Lecture 6

Conditional Flows

UBS
ITU INFORMATION TECHNOLOGY UNIVERSITY

QUIZ

سورة طه پارہ: 16

قَالَ رَبِّ اشْرَحْ لِيْ صَدْرِيْ ۙ ﴿٢٥﴾

[قَالَ: اس نے کہا] [رَبِّ: اے رب !] [اشْرَحْ: کھول دے] [لِيْ: میرے لیے] [صَدْرِيْ : میرا سینہ]

وَيَسِّرْ لِيْٓ اَمْرِيْ ۙ ﴿٢٦﴾

[وَيَسِّرْ: اور آسان کر دے] [لِيْ: میرے لیے] [اَمْرِيْ : میرا کام]

وَاحْلُلْ عُقْدَةً مِّنْ لِّسَانِيْ ۙ ﴿٢٧﴾

[وَاحْلُلْ: اور کھول دے] [عُقْدَةً: گرہ] [مِّنْ: سے] [لِّسَانِيْ: میری زبان]

يَفْقَهُوْا قَوْلِيْ ۖ ﴿٢٨﴾

[يَفْقَهُوْا: وہ سمجھ سکیں] [قَوْلِيْ : میری بات]

QUESTIONS / FEEDBACK / CONCERNS

UBS

ITU INFORMATION TECHNOLOGY UNIVERSITY

SE SEC A
SLIDE OF FAME

Muhammad Abdullah
BSSE23005
WEEK-1

YOUR NAME
WEEK-2

YOUR NAME
WEEK-3

YOUR NAME
WEEK-4

YOUR NAME
WEEK-5

YOUR NAME
WEEK-6

YOUR NAME
WEEK-7

YOUR NAME
WEEK-8

YOUR NAME
WEEK-9

YOUR NAME
WEEK-10

YOUR NAME
MIDTERM

YOUR NAME
WEEK-11

YOUR NAME
WEEK-12

YOUR NAME
WEEK-13

YOUR NAME
WEEK-14

YOUR NAME
WEEK-15

UBS

ITU INFORMATION TECHNOLOGY UNIVERSITY

SE SEC B
SLIDE OF FAME

Muhammad Mukarram
BSSE23029
WEEK-1

YOUR NAME
WEEK-2

YOUR NAME
WEEK-3

YOUR NAME
WEEK-4

YOUR NAME
WEEK-5

YOUR NAME
WEEK-6

YOUR NAME
WEEK-7

YOUR NAME
WEEK-8

YOUR NAME
WEEK-9

YOUR NAME
WEEK-10

YOUR NAME
MIDTERM

YOUR NAME
WEEK-11

YOUR NAME
WEEK-12

YOUR NAME
WEEK-13

YOUR NAME
WEEK-14

YOUR NAME
WEEK-15

UBS

INFORMATION TECHNOLOGY UNIVERSITY

RECAP

GitHub

Tools (Cygwin, IDE, GitHub)

Flowcharts

Algorithms

Approach towards a word problem

Pseudocode

Flowcharts Advantages & Disadvantages

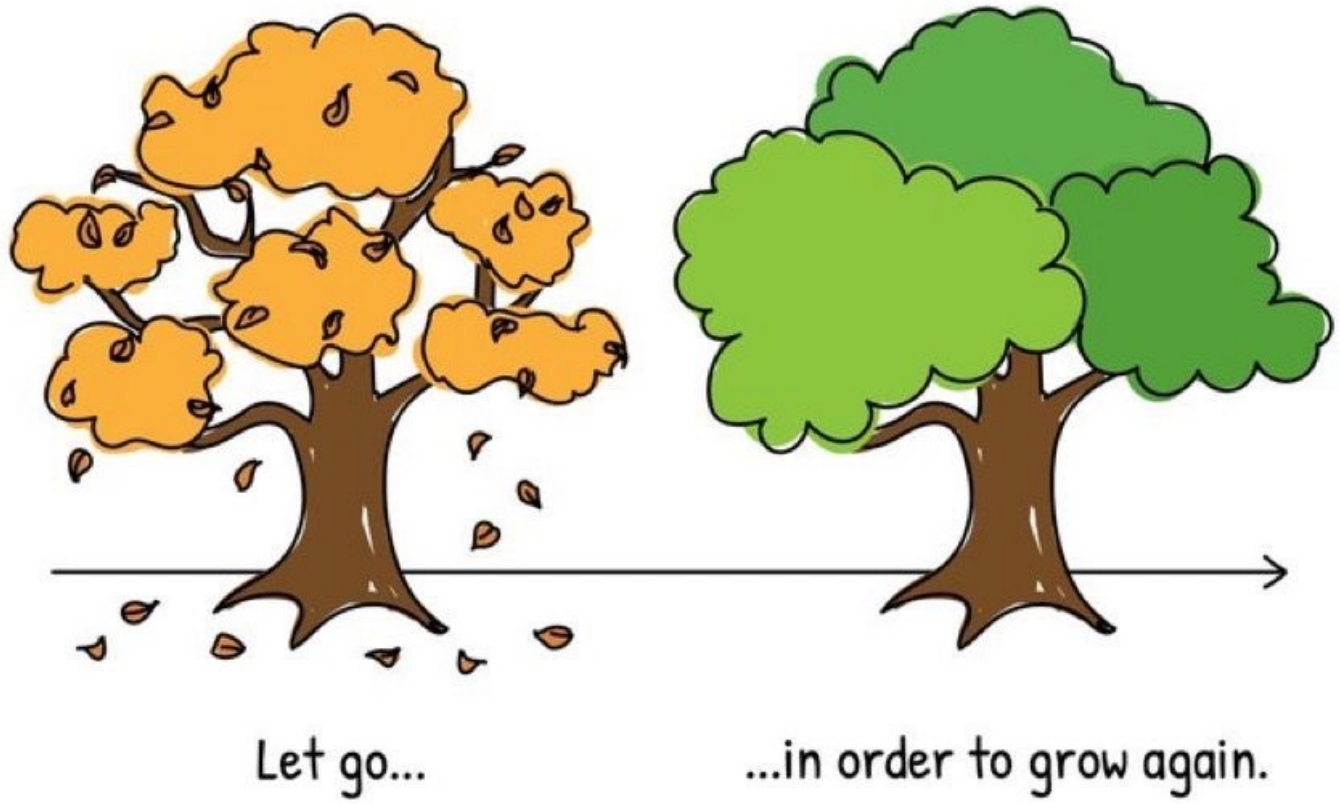
Numbers Systems (Decimal, Binary, Octal & Hexadecimal)

Ten’s Complement

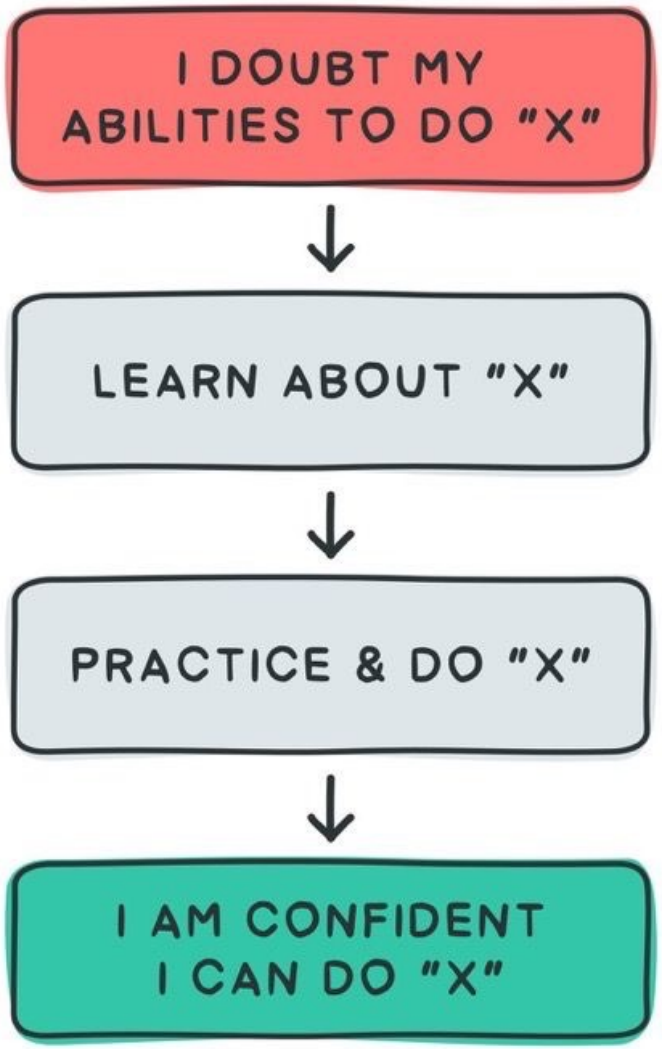
Twos Complement

main function

Stream in and strem out operators



INFORMATION TECHNOLOGY UNIVERSITY



Using The 2's Compliment Process

Use the 2's complement process to add together the following numbers.

POS		9		NEG		(-9)
+ POS	⇒	+ 5		+ POS	⇒	+ 5
POS		14		NEG		- 4

POS		9		NEG		(-9)
+ NEG	⇒	+ (-5)		+ NEG	⇒	+ (-5)
POS		4		NEG		- 14

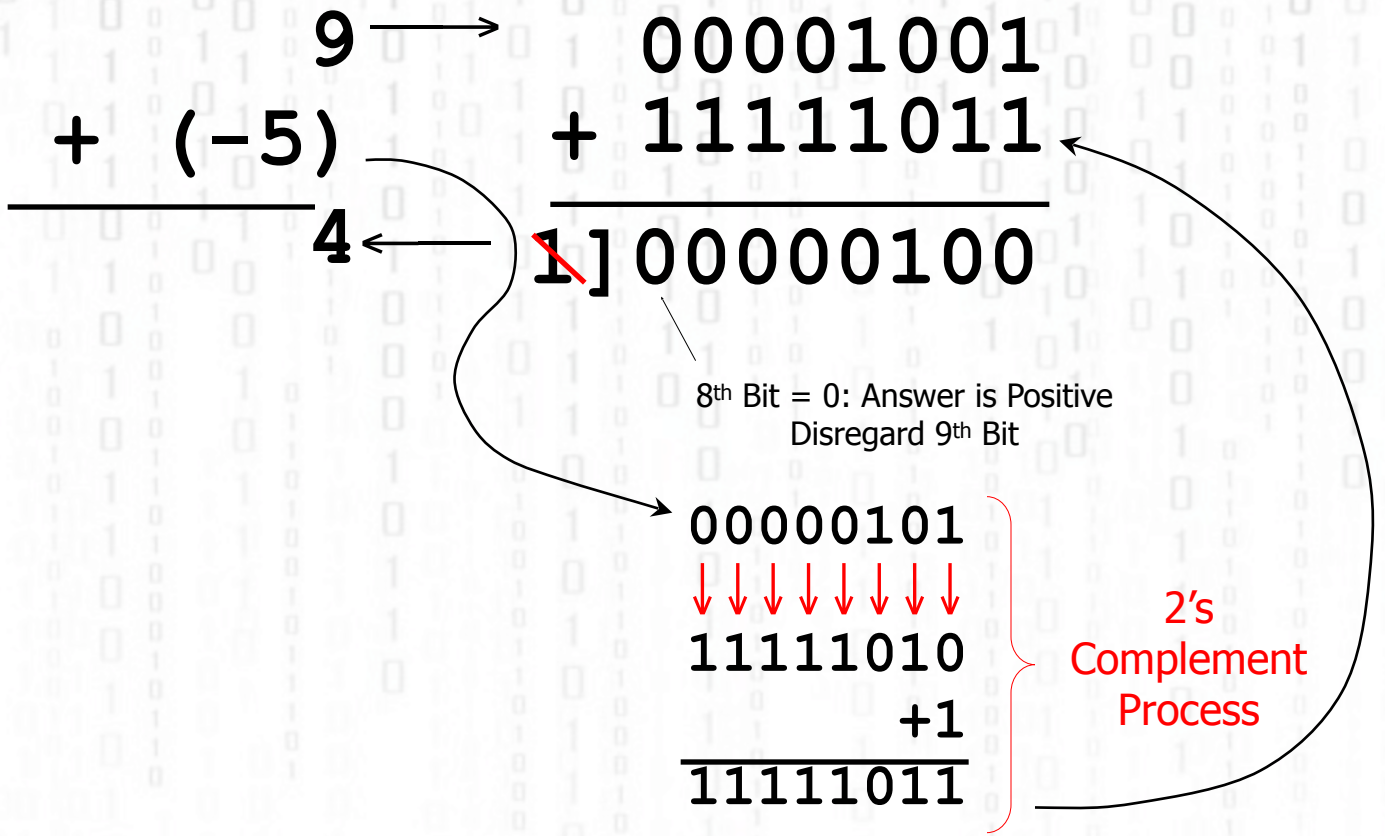
POS + POS → POS Answer

If no 2's complement is needed, use regular binary addition.

```
   9 ⟶    00001001
+  5 ⟶  + 00000101
------   ----------
  14 ⟵    00001110
```

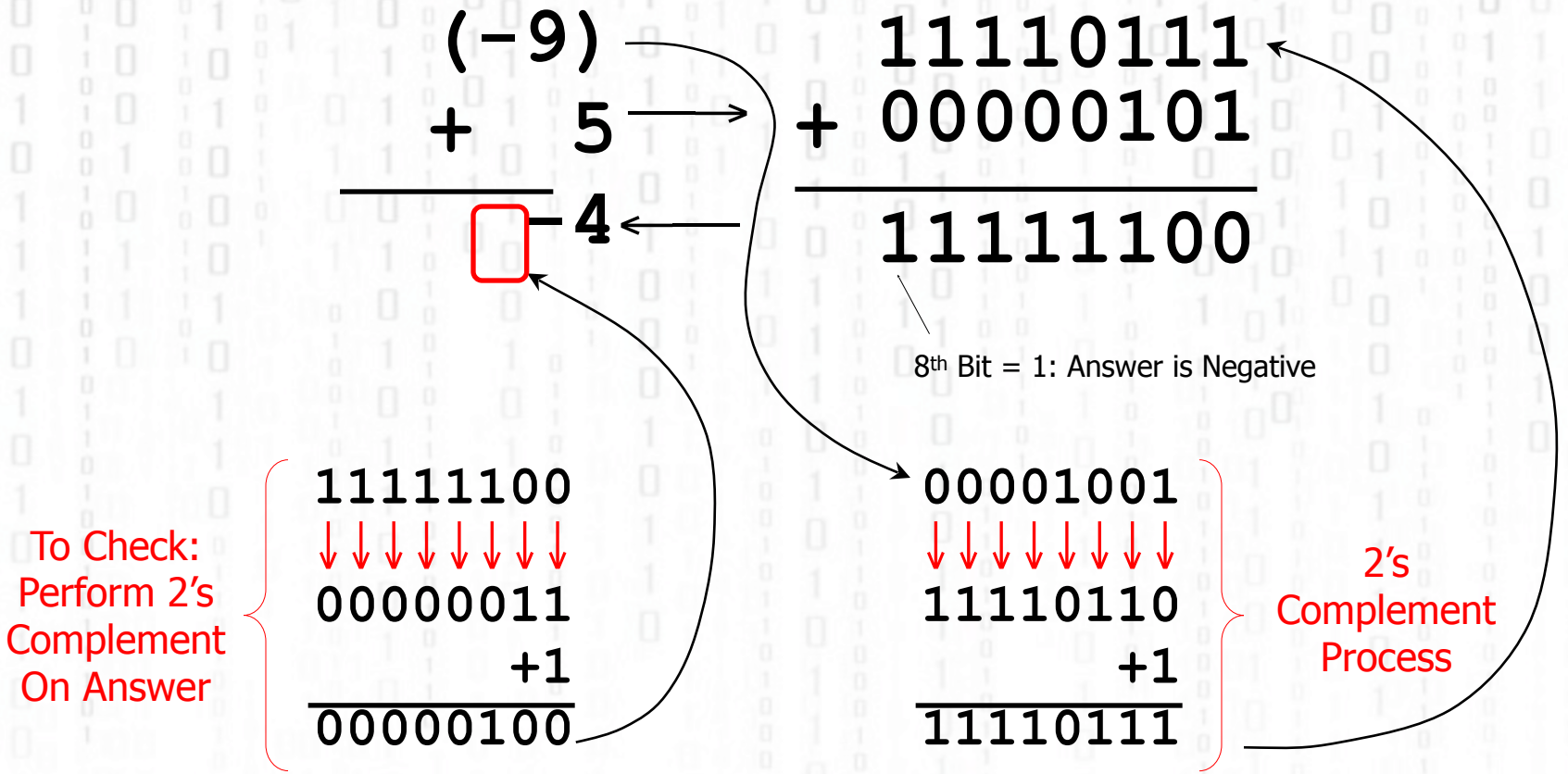
POS + NEG → POS Answer

Take the 2's complement of the negative number and use regular binary addition.



POS + NEG → NEG Answer

Take the 2's complement of the negative number and use regular binary addition.



NEG + NEG → NEG Answer

Take the 2's complement of both negative numbers and use regular binary addition.

```
   (-9) ⟶    11110111
 + (-5) ⟶  + 11111011
--------    --------
  -14  ⟵   1]11110010
```

2's Complement Numbers, See Conversion Process In Previous Slides

8th Bit = 1: Answer is Negative
Disregard 9th Bit

To Check: Perform 2's Complement On Answer

```
11110010
↓↓↓↓↓↓↓↓
00001101
      +1
--------
00001110
```

HOW TO CODE?

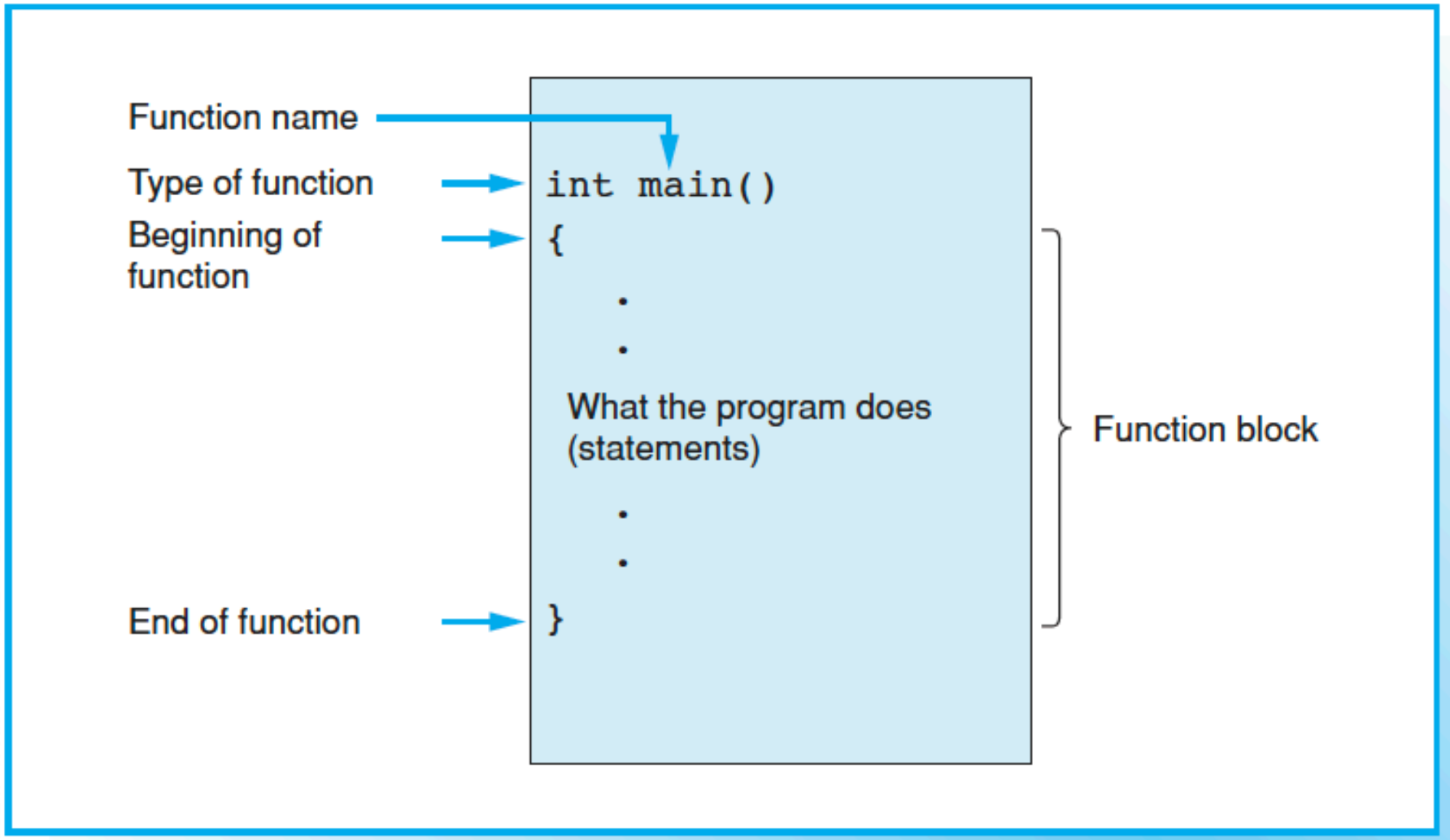
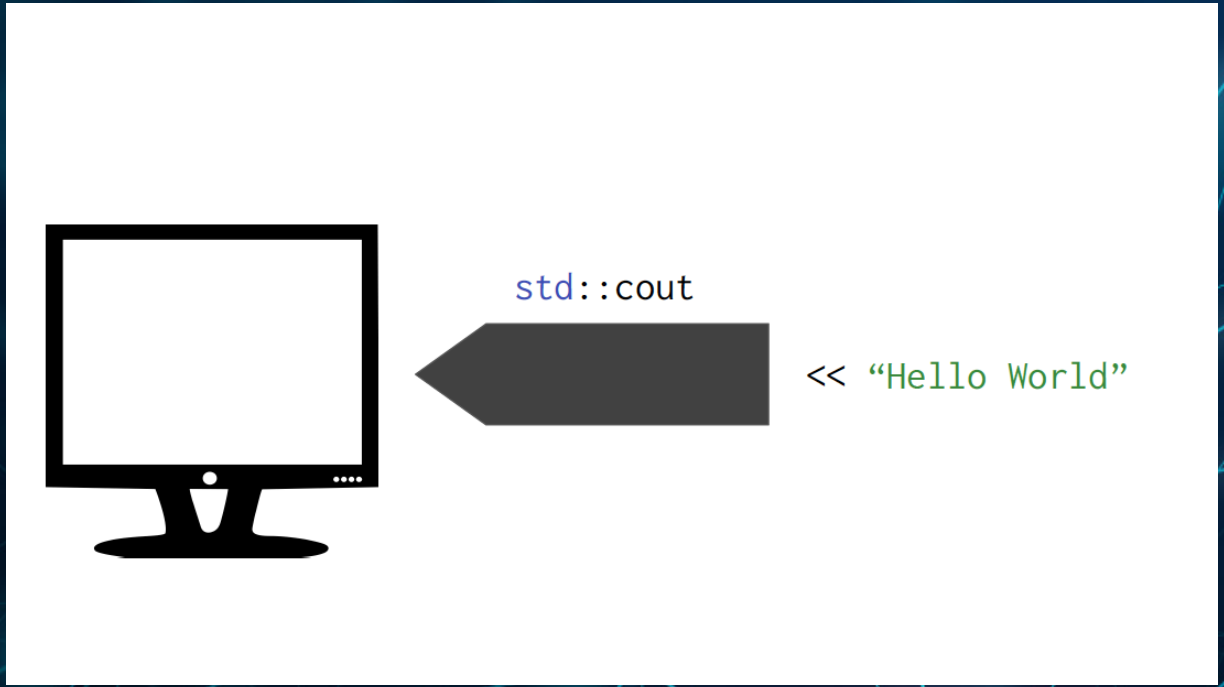


Image Reference : A Complete Guide to Programming in C++

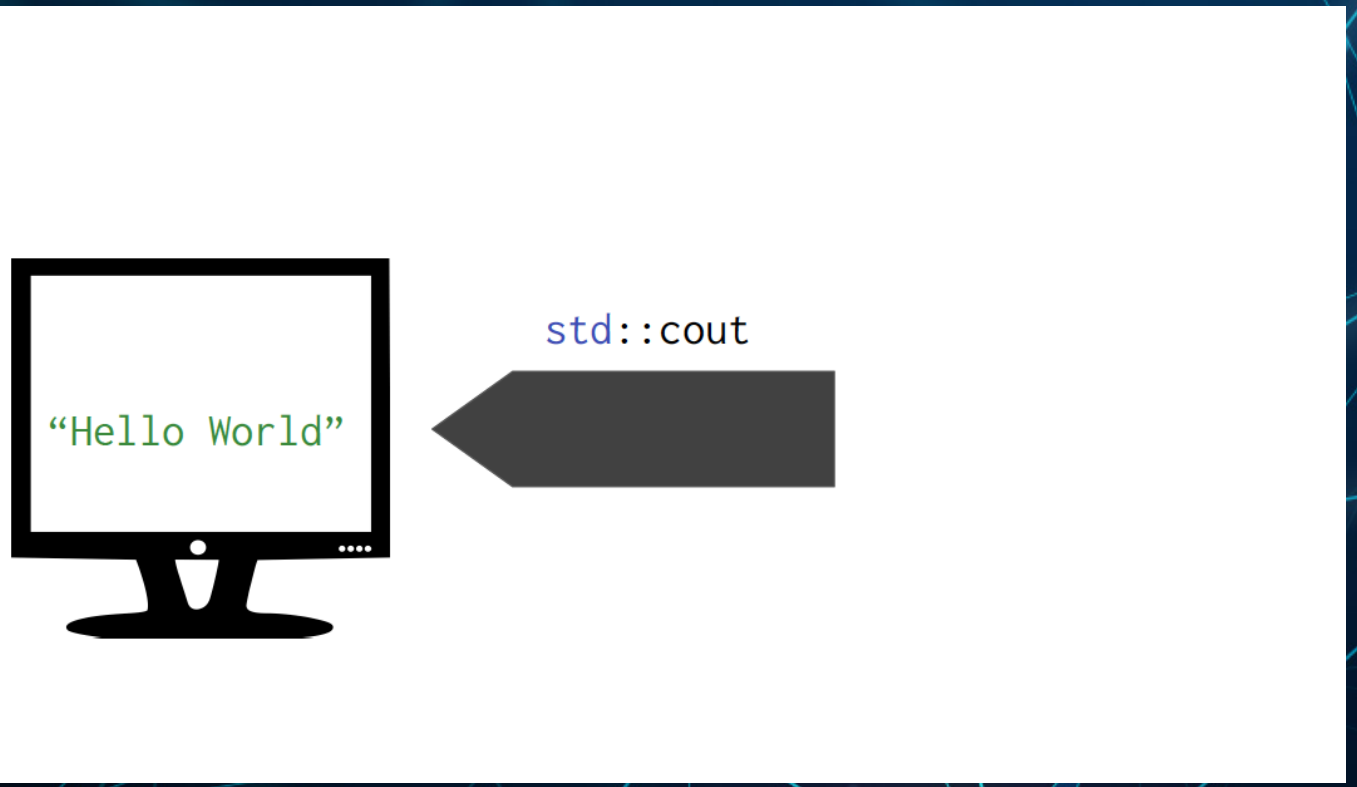
Streams

- A stream is an abstraction for input/output



Streams

- A stream is an abstraction for input/output



Stream out

```
cout << “Strings work!” << endl;
cout << 1729 << endl;
cout << 3.14 << endl;
cout << “Mixed types: ” << 1123 << endl;
```

Stream in

```
int x ;

cin >> x
```

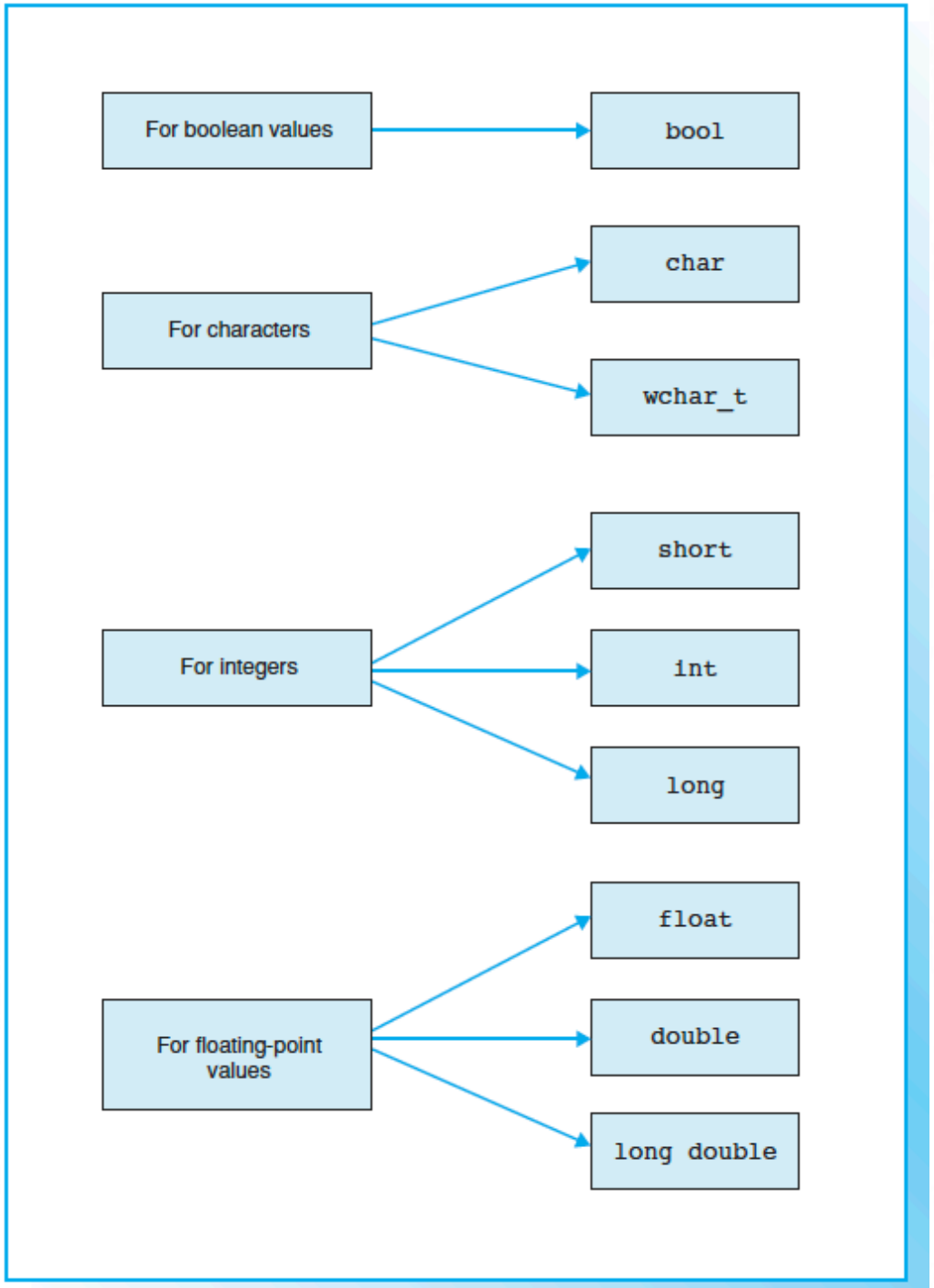


Image Reference : A Complete Guide to Programming in C++

Type	Size	Range of Values (decimal)
`char`	1 byte	-128 to +127 or 0 to 255
`unsigned char`	1 byte	0 to 255
`signed char`	1 byte	-128 to +127
`int`	2 byte resp. 4 byte	-32768 to +32767 resp. -2147483648 to +2147483647
`unsigned int`	2 byte resp. 4 byte	0 to 65535 resp. 0 to 4294967295
`short`	2 byte	-32768 to +32767
`unsigned short`	2 byte	0 to 65535
`long`	4 byte	-2147483648 to +2147483647
`unsigned long`	4 byte	0 to 4294967295

Type	Size	Range of Values	Lowest Positive Value	Accuracy (decimal)
`float`	4 bytes	–3.4E+38	1.2E—38	6 digits
`double`	8 bytes	–1.7E+308	2.3E—308	15 digits
`long double`	10 bytes	–1.1E+4932	3.4E—4932	19 digits

Image Reference : A Complete Guide to Programming in C++

ARITHMETIC OPERATORS

Operator	Significance
+	Addition
–	Subraction
*	Multiplication
/	Division
%	Remainder

Image Reference : A Complete Guide to Programming in C++

RELATIONAL OPERATORS

Operator	Significance
<	less than
<=	less than or equal to
>	greater than
>=	geater than or equal to
==	equal
!=	unequal

LOGICAL OPERATORS

A	!A
true	false
false	true

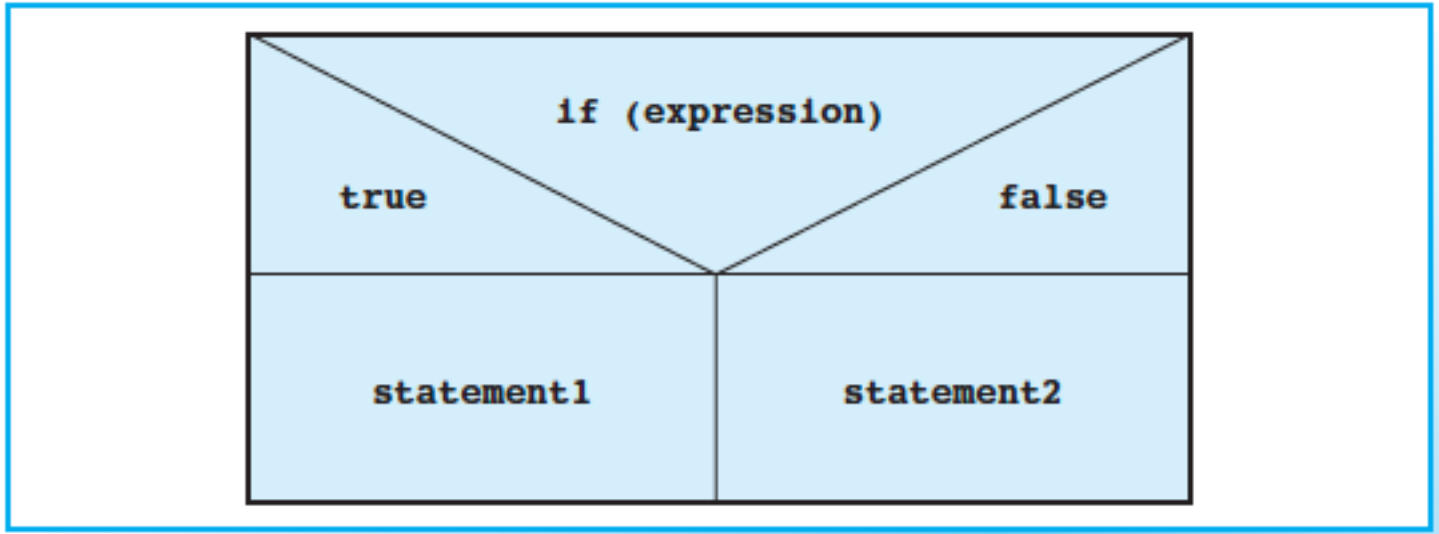
Image Reference : A Complete Guide to Programming in C++

LOGICAL OPERATORS

A	B	A && B	A || B
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Image Reference : A Complete Guide to Programming in C++

IF ELSE



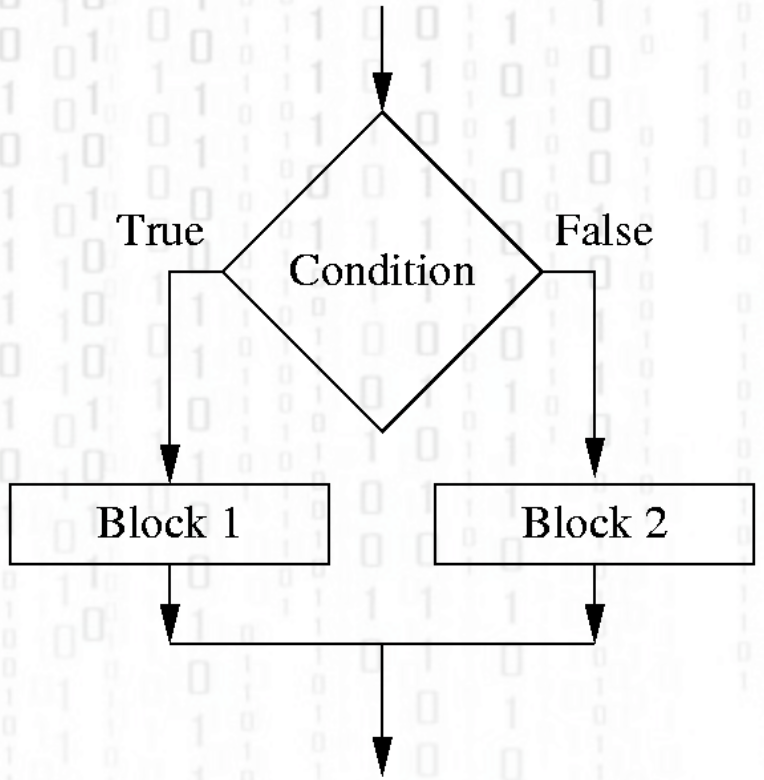


Image Reference : A Complete Guide to Programming in C++

IF ELSE

```
// if_else.cpp
// Demonstrates the use of if-else statements

#include <iostream>
using namespace std;
int main()
{
   float x, y, min;

   cout << "Enter two different numbers:\n";
   if( cin >> x && cin >> y)   // If both inputs are
   {                           // valid, compute
     if( x < y )               // the lesser.
        min = x;
     else
       min = y;
     cout << "\nThe smaller number is: " << min << endl;
   }
   else
     cout << "\nInvalid Input!" << endl;

   return 0;
}
```

ELSE IF CHAINS

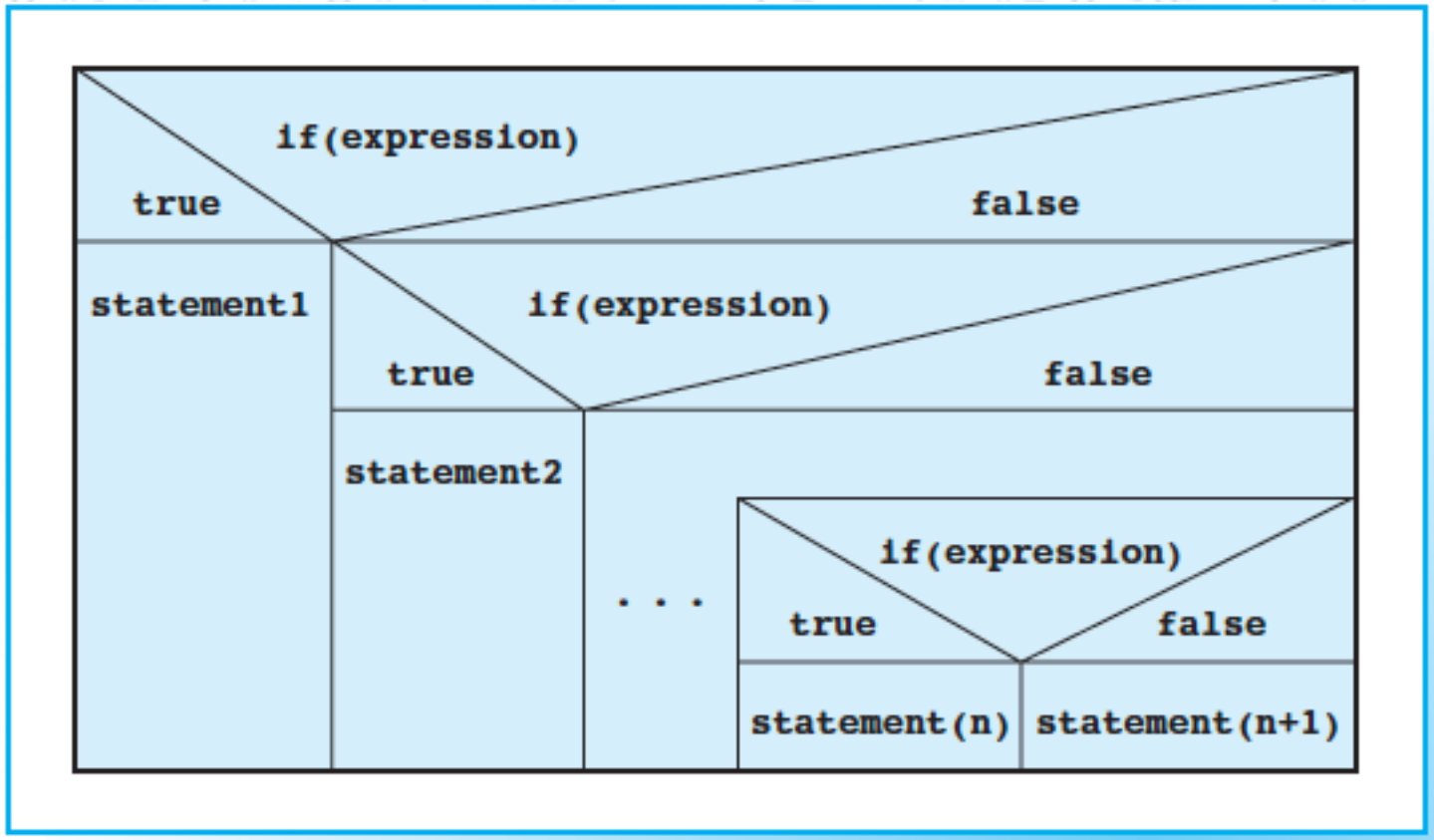
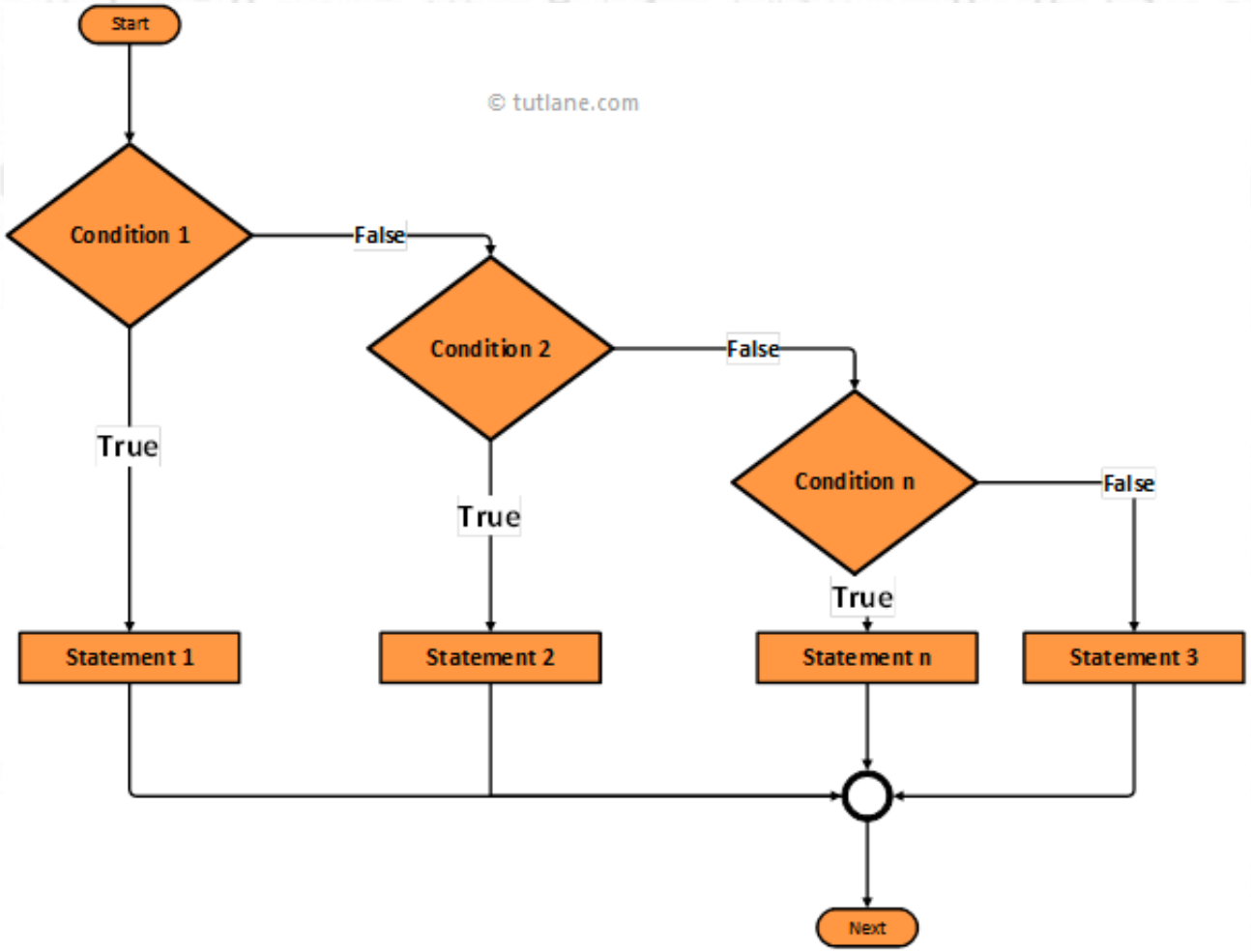


Image Reference : A Complete Guide to Programming in C++

tutlane.com

ELSE IF CHAINS



INFORMATION TECHNOLOGY UNIVERSITY

ELSE IF CHAINS

```
// speed.cpp
// Output the fine for driving too fast.

#include <iostream>
using namespace std;

int main()
{
   float limit, speed, toofast;
   cout << "\nSpeed limit: ";
   cin >> limit;
   cout << "\nSpeed: ";
   cin >> speed;

   if( (toofast = speed - limit ) < 10)
     cout << "You were lucky!" << endl;
   else if( toofast < 20)
     cout << "Fine payable: 40,-. Dollars" << endl;
   else if( toofast < 30)
     cout << "Fine payable: 80,-. Dollars" << endl;
   else
     cout << "Hand over your driver's license!" << endl;
   return 0;
}
```

Functions

```
[type] name([declaration_list]) // Function header
{                               // Beginning
    .
    .
   What will be done            // Function block
    .
    .
}                               // End
```

Image Reference : A Complete Guide to Programming in C++

Functions

```
// func1.cpp
#include <iostream>
using namespace std;

void test( int, double );                    // Prototype

int main()
{
  cout << "\nNow function test() will be called.\n";
  test( 10, -7.5);                           // Call
  cout << "\nAnd back again in main()." << endl;

  return 0;
}

void test(int arg1, double arg2 )            // Definition
{
   cout << "\nIn function test()."
        << "\n  1. argument: " << arg1
        << "\n  2. argument: " << arg2 << endl;
}
```

Functions

```
// area.cpp
// Example for a simple function returning a value.
//-------------------------------------------------
#include <iostream>
#include <iomanip>
using namespace std;

double area(double, double);          // Prototype

int main()
{
    double  x = 3.5, y = 7.2,  res;

    res = area( x, y+1);              // Call

   // To output to two decimal places:
    cout << fixed << setprecision(2);
    cout << "\n The area of a rectangle "
         << "\n with width  " << setw(5)  << x
         << "\n and length  " << setw(5) << y+1
         << "\n is          " << setw(5) << res
         << endl;
    return 0;
}

// Defining the function area():
// Computes the area of a rectangle.
double area( double width, double len)
{
    return (width * len);    // Returns the result.
}
```

Functions

```
long func2(int, double);          // Prototype
//  . . .
void func1()
{
   int x = 1.1;
   double y;
   . . .
   long a = func2(x,y);           // Call of func2().
   . . .
}                                 // Pass by value

long func2(int a, double b) // Definition
{
   double x = 2.2;
   long result;
      .          // Here the result
      .          // is computed.
      .
   return result;
}
```

Functions

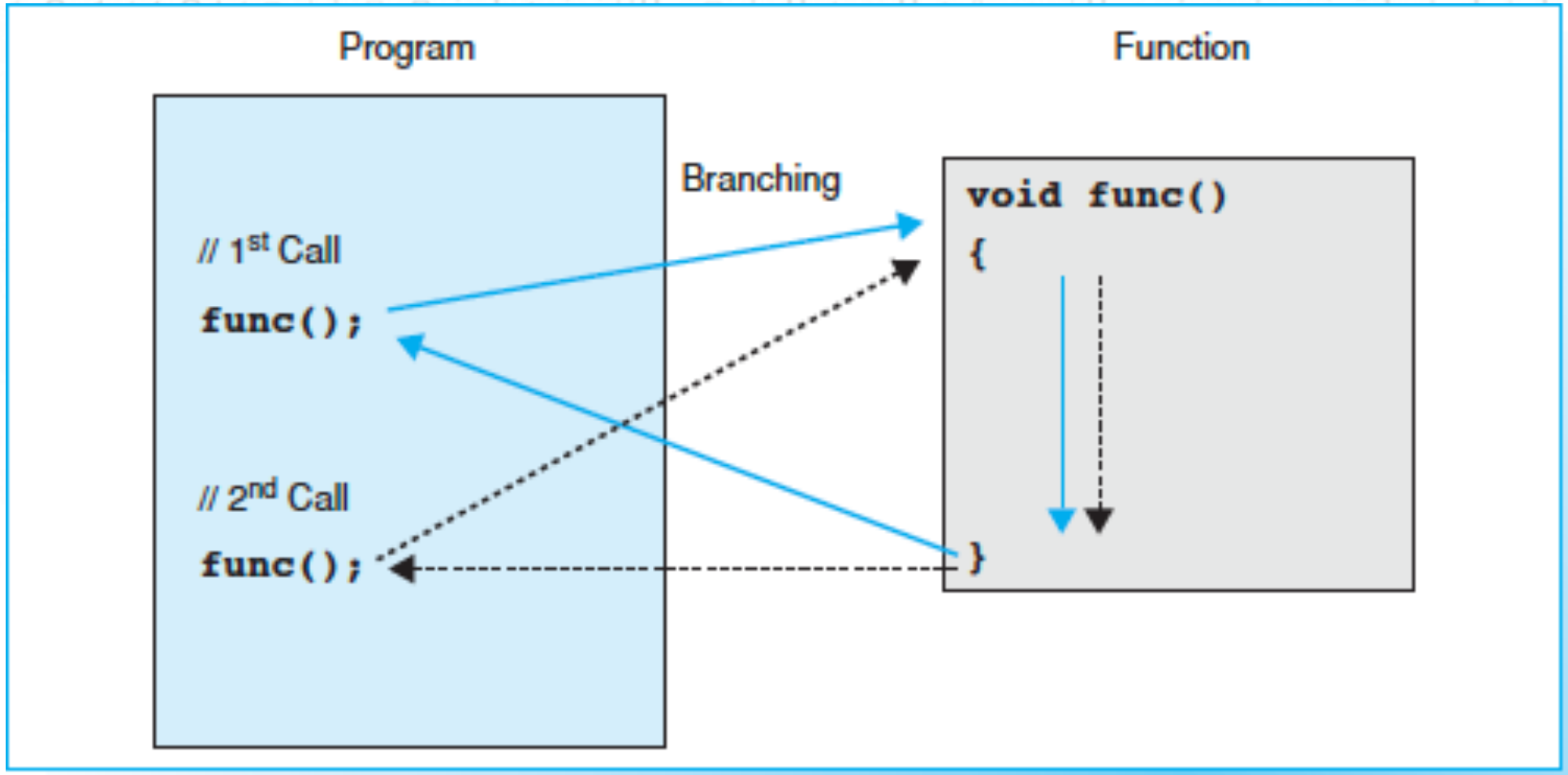


Image Reference : A Complete Guide to Programming in C++

Functions

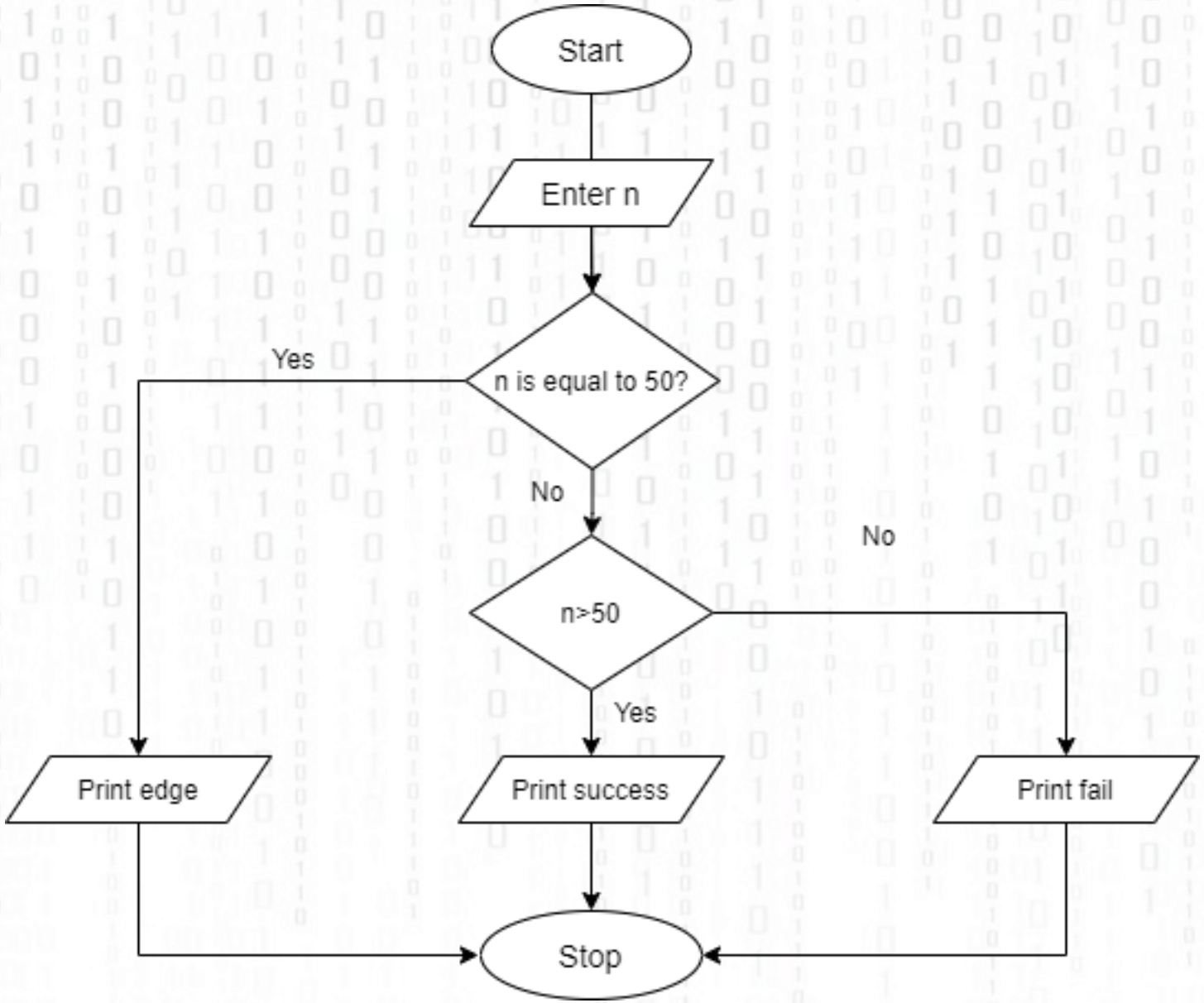
```
// Function capital() with two default arguments
// Prototype:
double capital( double k0, double p=3.5, double n=1.0);

double endcap;

endcap = capital( 100.0, 3.5, 2.5);
endcap = capital( 2222.20, 4.8);
endcap = capital( 3030.00);

endcap = capital( );


endcap = capital( 100.0, , 3.0);


endcap = capital( , 5.0);
```

Image Reference : A Complete Guide to Programming in C++

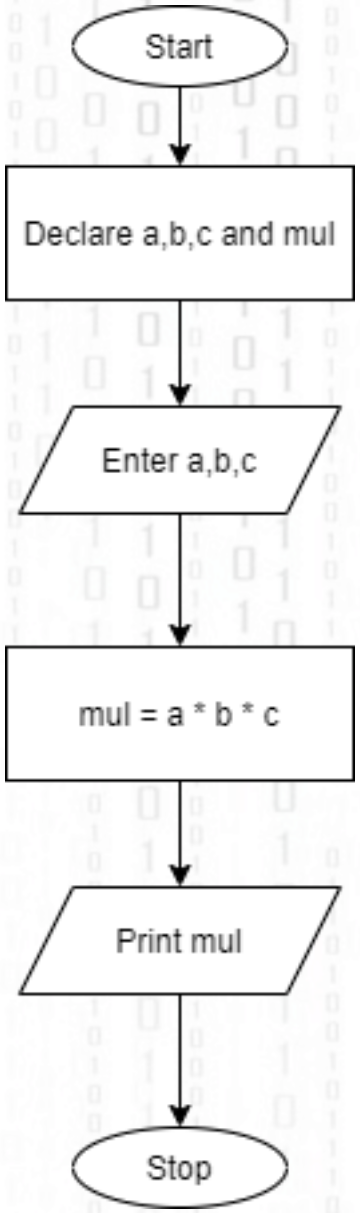
Functions

```
// Function capital() with two default arguments
// Prototype:
double capital( double k0, double p=3.5, double n=1.0);

double endcap;

endcap = capital( 100.0, 3.5, 2.5);   // ok
endcap = capital( 2222.20, 4.8);      // ok
endcap = capital( 3030.00);           // ok

endcap = capital( );                  // not ok
// The first argument has no default value.

endcap = capital( 100.0, , 3.0);      // not ok
// No gap!

endcap = capital( , 5.0);             // not ok
// No gap either.
```

A number is given as input. If the number is greater than 50, print ‘success’. If the number is less than 50, print ‘fail’. If the number is equal to 50, print ‘edge’



Multiplication of three numbers.



Check if the given number is even or odd

Your program should take marks obtained by the student and the total marks of the student from the user. Then based on following percentage ranges grade them letter grade.

0% -49.99% - F

50% -59.99% - E

60% -69.99% - D

70% -79.99% - C

80% -89.99% - B

90% -100.00% - A

Output could be

Passed with Grade A
Passed with Grade B
Passed with Grade C
Passed with Grade D
Passed with Grade E
Failed with Grade F