K
C++ Documentation	http://bit.ly/1flmcHO
CPP For School	http://bit.ly/2kifMdj
C++ Language Tutorial	http://bit.ly/1kyBMdz
C++ Language Tutorial	http://bit.ly/2rzE4hQ
C++ Tutorial Point	http://bit.ly/2BGFeO0
Fundamentals of C++ Programming	http://bit.ly/2rJHhyl
Teach Yourself C++ in 21 Days	http://bit.ly/1JXhDtL
A Complete Guide to Programming in C++	http://bit.ly/2dVkGY9