Lecture 5

Number System & Coding

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

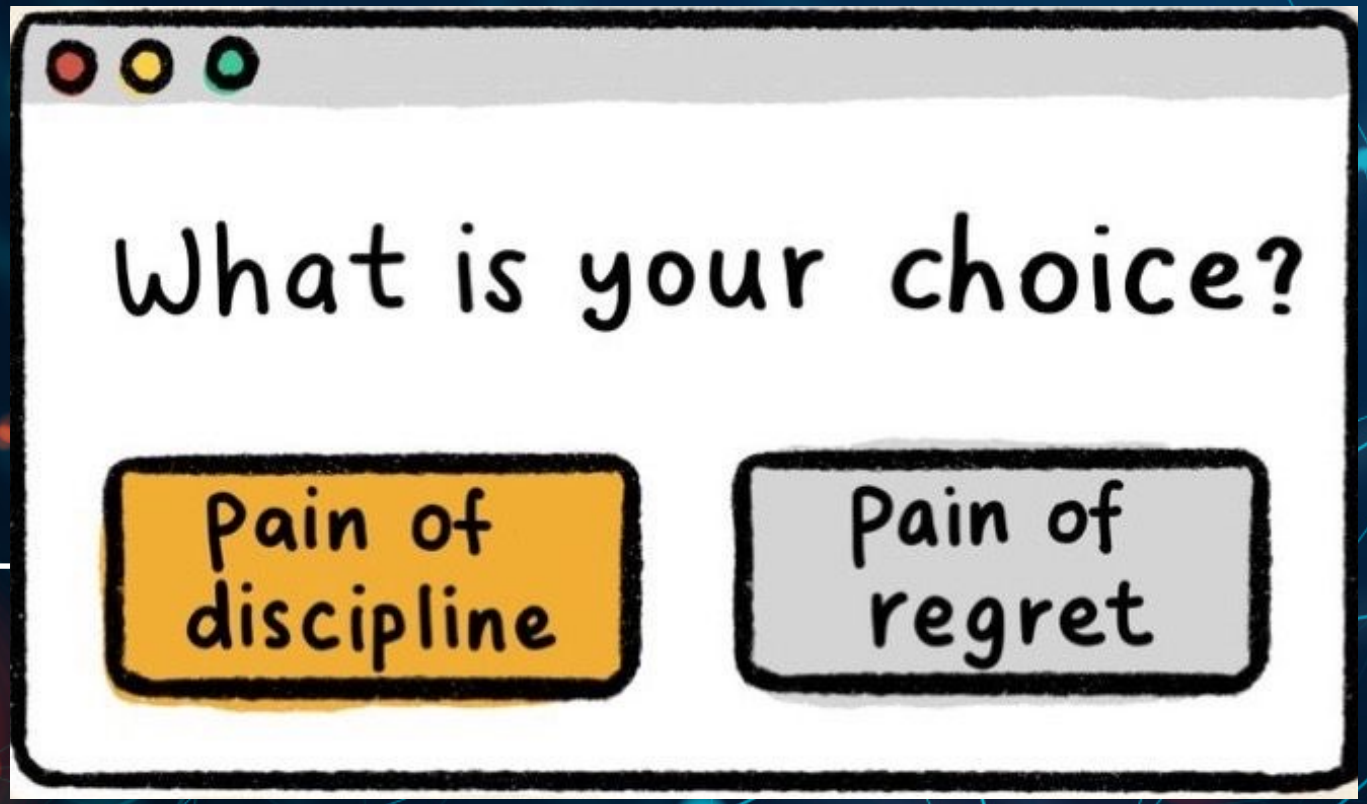
YOUR NAME
WEEK-13

YOUR NAME
WEEK-14

YOUR NAME
WEEK-15

UBS

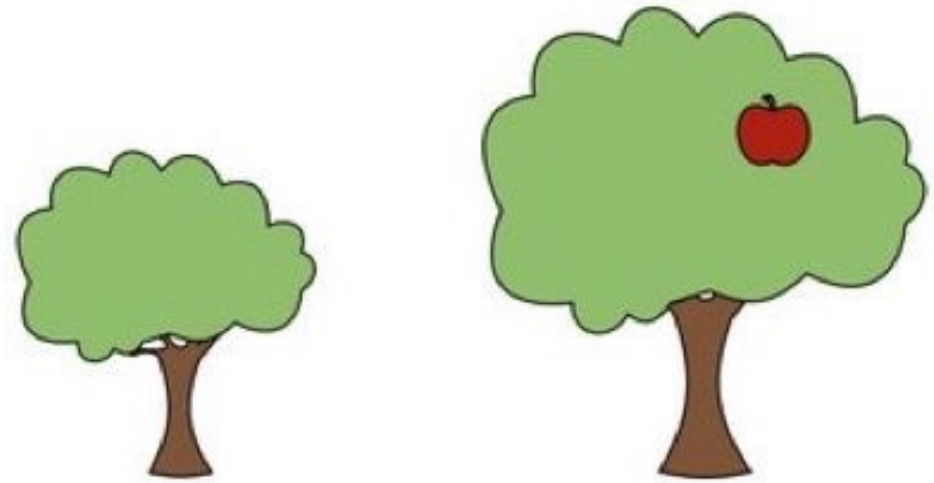
INFORMATION TECHNOLOGY UNIVERSITY



INFORMATION TECHNOLOGY UNIVERSITY

EVALUATING YOUR GROWTH

YOURSELF VS OTHERS



YOURSELF 2021

YOURSELF 2022

RECAP

GitHub

Tools (Cygwin, IDE, GitHub)

Flowcharts

Algorithms

Approach towards a word problem

Pseudocode

Flowcharts Advantages & Disadvantages

Numbers Systems (Decimal, Binary, Octal & Hexadecimal)

110

NUMBER SYSTEMS

110_{10}

Decimal

INFORMATION TECHNOLOGY UNIVERSITY

INFORMATION TECHNOLOGY UNIVERSITY

2347

- As you saw, there are 2 groups of a thousand. Not coincidentally, 1000 = 10*10*10, which can also be written as 10^3.
- There are 3 groups of a hundred. Again, not coincidentally, 100 = 10*10 or 10^2.
- There are 4 groups of ten, and, 10 = 10^1.
- Finally, there are 7 groups of one, and 1 = 10^0. (That may seem strange, but any number to the power of 0 equals 1, by definition.)

https://code.tutsplus.com/articles/number-systems-an-introduction-to-binary-hexadecimal-and-more--active-10848

$$892 = 8*10^2+9*10^1+2*10^0$$

$$1147 = 1*10^3+1*10^2+4*10^1+7*10^0$$

$$53 = 5*10^1+3*10^0$$

https://code.tutsplus.com/articles/number-systems-an-introduction-to-binary-hexadecimal-and-more--active-10848

Numbers System	Base
Binary	2
Octal	8
Decimal	10
Hexadecimal	16

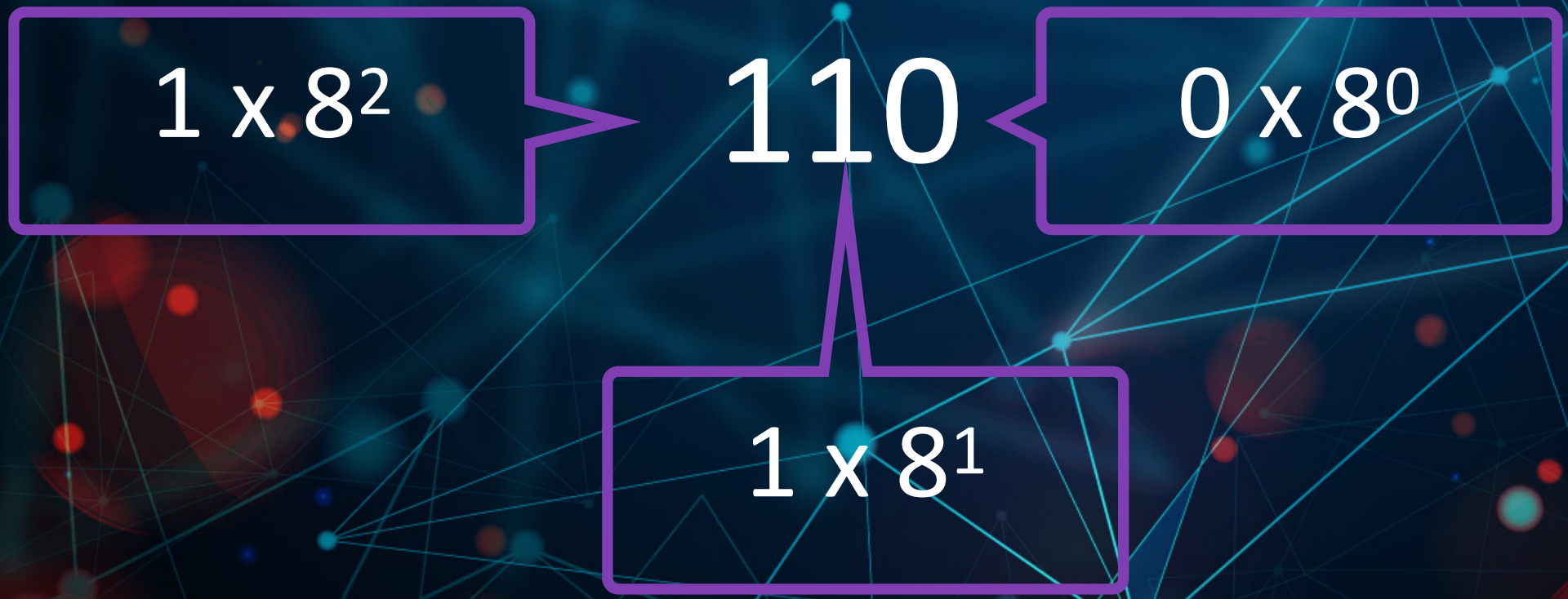
Octal

Octal

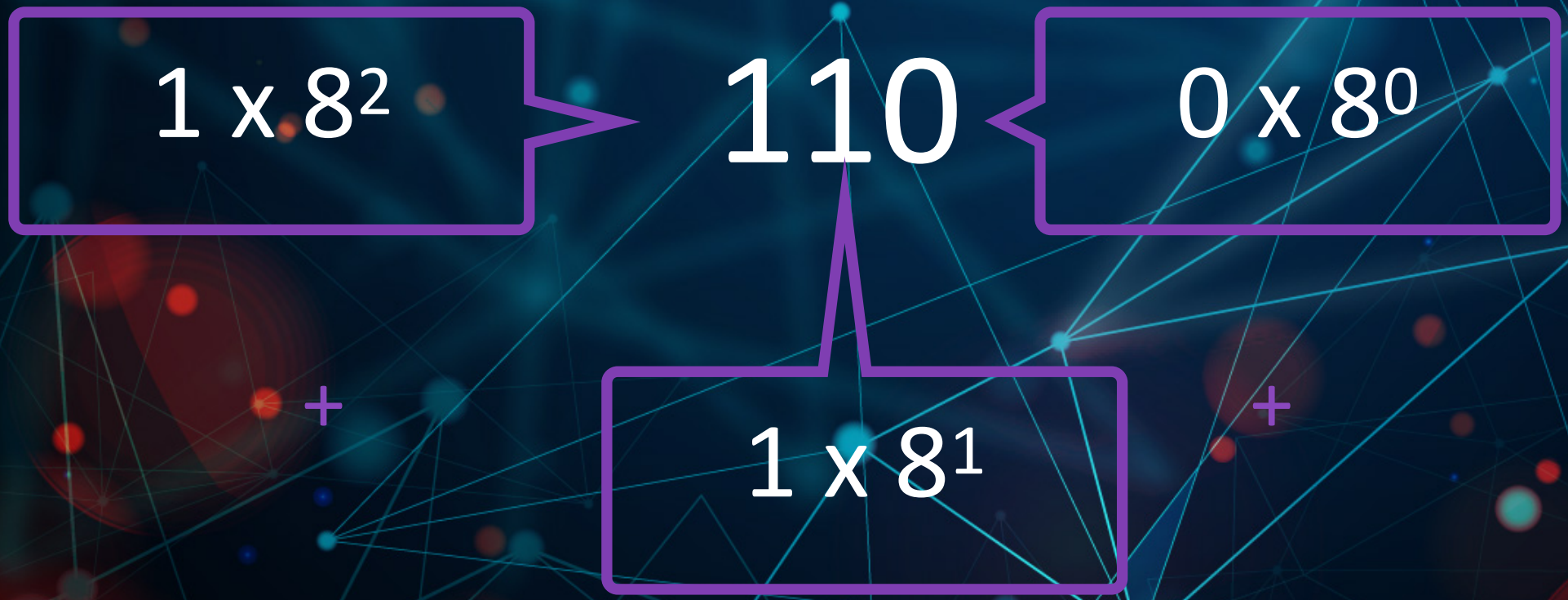
only eight digits: 0, 1, 2, 3, 4, 5, 6, and 7

110_8

INFORMATION TECHNOLOGY UNIVERSITY



INFORMATION TECHNOLOGY UNIVERSITY



INFORMATION TECHNOLOGY UNIVERSITY

$$110_8 = 72_{10}$$

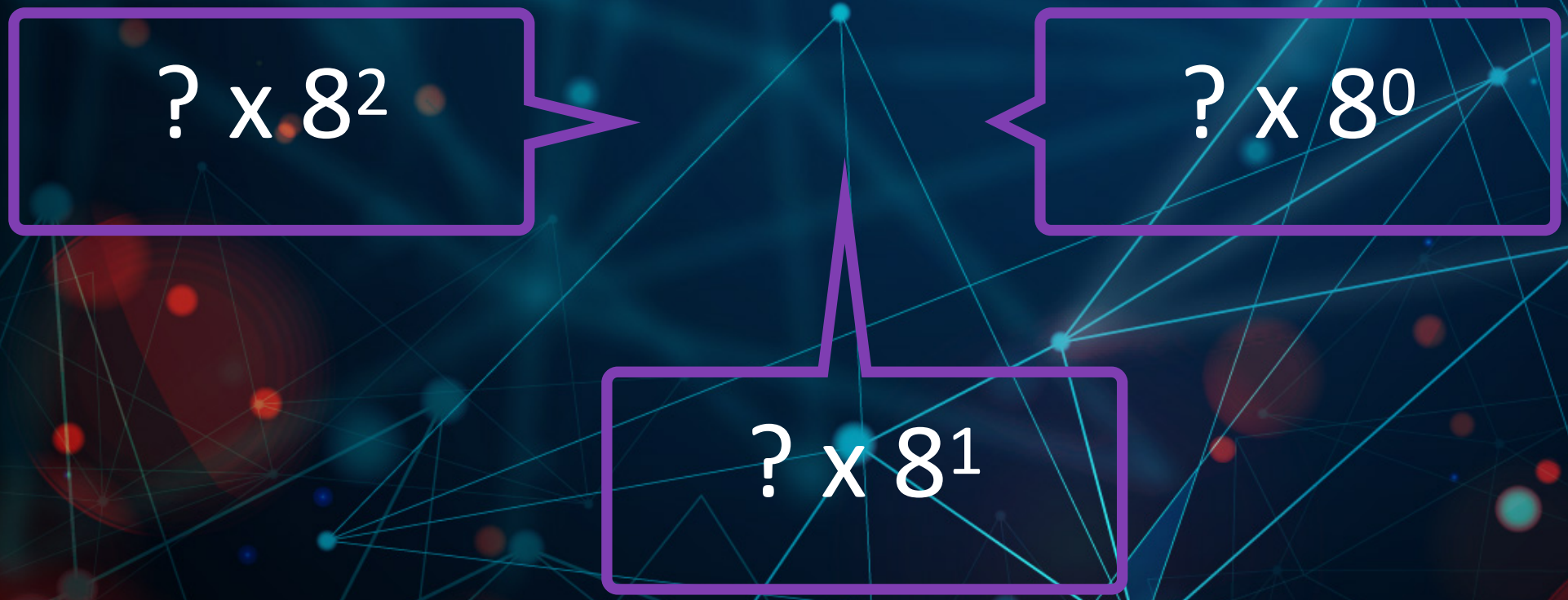
77_8

John offers to give you 47_8 cookies, and Jane offers to give you 43_{10} cookies. Whose offer do you take?

https://code.tutsplus.com/articles/number-systems-an-introduction-to-binary-hexadecimal-and-more--active-10848

77_{10}

77_{10}



$(77\text{-}64)_{10}$ 13_{10}

1×8^2

$? \times 8^0$

$? \times 8^1$

$(77\text{-}64\text{-}8)_{10}$ 5_{10}

1×8^2

$? \times 8^0$

1×8^1

$(77\text{-}64\text{-}8\text{-}5)_{10}$ 0_{10}

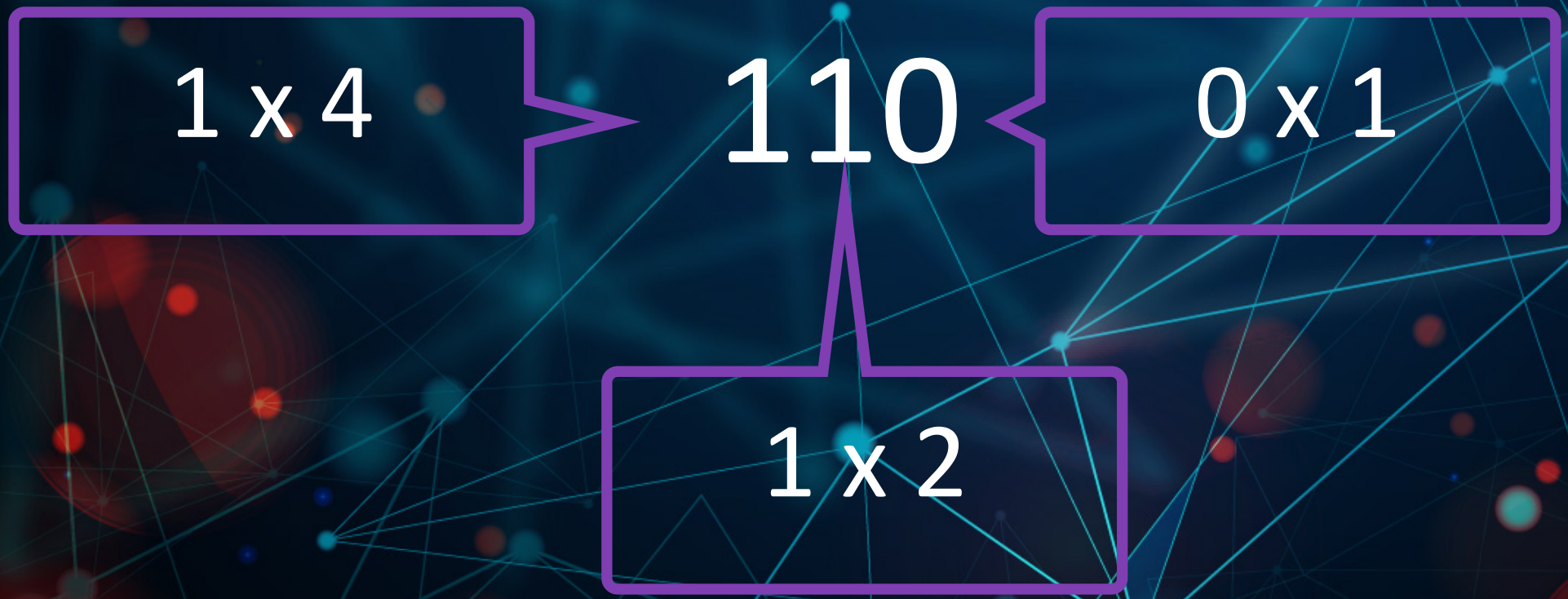
INFORMATION TECHNOLOGY UNIVERSITY

Binary

Binary

only two digits: 0, and 1

110_2



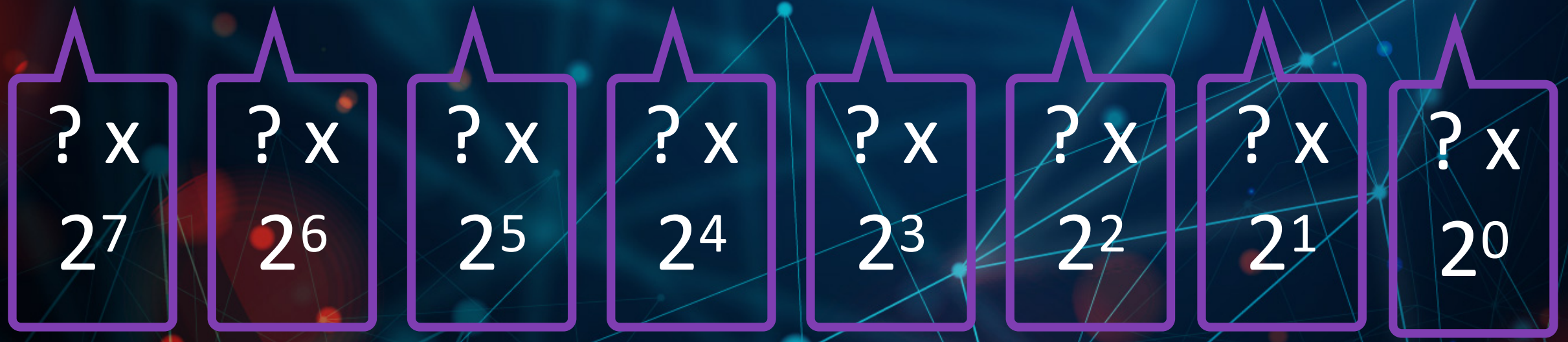
INFORMATION TECHNOLOGY UNIVERSITY

$$110_2 = 6_{10}$$

1111_2

77_{10}

77_{10}



$(77-64)_{10}$ 13_{10}

0 x	1 x	? x	? x	? x	? x	? x	? x
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$(77\text{-}64\text{-}8)_{10}$ 5_{10}

0 x 2^7	1 x 2^6	0 x 2^5	0 x 2^4	1 x 2^3	? x 2^2	? x 2^1	? x 2^0

$(77\text{-}64\text{-}8\text{-}4)_{10}$ 1_{10}

0 x 2^7	1 x 2^6	0 x 2^5	0 x 2^4	1 x 2^3	1 x 2^2	? x 2^1	? x 2^0

$(77\text{-}64\text{-}8\text{-}4\text{-}1)_{10}$ 0_{10}

0×2^7	1×2^6	0×2^5	0×2^4	1×2^3	1×2^2	0×2^1	1×2^0

$$01001101_2 = 77_{10}$$

Hexadecimal

Hexadecimal

only sixteen digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

110_{16}

$$110_{16} = 272_{10}$$

2's Complement Arithmetic

- That subtracting one number from another is the same as making one number negative and adding.
- How to create negative numbers in the binary number system.
- The 2's Complement Process.
- How the 2's complement process can be use to add (and subtract) binary numbers.

Negative Numbers?

- Digital electronics requires frequent addition and subtraction of numbers. You know how to design an adder, but what about a subtract-er?
- A subtract-er is not needed with the 2's complement process. The 2's complement process allows you to easily convert a positive number into its negative equivalent.
- Since subtracting one number from another is the same as making one number negative and adding, the need for a subtract-er circuit has been eliminated.

How To Create A Negative Number

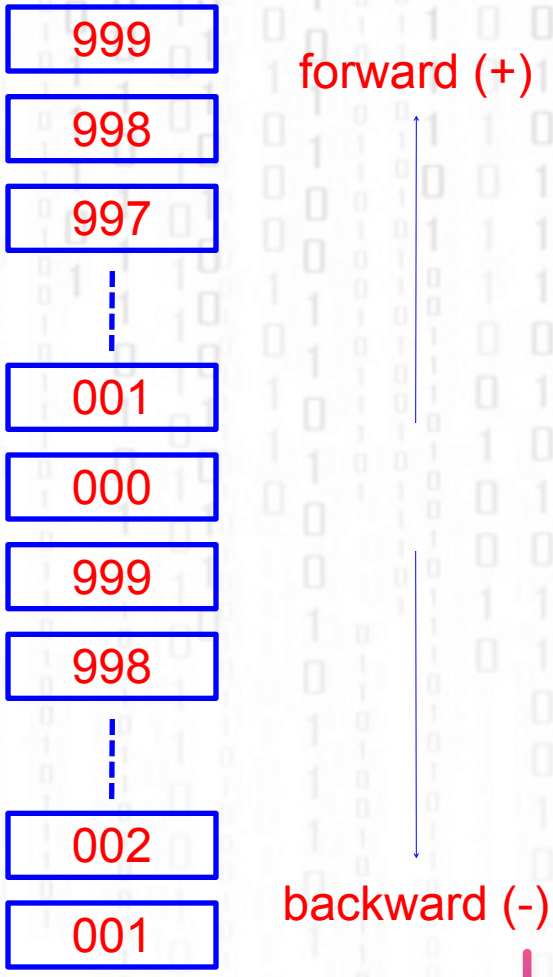
- In digital electronics you cannot simply put a minus sign in front of a number to make it negative.
- You must represent a negative number in a *fixed-length* binary number system. All signed arithmetic must be performed in a *fixed-length* number system.
- A physical *fixed-length* device (usually memory) contains a fixed number of bits (usually 4-bits, 8-bits, 16-bits) to hold the number.

3-Digit Decimal Number System

A bicycle odometer with only three digits is an example of a fixed-length decimal number system.

The problem is that without a negative sign, you cannot tell a +998 from a -2 (also a 998). Did you ride forward for 998 miles or backward for 2 miles?

Note: Car odometers do not work this way.



Negative Decimal

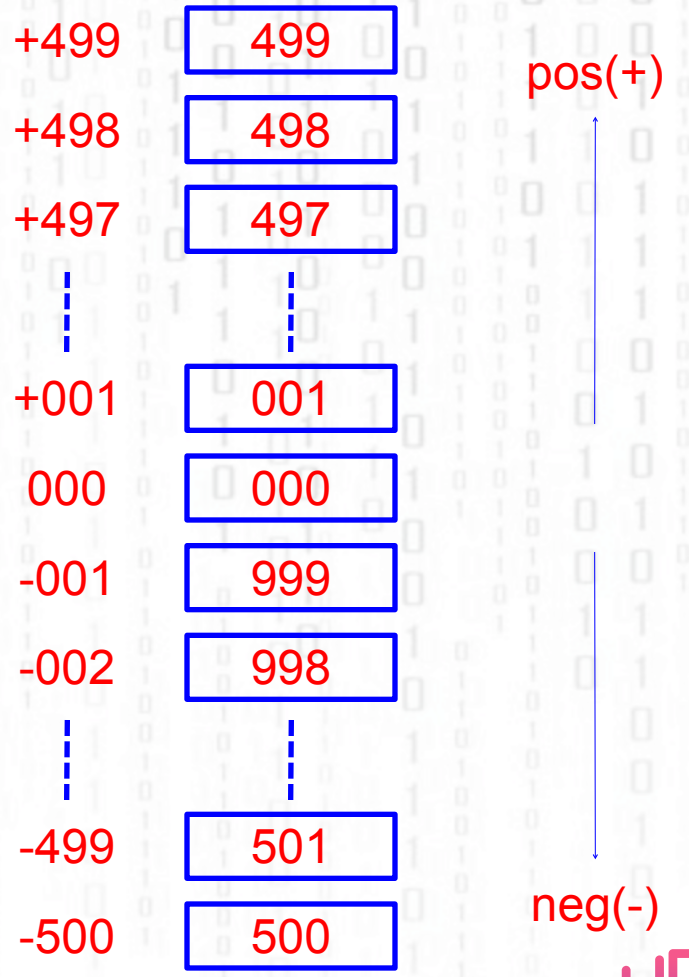
How do we represent negative numbers in this 3-digit decimal number system without using a sign?

→Cut the number system in half.

→Use 001 – 499 to indicate positive numbers.

→Use 500 – 999 to indicate negative numbers.

→Notice that 000 is not positive or negative.



“Odometer” Math Examples

$$\begin{array}{r} 3 \\ +\ 2 \\ \hline 5 \end{array} \qquad \begin{array}{r} 003 \\ +\ 002 \\ \hline 005 \end{array} \qquad \begin{array}{r} 6 \\ +\ (-3) \\ \hline 3 \end{array} \qquad \begin{array}{r} 006 \\ +\ 997 \\ \hline 1\,003 \end{array}$$

Disregard Overflow

$$\begin{array}{r} (-5) \\ +\ 2 \\ \hline (-3) \end{array} \qquad \begin{array}{r} 995 \\ +\ 002 \\ \hline 997 \end{array} \qquad \begin{array}{r} (-2) \\ +\ (-3) \\ \hline (-5) \end{array} \qquad \begin{array}{r} 998 \\ +\ 997 \\ \hline 1\,995 \end{array}$$

Disregard Overflow

It Works!

Complex Problems

- The previous examples demonstrate that this process works, but how do we *easily* convert a number into its negative equivalent?
- In the examples, converting the negative numbers into the 3-digit decimal number system was fairly easy. To convert the (-3), you simply counted backward from 1000 (i.e., 999, 998, 997).
- This process is not as easy for large numbers (e.g., -214 is 786). How did we determine this?
- To convert a large negative number, you can use the 10's Complement Process.

10's Complement Process

The **10's Complement** process uses base-10 (decimal) numbers. Later, when we're working with base-2 (binary) numbers, you will see that the **2's Complement** process works in the same way.

First, complement all of the digits in a number.

- A digit's complement is the number you add to the digit to make it equal to the largest digit in the base (i.e., 9 for decimal). The complement of 0 is 9, 1 is 8, 2 is 7, etc.

Second, add 1.

- Without this step, our number system would have two zeroes (+0 & -0), which no number system has.

10's Complement Examples

Example #1

```
-003
 ↓↓↓
 996
  +1
 ---
 997
```

Complement Digits

Add 1

Example #2

```
-214
 ↓↓↓
 785
  +1
 ---
 786
```

Complement Digits

Add 1

8-Bit Binary Number System

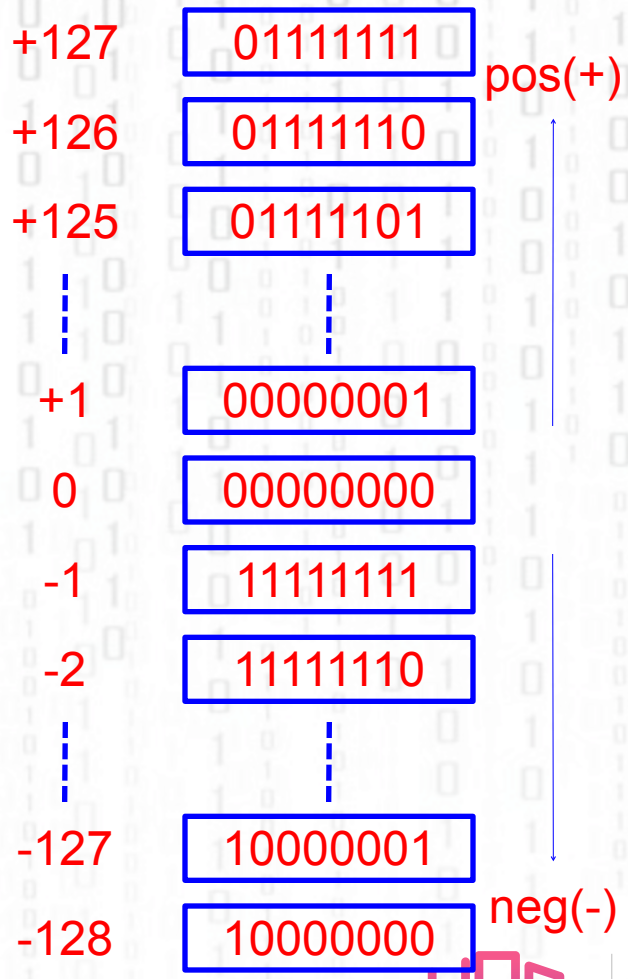
Apply what you have learned to the binary number systems. How do you represent negative numbers in this 8-bit binary system?

→Cut the number system in half.

→Use 00000001 – 01111111 to indicate positive numbers.

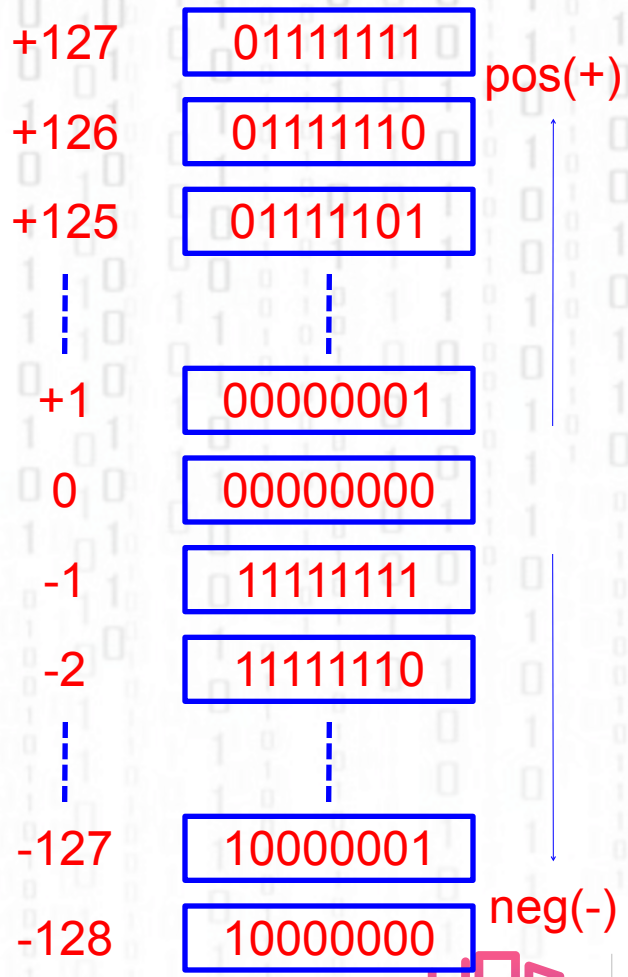
→Use 10000000 – 11111111 to indicate negative numbers.

→Notice that 00000000 is not positive or negative.



Sign Bit

- What did do you notice about the most significant bit of the binary numbers?
- The MSB is (0) for all positive numbers.
- The MSB is (1) for all negative numbers.
- The MSB is called the sign bit.
- In a signed number system, this allows you to instantly determine whether a number is positive or negative.



2’S Complement Process

The steps in the **2’s Complement** process are similar to the 10’s Complement process. However, you will now use the base two.

First, complement all of the digits in a number.

- A digit’s complement is the number you add to the digit to make it equal to the largest digit in the base (i.e., 1 for binary). In binary language, the complement of 0 is 1, and the complement of 1 is 0.

Second, add 1.

- Without this step, our number system would have two zeroes (+0 & -0), which no number system has.

2's Complement Examples

Example #1

```
 5 = 00000101
     ↓↓↓↓↓↓↓↓
     11111010
           +1
     --------
-5 = 11111011
```

Complement Digits

Add 1

Example #2

```
-13 = 11110011
      ↓↓↓↓↓↓↓↓
      00001100
            +1
      --------
 13 = 00001101
```

Complement Digits

Add 1

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

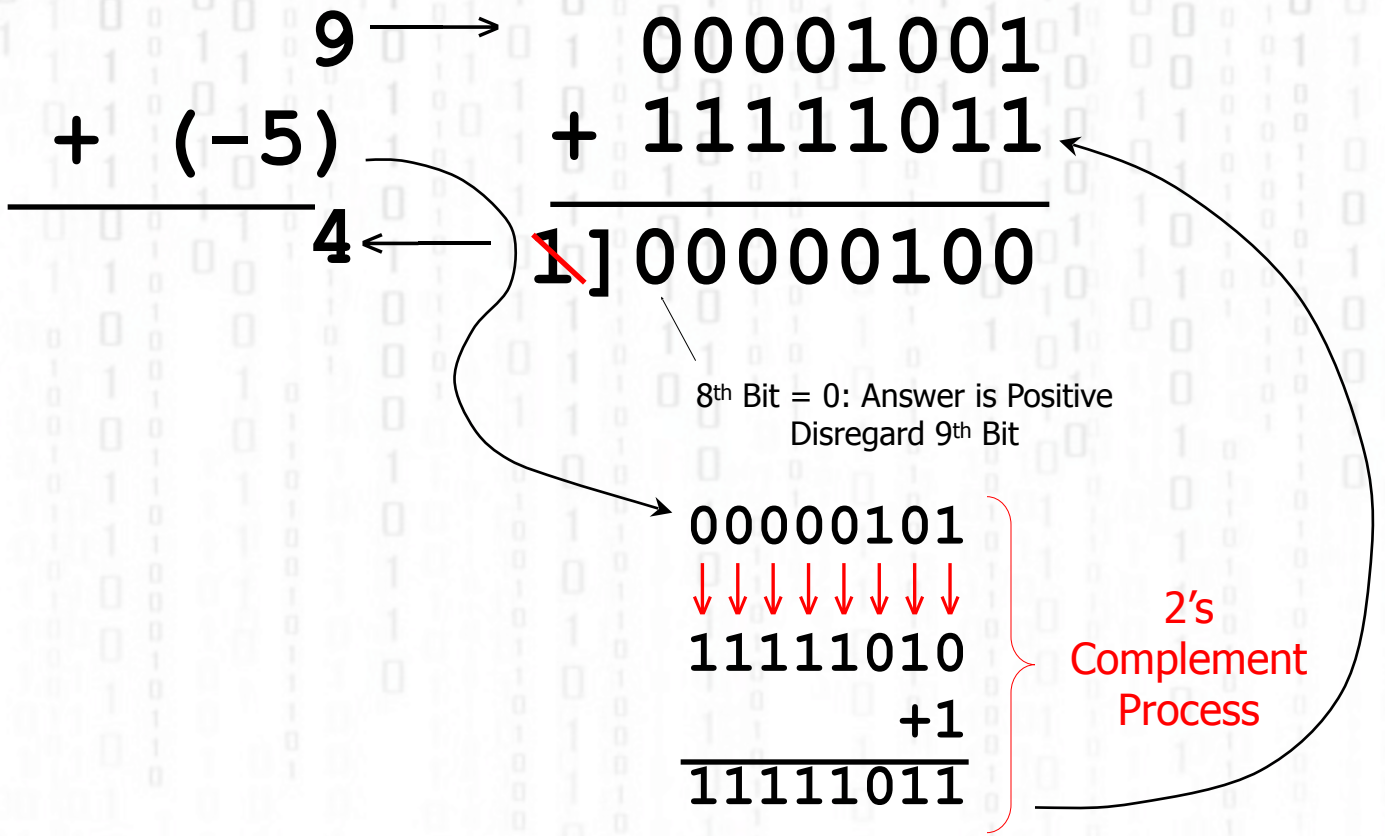
POS + POS → POS Answer

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

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

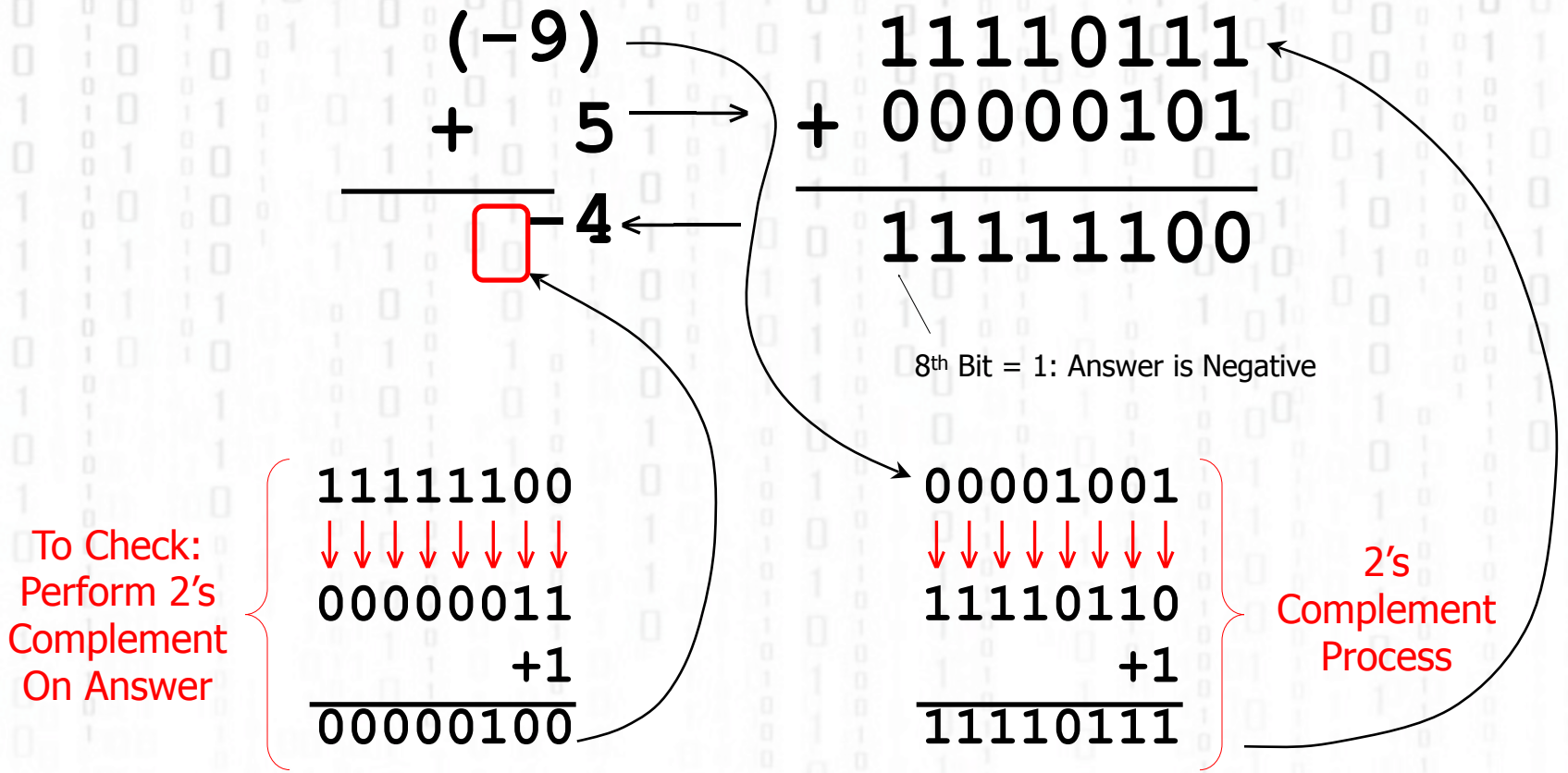
POS + NEG → POS Answer

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?

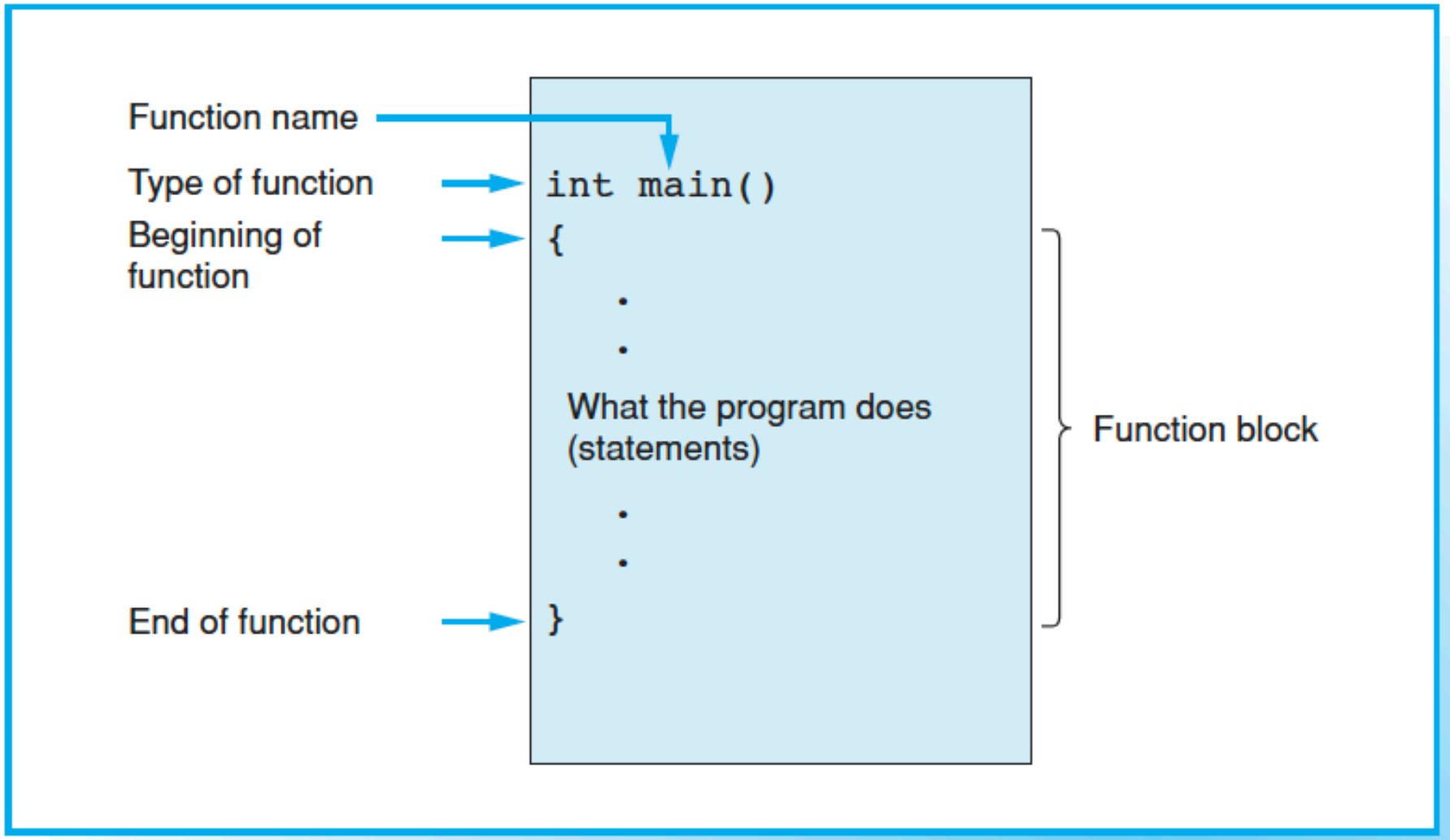
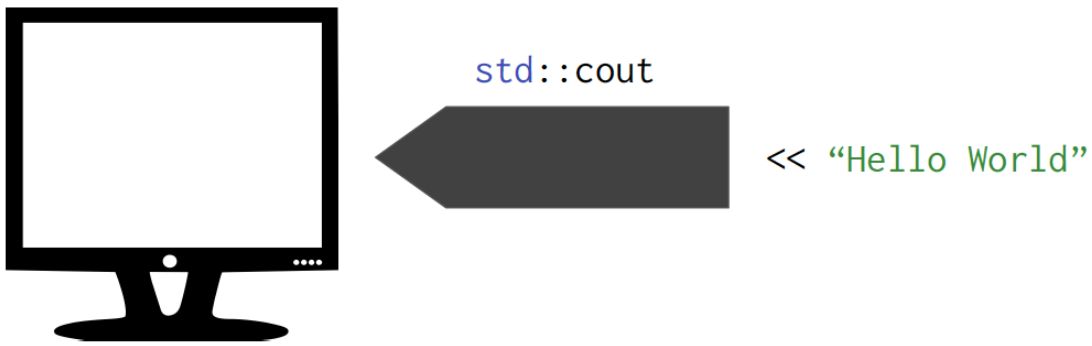


Image Reference : A Complete Guide to Programming in C++

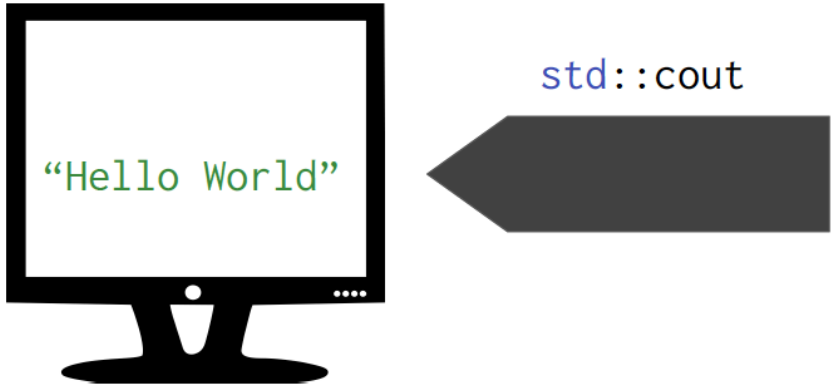
Streams

- 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
```

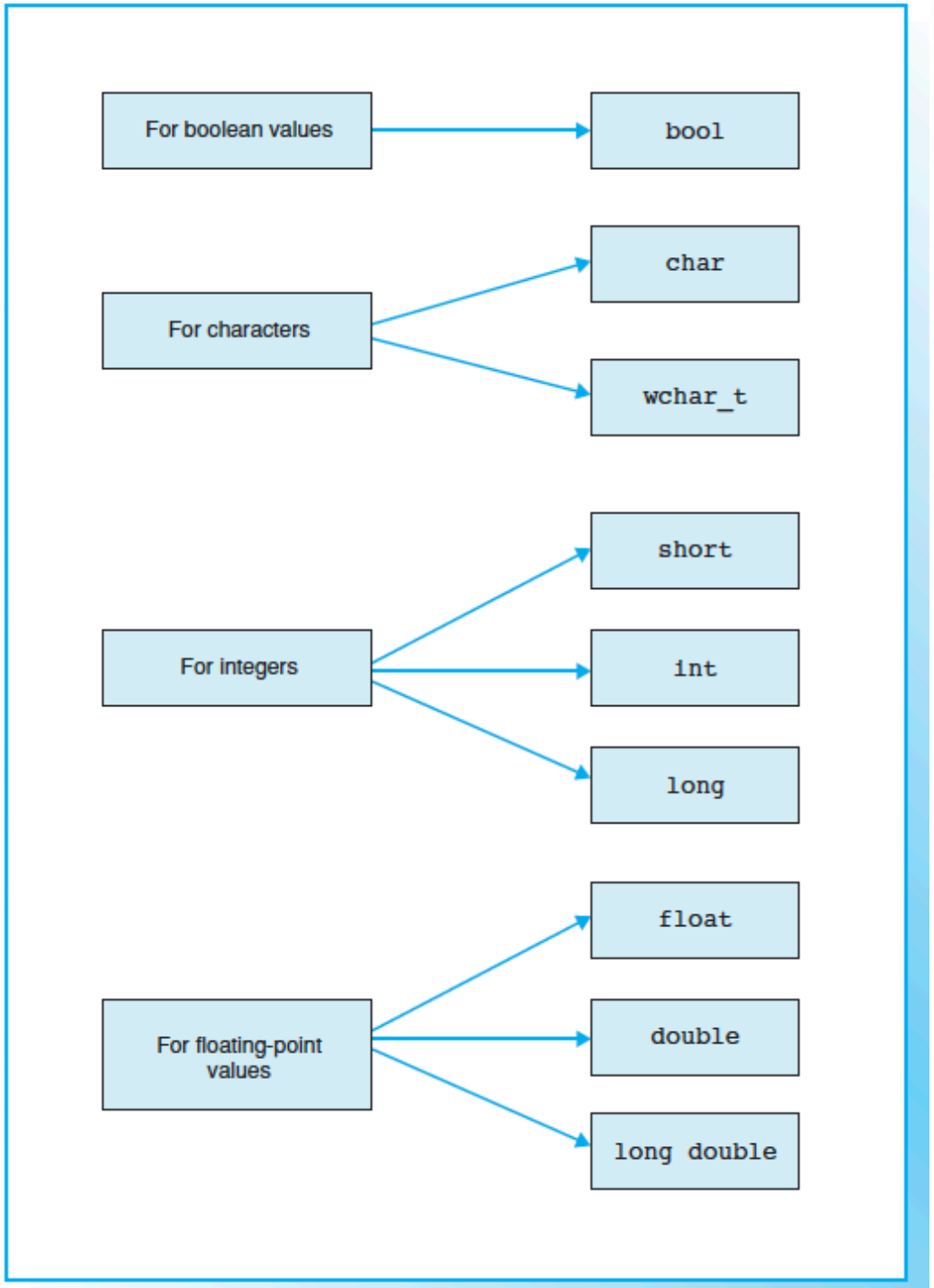


Image Reference : A Complete Guide to Programming in C++

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

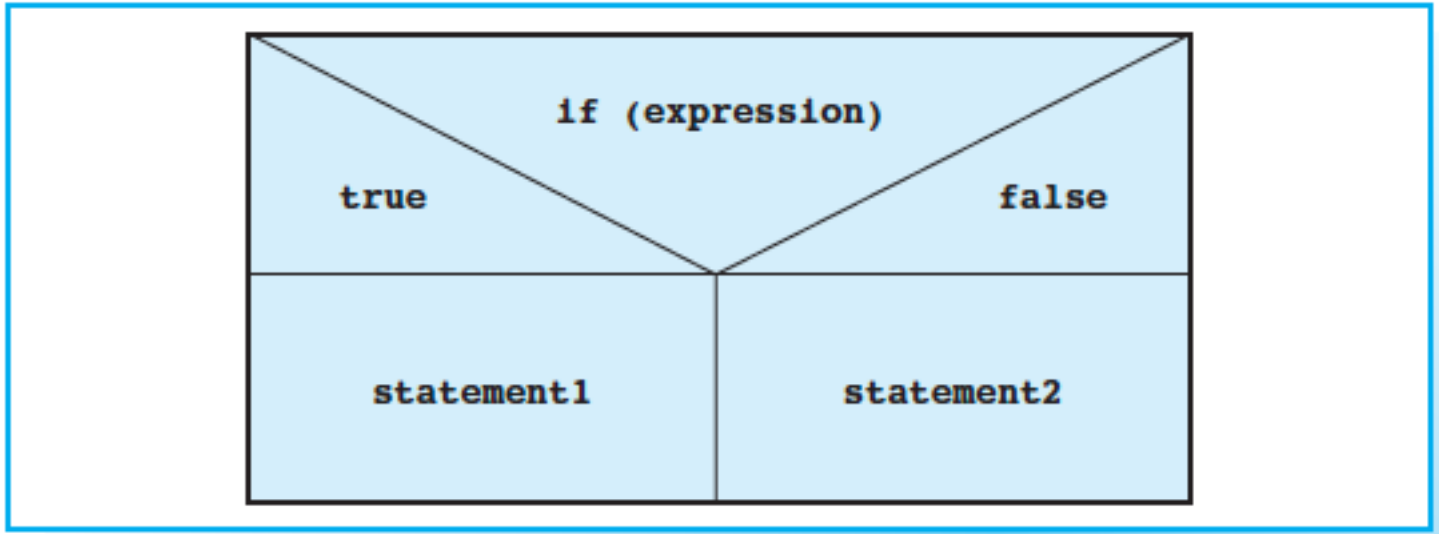
Operator	Significance
+	Addition
–	Subraction
*	Multiplication
/	Division
%	Remainder

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

IF ELSE



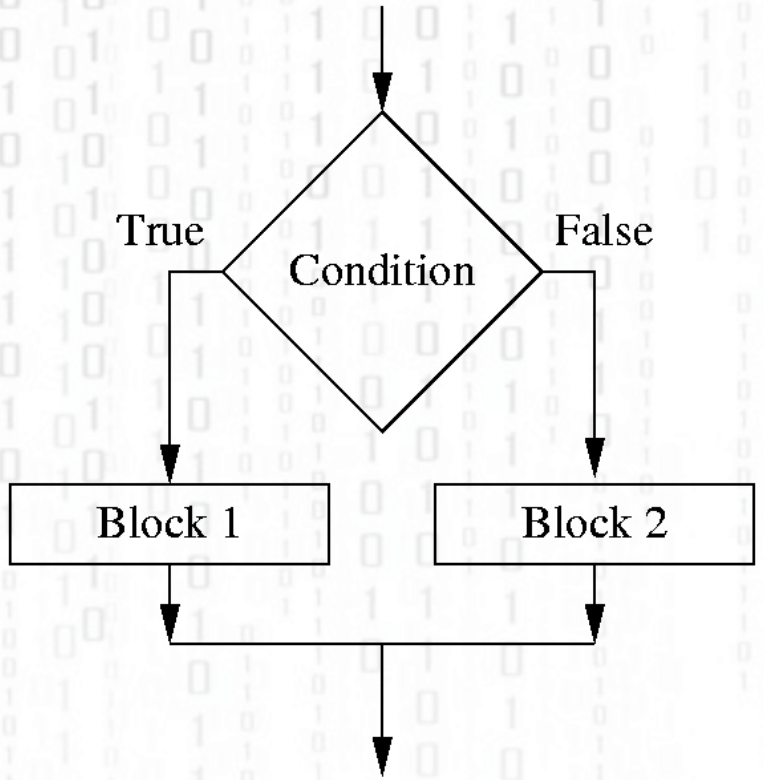


Image Reference : A Complete Guide to Programming in C++

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;
}
```