IEEE
Kenyatta University
Student Branch

WORKSHOP 1

BEGINNING C++

PROGRAMMING SERIES

ABOUT ME

ALLAN K. KOECH

4th Year Biomedical Engineering Student

Chair IEEE Kenyatta University Student Branch

Chair IEEE Photonics Society Kenyatta University

CInO at THE AFECS LTD.

WHAT WE WILL COVER TODAY

1. What is C++
2. A bit of history of C++
3. Keywords to know (Compiler, Assembler, etc)
4. The environment (IDE & setup)
5. Syntax
6. Data types & variable types
7. Example codes

WHAT IS C++?

- Just like Python, C, JS, Go, PHP, etc, C++ is a programming language
- C++ is a middle-level programming language
- Runs on a variety of platforms like Windows, MaC, Linux, (*nix), etc.
- C++ is a general-purpose language
- Is a compiled language not interpreted

where is it used?

Used in many fields, areas, applications, etc

1. Operating systems : Most use comination of C/C++ be it Windows, Linux, Mac, Android, etc
   Linux Kernel : ***https://github.com/torvalds/linux***

2. Browsers - Rendering engines for browsers utilize C++ ie

   Chrome V8 engine - ***https://github.com/v8/v8***

3. Libraries - ML, Computer Vision, etc ie Tensorflow, opencv

   https://github.com/tensorflow/tensorflow, ***https://github.com/opencv/opencv***

4. Databases like **MySQL, PostgreSQL**

where is it used?

5. Embedded systems - Arduino, STM32, Pico, etc
6. Compilers - Many languages have their compilers witten in C/C++ ie CPython for python,
7. Banking systems
8. Medical devices

why use C++?

1. Speed
2. Close to hardware
3. Concurrency support
4. Statically typed

BUT, learning C++ may be harder than learning other higher level languages

HISTORY OF C++

Developed in 1979 by Bjarne Stroustrup at bell laboratories of AT&T (American Telephone & Telegraph), located in U.S.A.

Called a superset of C, it means any valid C program is also a valid C++ program

1983, name changed from 'C with classes' to C++

2011 new standard version was released called C++11

simula

TERMINOLOGIES

Compiler - converts instructions into machine-code or low-level form so that they may be executed by the machine

Assembler - Program that converts assembly language to machine code

Interpreter - Program that directly executes instructions in higher level languages / scripting lang without requiring prior compilation

Compiled language vs interpreted language

Compiled: C, **C++**, Go, Rust, ...
Interpreted: Python, JS, PHP, Perl, ...

So, what do you need to write and run cpp files?

1. A text editor ie Notepad++, VSCode, notepad, etc
2. A compiler ie gcc, msvc, mingw, clang, etc
3. An IDE - like Visual Studio, DevC++, Eclipse C++, Qt Creator, etc

Setting up our environment

Online GDB at ***https://www.onlinegdb.com/online_c++_compiler***

Run Debug Stop Share Save { } Beautify Language C++

main.cpp

```
/******************************************************************************

                              Online C++ Compiler.
                   Code, Compile, Run and Debug C++ program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <iostream>

using namespace std;

int main()
{
    cout<<"Hello World";

    return 0;
}

```

input

Command line arguments:

Standard Input: Interactive Console Text

Huuh! Fear NOT

Boilerplate code!

main.cpp

flashtext > main.cpp > ...

```cpp
/*
This is a basic HEllo World example
by ME
*/

#include <iostream>

using namespace std;

// The main function
int main()
{
    cout << "Hello World!" << endl;

    return 0;
}
```

The output

```
lalanke@devqt:~/Downloads/Com
Hello World!
lalanke@devqt:~/Downloads/Com
```

let's break it down

File ends with a **.cpp** or **.cc**, in our case ***main.cpp***

1 & **4** are comments

2 is the preprocessor command to include iostream library to our program

3 Tells the compiler to use the std namespace

5 The entry point to your application, all C/C++ files start execution at the main function.

main.cpp

flashtext > main.cpp > ...

```
1  /*
2  This is a basic HEllo World example
3  by ME
4  */
5
6  #include <iostream>
7
8  using namespace std;
9
10 // The main function
11 int main()
12 {
13     cout << "Hello World!" << endl;
14
15     return 0;
16 }
```

1 2 3 4 5 6

let's break it down

5 *return 0;* Terminates the main function and causes it to exit with a 0 code

cout << "Hello World" << endl;

cout - command part of the iostream library that causes the "Hello World" to be displayed on the console

endl - command part of the iostream library- end line, similar to '\n'

Primitive data types

Type	Keyword
Boolean	bool
Character	char
Integer	int
Floating point	float
Double floating point	double
Valueless	void
Wide character	wchar_t

Variable definition

data_type variable_name;

```
int x;
char v;
float distance;
```

Variable instantiation

variable_name = value;

```
x =5;
v = 'A';
distance = 34.5;
```

Or simply

```
int age = 5;
```

Rules for variables

You cant use keyword as a variable

Can comprise of AZaz_09

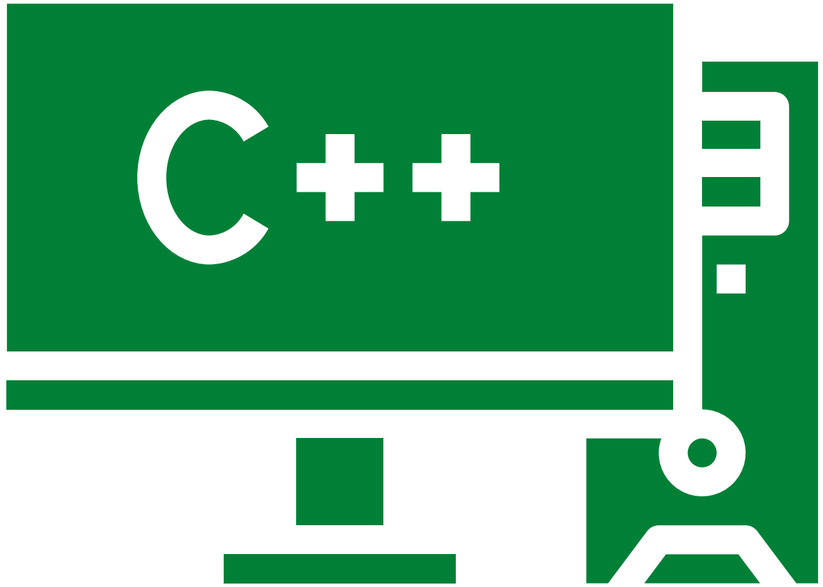
They are case sensitive

Can only start with a letter or an _,
not a number

alignas (since C++11)	decltype (since C++11)	reflexpr (reflection TS)
alignof (since C++11)	default (1)	register (2)
and	delete (1)	reinterpret_cast
and_eq	do	requires (since C++20)
asm	double	return
atomic_cancel (TM TS)	dynamic_cast	short
atomic_commit (TM TS)	else	signed
atomic_noexcept (TM TS)	enum	sizeof (1)
auto (1)	explicit	static
bitand	export (1) (3)	static_assert (since C++11)
bitor	extern (1)	static_cast
bool	false	struct (1)
break	float	switch
case	for	synchronized (TM TS)
catch	friend	template
char	goto	this
char8_t (since C++20)	if	thread_local (since C++11)
char16_t (since C++11)	inline (1)	throw
char32_t (since C++11)	int	true
class (1)	long	try
compl	mutable (1)	typedef
concept (since C++20)	namespace	typeid
const	new	typename
consteval (since C++20)	noexcept (since C++11)	union
constexpr (since C++11)	not	unsigned
constinit (since C++20)	not_eq	using (1)
const_cast	nullptr (since C++11)	virtual
continue	operator	void
co_await (since C++20)	or	volatile
co_return (since C++20)	or_eq	wchar_t
co_yield (since C++20)	private	while
	protected	xor
	public	xor_eq

Lets Code! It's fun.

Let's use the online gdb to write some code



https://www.onlinegdb.com/online_c++_compiler

What we have looked at today

What is C++
Its origins
Its Syntax
Common terms
Data types & Variable types
Basic Arithmetics

You will find today's code and an exercise in the folder Workshop1/main.cpp here https://github.com/lalan-ke/IEEE_KU-beginning-cpp

IEEE
Kenyatta University
Student Branch

@lalan_KE
@Ku_Photonics
@ieeewie_ku
@IEEEEmbsKU
@ieee_pes_ku

THANK YOU