



Operators, Datatypes & Variables

Prepared and presented
by Daniel Gordon




Overview:

1. Operators

2. Datatypes

3. Variables

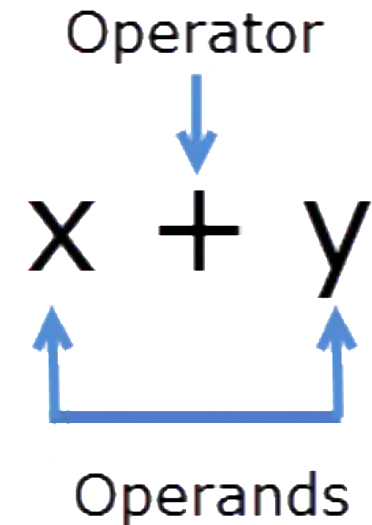




What are Operators?

Operators

- Symbols performing mathematical and logical computations operands
- Operands are the input to a process (mathematical or logical)
- Operators execute computation on operands



Operators in C

	Operator	Type
Unary operator →	+ +, - -	Unary operator
Binary operator {	+, -, *, /, %	Arithmetic operator
	<, <=, >, >=, ==, !=	Relational operator
	&&, , !	Logical operator
	&, , <<, >>, ~, ^	Bitwise operator
	=, +=, -=, *=, /=, %=	Assignment operator
Ternary operator →	?:	Ternary or conditional operator

Let's Code With Operators



```
#include<stdio.h> // source: https://overiq.com/1c-programming-101/arithmetic-operators-in-c/
int main() {
    //Here we use printf instead of iostream
    //It's up to you as to which one you'd like to use
    //Declare and initialize variable a and b
    int a = 7, b = 7; //Use this line for now, it'll become clearer later
    printf("a + b = %d\n", a + b); // 142
    printf("a - b = %d\n", a - b); //128
    printf("a * b = %d\n", a * b); //??
    printf("a / b = %d\n", a / b);
    // because both operands are integer result will be an integer
    printf("a %% b = %d\n", a % b); // % operator returns the remainder of 11/4 i.e 3
    // We want to find what a not equal to be using the operator: !=
    printf("a != b = %d\n", a != b);
    printf("a == b = %d\n", a == b);
    printf("a && b = %d\n", a && b);
    printf("a & b = %d\n", a & b);
    printf("a *= b %d\n", a *= b);
    printf("a = %d\n", a);
    printf("b = %d\n", b);
    return 0; // Signal to operating system everything works fine
}
```

Data...types?



DataTypes in C / C++

Primary

Integer

Character

Boolean

Floating Point

Double Floating Point

Void

Derived

Function

Array

Pointer

Reference

User Defined

Class

Structure

Union

Enum

Typedef

Primitive DataTypes

Primary

Integer

```
int a;
```

Character

```
char b;
```

Boolean

```
bool c;
```

Floating Point

```
float d;
```

Double Floating Point

```
double e;
```

Void

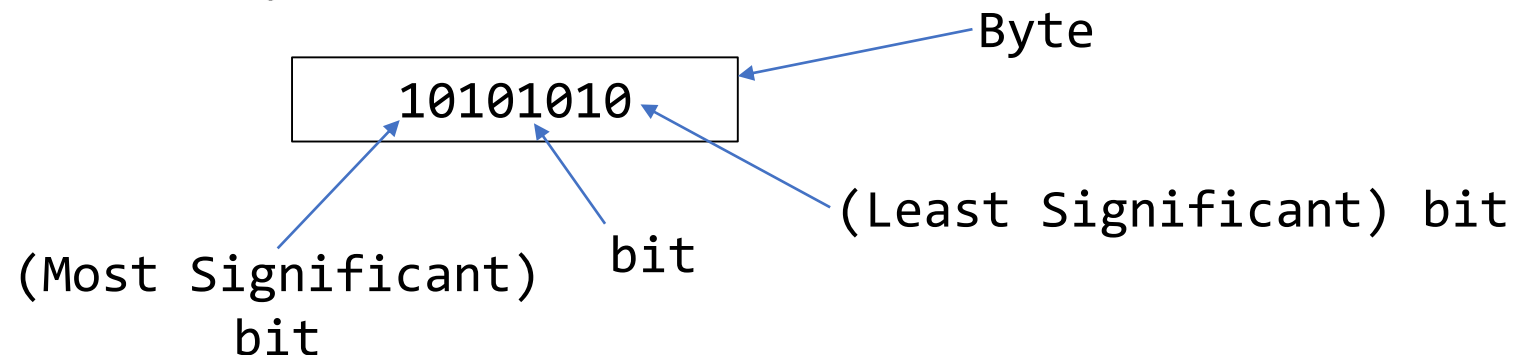
```
void ;
```

Wide Character

```
wchar_t f;
```

Data has a special representation

- Computers can only understand binary, i.e. '1' and '0'
- These values can be read as High or Low, True or False and often refers to logic levels.
- The code that you write is compiled straight down to binary – Hopefully you watched that [video](#) from the first presentation
- A group of 8 bits is called a byte



Data has a special representation

- A byte is 8 bits, which is $2^8 = 256$, which means the highest number we can represent is 256!
- That's not really useful, so how do computers work with higher numbers or even decimals?
- Fortunately though the ASCII table (binary encoding for letters, numbers) only goes up to 256
- But that's only the English alphabet, so, we're in the same issue again.

Primitive DataTypes: Sizes

```
std::cout << "Int = " << sizeof(int) << " byte(s)" << std::endl;  
std::cout << "Char = " << sizeof(char) << " byte(s)" << std::endl;  
std::cout << "Float = " << sizeof(float) << " byte(s)" << std::endl;  
std::cout << "Double = " << sizeof(double) << " byte(s)" << std::endl;  
std::cout << "Void = " << "0" << " byte(s)" << std::endl;  
std::cout << "Wide Character = " << sizeof(double) << " byte(s)" << std::endl;
```

Int = 4 byte(s)

Char = 1 byte(s)

Float = 4 byte(s)

Double = 8 byte(s)

Void = 0 byte(s)

Wide Character = 8 byte(s)

Modifiers

- Manipulate datatypes' data representation or storage capacity to contain more, less or specific types of numbers.

Modifiers in C++



DATA TYPE	SIZE (IN BYTES)	RANGE
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	$-(2^{63})$ to $(2^{63})-1$
unsigned long long int	8	0 to 18,446,744,073,709,551,615
signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	
wchar_t	2 or 4	1 wide character

Note : Above values may vary from compiler to compiler. In above example, we have considered GCC 64 bit.

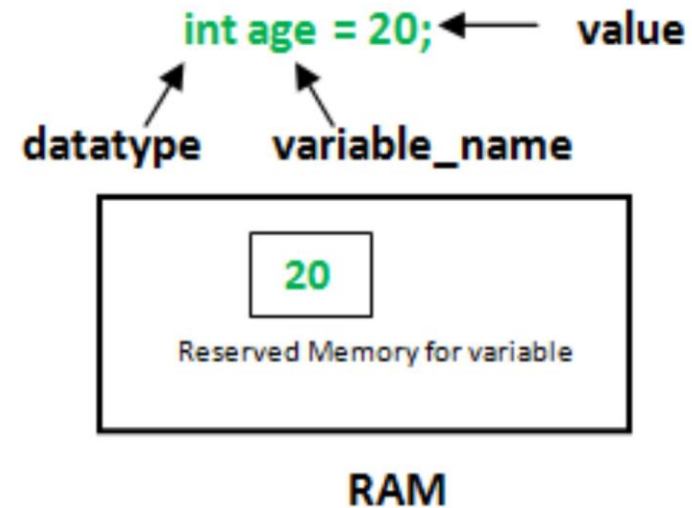


Variables

Variables

- A referable name given to a memory location
- Allows operators to locate and modify the information at that location.

Variables in C++



There are three types in C++:

1) Local:

-> Variable(s) declared in a block of code

2) Instance:

-> Variable(s) defined outside a function or class

3) Static:

-> Variable(s) declared and located similarly to an instance one. However, it includes the keyword "static" and is stored differently.

Types of variables in C++

```
class GFG {
```

```
public:
```

```
static int a;
```

— **Static Variable**

```
int b;
```

— **Instance Variable**

```
public:
```

```
func()
```

```
{
```

```
int c;
```

— **Local Variable**

```
};
```

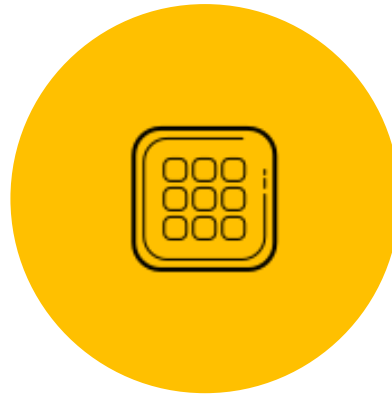
```
};
```



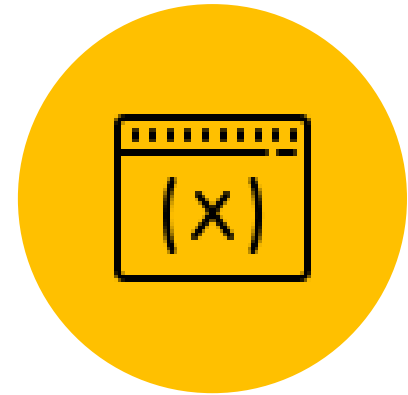
Recap:



OPERATORS



DATATYPES



VARIABLES