UNIVERSITAS
GADJAH MADA

LOCALLY ROOTED,

GLOBALLY RESPECTED

Meeting 2

Overview of C++

Fundamentals of Programming
TKU211131

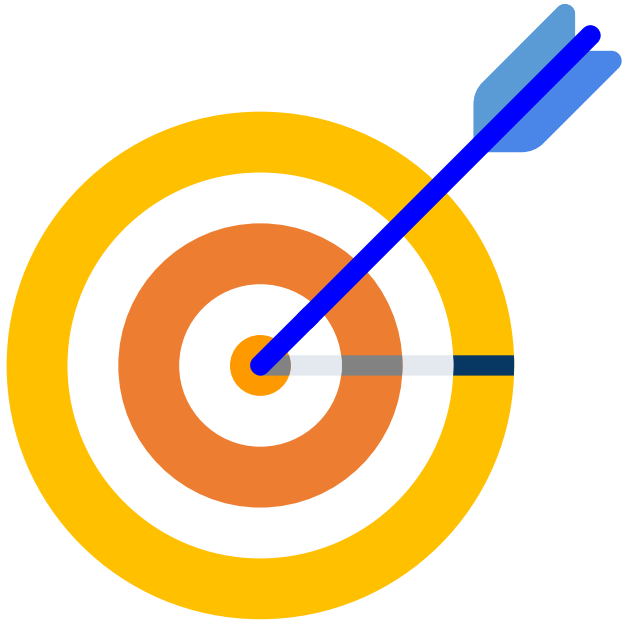
DTETI FT UGM

Bimo Sunarfri Hantono
bhe@ugm.ac.id

Topics

UNIVERSITAS GADJAH MADA

- Programming Languages Overview
- Programming Language Trends
- Introduction to C++
- IDEs and Text Editors
- Basic Structure of C++ Program



UNIVERSITAS
GADJAH MADA

Programming Languages Overview

ugm.ac.id

LOCALLY ROOTED, GLOBALLY RESPECTED

What is a Programming Language?

UNIVERSITAS GADJAH MADA

Definition:

- A formal way to communicate with computers
- Contains instructions for specific tasks
- Examples: calculate numbers, display text, control hardware

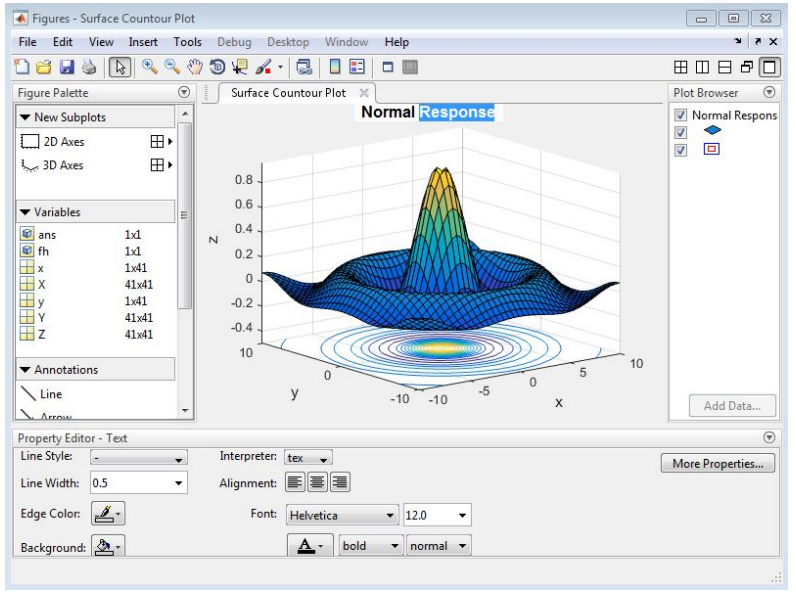
Purpose of Programming Languages

- Automate tasks
- Solve problems
- Create software & applications
- Control hardware

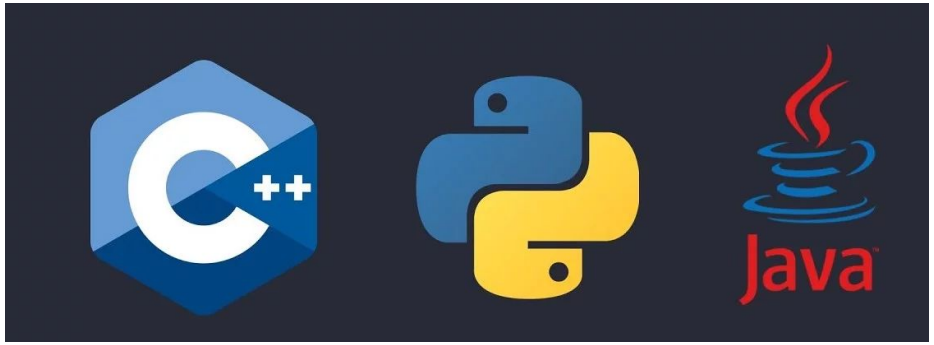
Specific-Purpose vs General-Purpose

UNIVERSITAS GADJAH MADA

- **Specific-purpose**: Designed for a narrow set of tasks
  - Examples: SQL (databases), MATLAB (math)



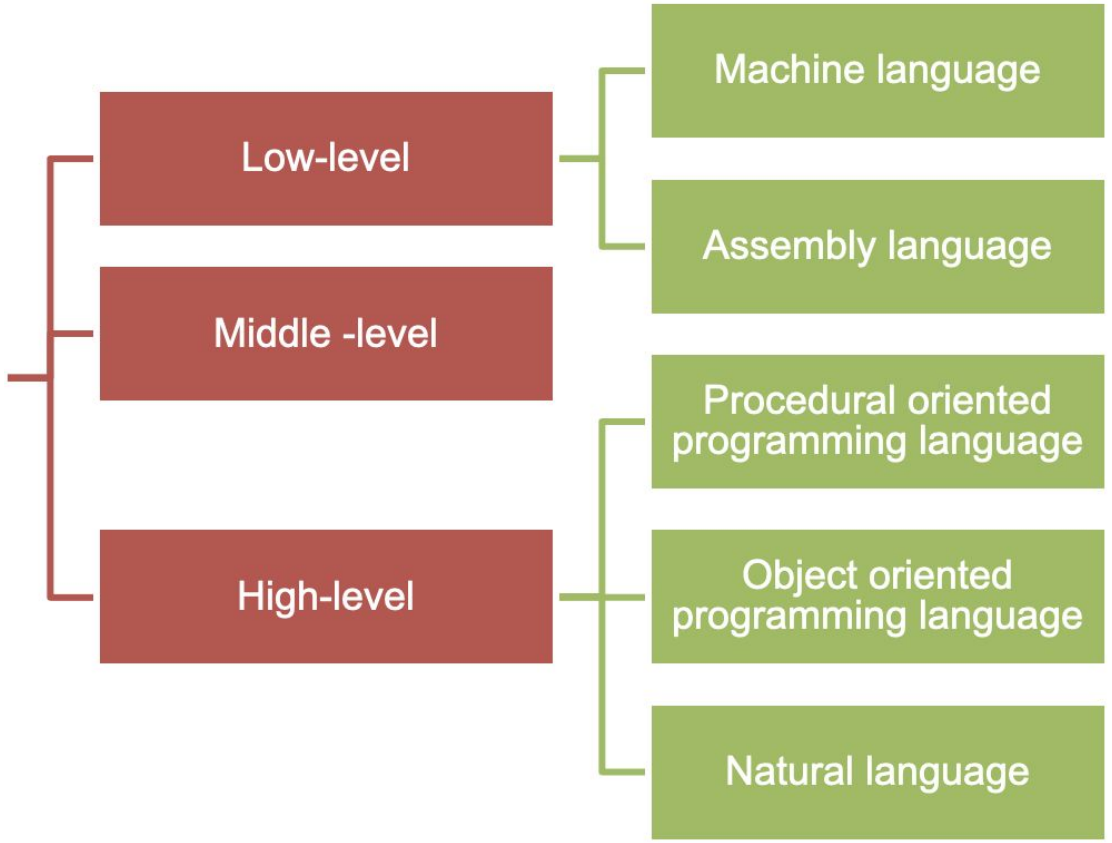
- **General-purpose**: Works for many types of tasks
  - Examples: C, Python, Java



Levels of Programming Languages

UNIVERSITAS GADJAH MADA

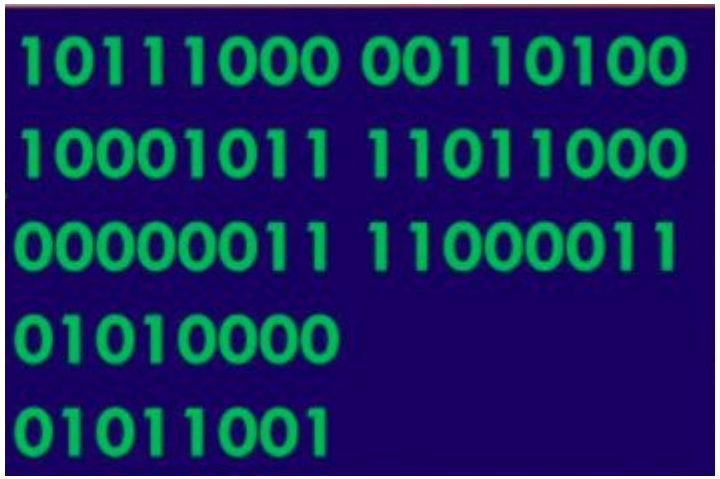
- **Low-Level:** Close to machine language, high performance, hard to read
- **Middle-Level:** Balance between low-level control and high-level readability
- **High-Level:** Easy to read, further from hardware



Example of Low Level Languages

UNIVERSITAS GADJAH MADA

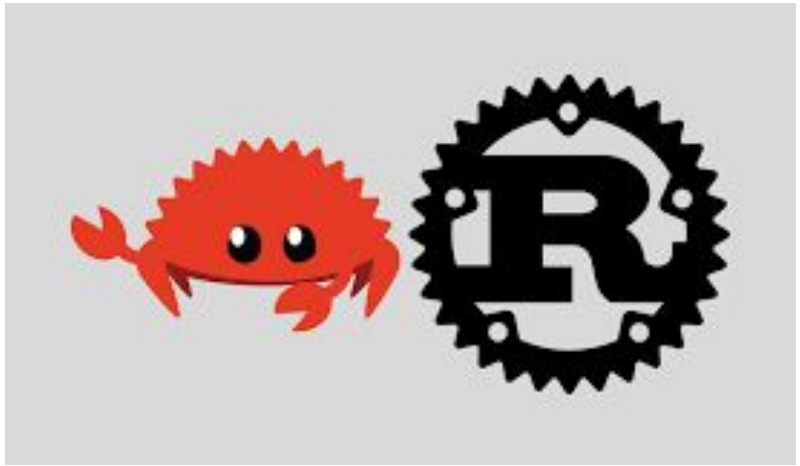
Language / Term	Application Area
Machine Code (Binary)	Boot sectors, bare-metal firmware, micro-optimizations
Assembly (x86/ARM/RISC-V)	Device drivers, ISRs/firmware, high-performance routines



Example of Middle Level Languages

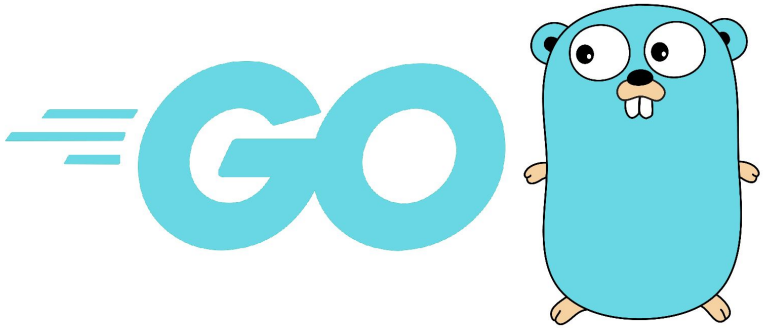
UNIVERSITAS GADJAH MADA

Language / Term	Application Area
C	Systems programming, OS/kernel, embedded firmware, network stacks
C++	Game engines, CAD/3D, browsers, low-latency trading, HPC
Rust	CLIs/services, OS components, embedded (*no_std*), WebAssembly



Example of High Level Languages (1)

UNIVERSITAS GADJAH MADA



Example of High Level Languages (2)

UNIVERSITAS GADJAH MADA

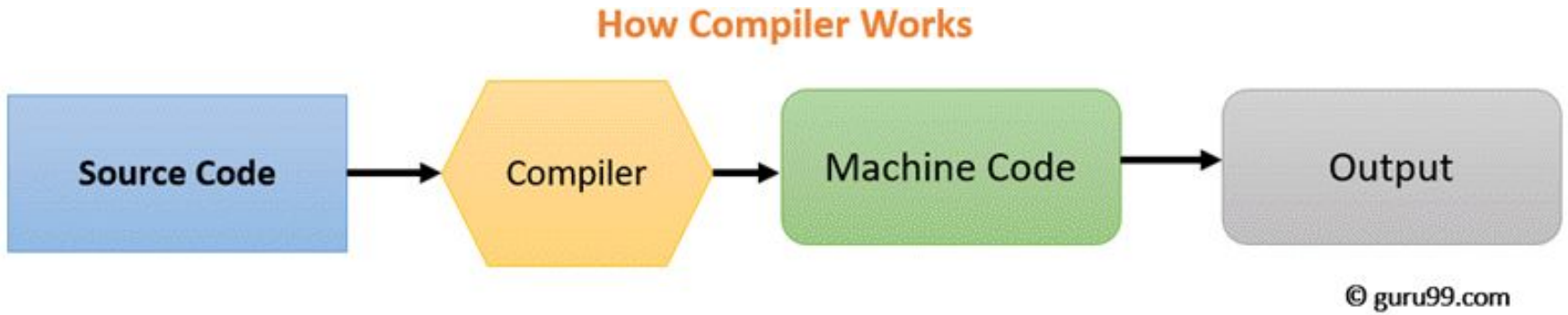
Language / Term	Application Area
Python	Data/ML, automation/scripting, web backends
Java	Enterprise backends, early Android, distributed systems
Javascript	Web front-end, Node.js backends
Golang	Cloud services, networking/DevOps tools, CLIs
C# (.NET)	Enterprise apps, ASP.NET, Unity game dev
Kotlin	Modern Android, server-side JVM

Compiled vs Interpreted Languages (1)

UNIVERSITAS GADJAH MADA

Compiled: Translated into machine code before execution. Examples: C, C++

Interpreted: Executed line-by-line at runtime. Examples: Python, JavaScript



Compiled vs Interpreted Languages (2)

UNIVERSITAS GADJAH MADA

Feature	Compiled	Interpreted
Execution Speed	Faster	Slower
Portability	Lower (depends on platform)	Higher (runs anywhere with interpreter)
Debugging	Harder (needs recompile)	Easier (runtime feedback)
Memory Usage	More efficient	Less efficient
Flexibility	Lower	Higher

UNIVERSITAS
GADJAH MADA

Programming Languages Trend

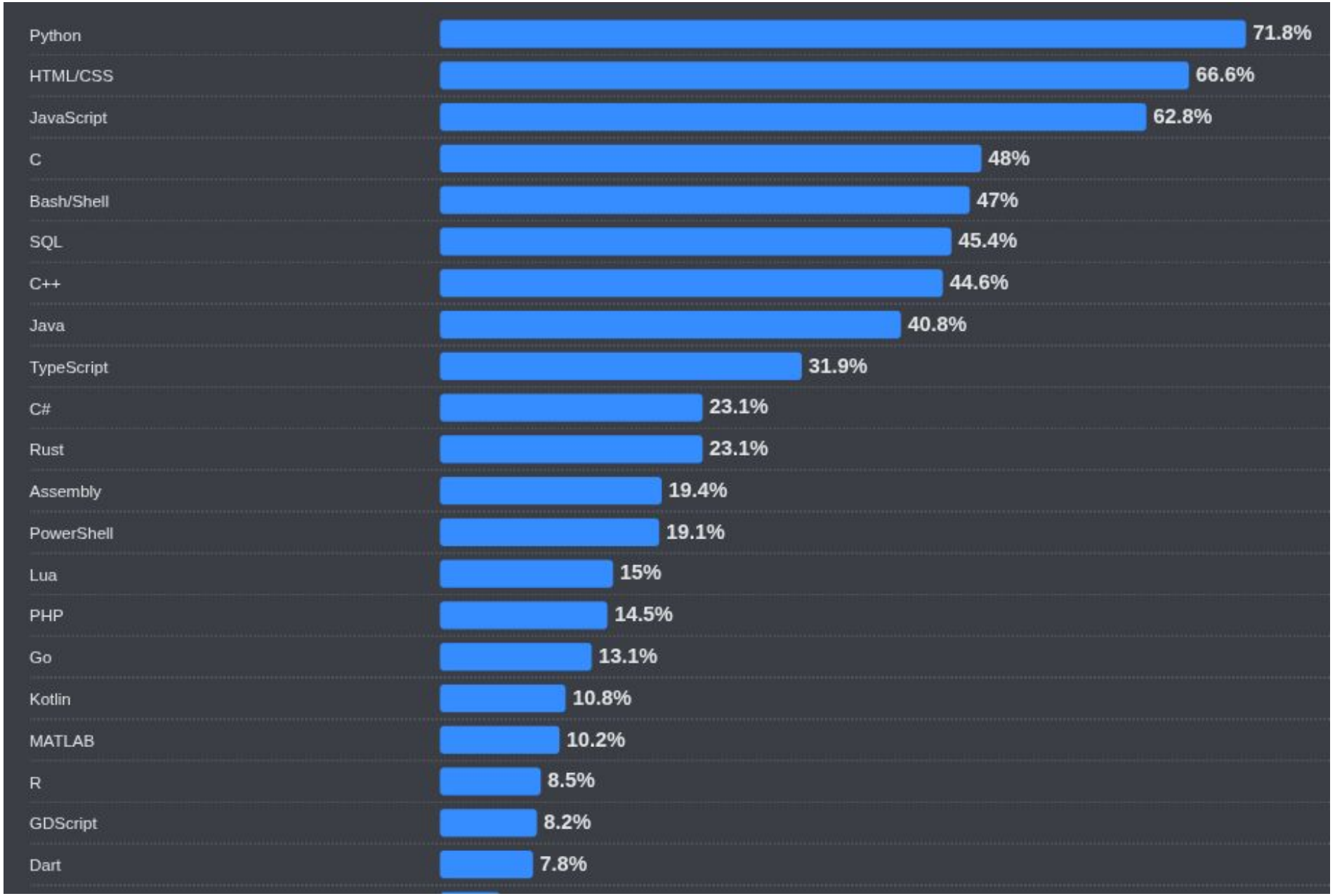
ugm.ac.id

LOCALLY ROOTED, GLOBALLY RESPECTED

Why Language Trends Matter?

UNIVERSITAS GADJAH MADA

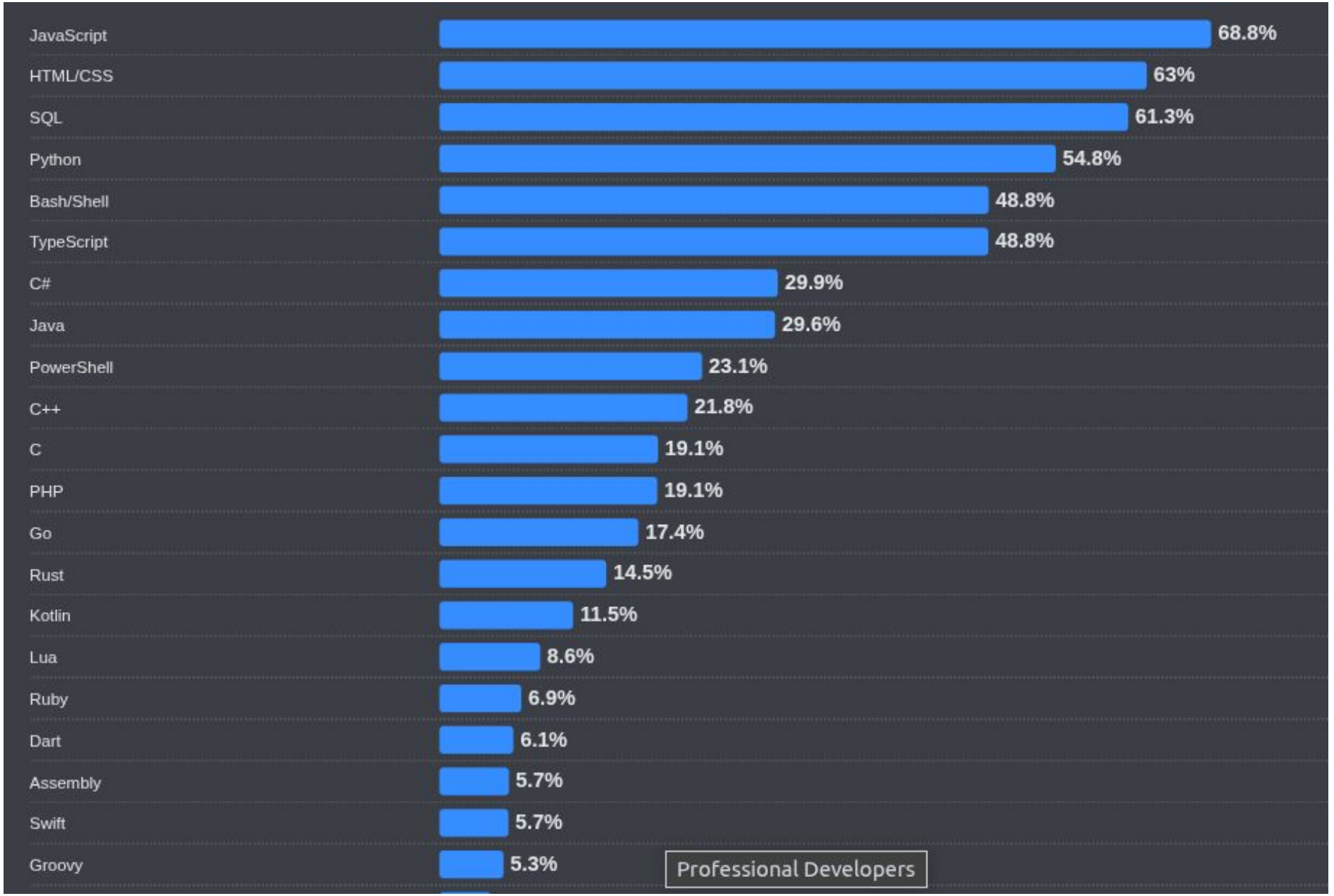
- Stay relevant in the job market
- Access to community and learning resources
- Better long-term career opportunities
- Ecosystem maturity and stability



UNIVERSITAS GADJAH MADA

Most Popular Programming Language? (Learning to Code)

https://survey.stackoverflow.co/2025/technology#most-popular-technologies-language-learn



UNIVERSITAS GADJAH MADA

Most Popular Programming Language? (Professional Developer)

https://survey.stackoverflow.co/2025/technology#most-popular-technologies-language-prof

Mini Discussion

If you had to choose one language to learn this year, which would it be and why?

UNIVERSITAS
GADJAH MADA

Introduction to C++

ugm.ac.id

LOCALLY ROOTED, GLOBALLY RESPECTED

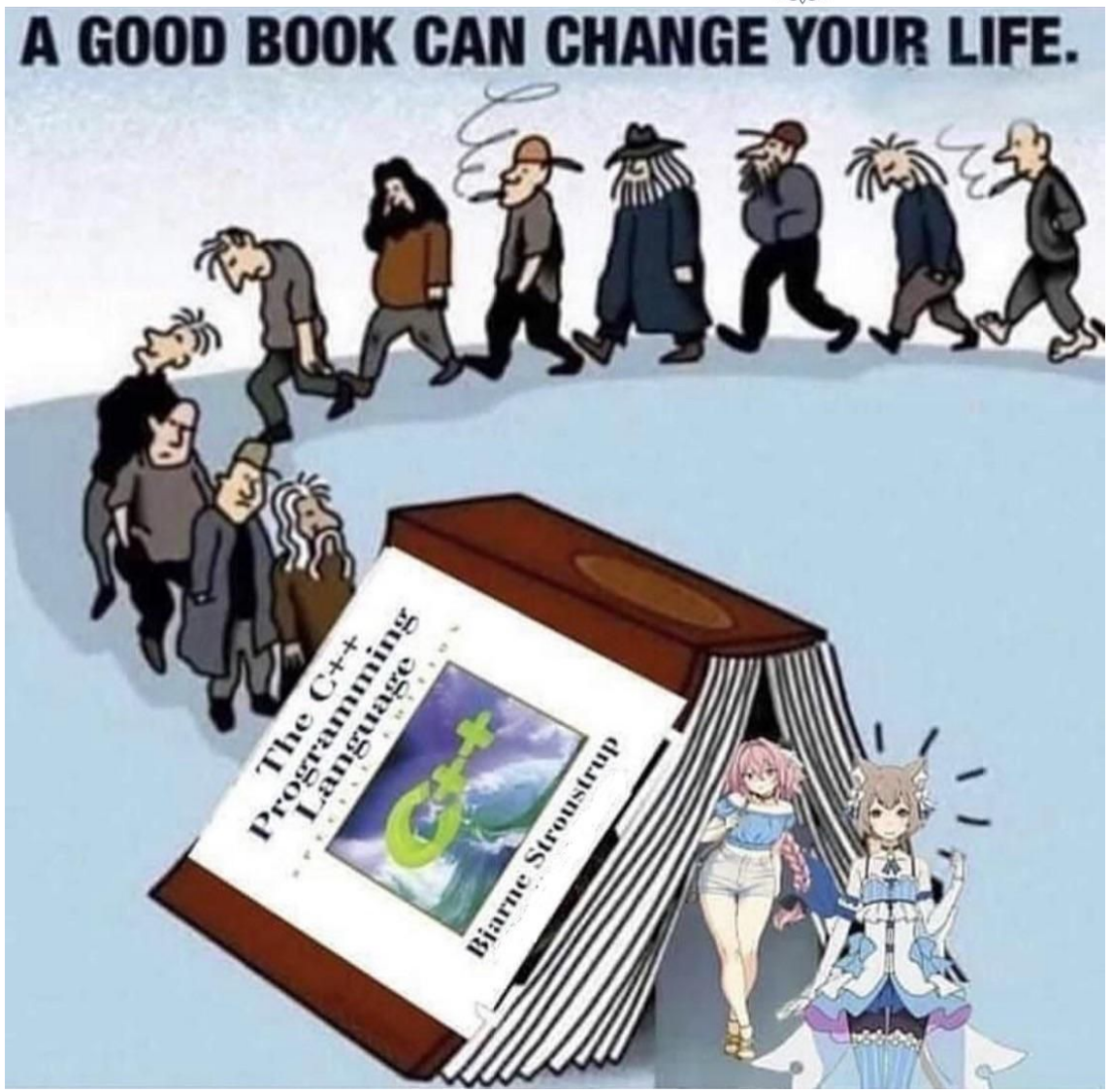
What is C++ ?

UNIVERSITAS GADJAH MADA

- General-purpose, compiled, middle-level programming language
- Developed by **Bjarne Stroustrup** in 1979 at Bell Labs
- Extension of the C programming language with object-oriented features

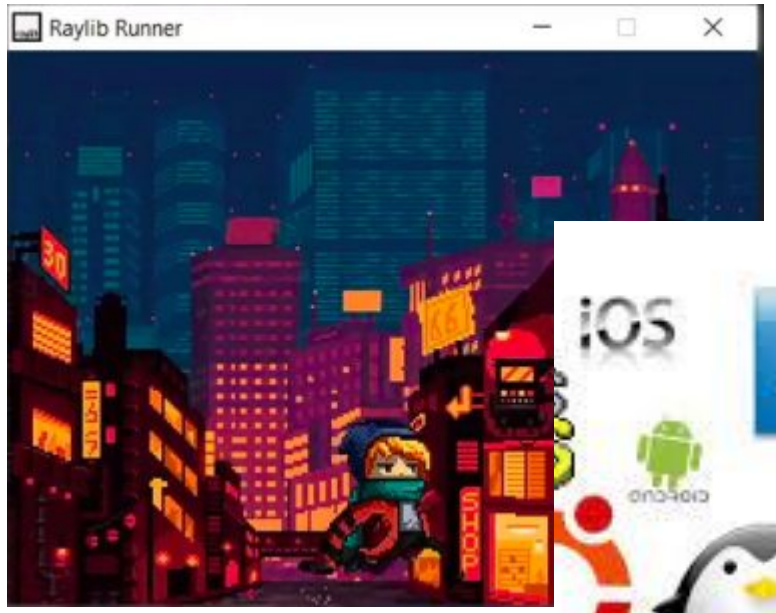
Why Learn C++ ?

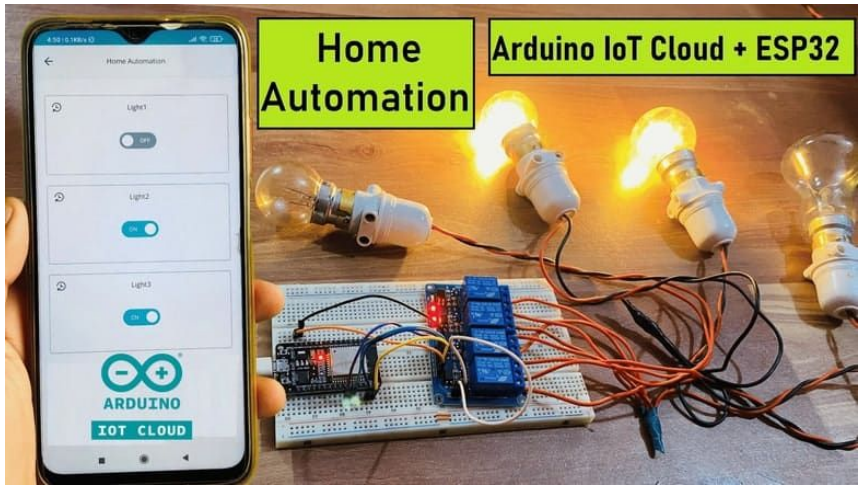
- High performance and efficiency
- Direct access to memory and hardware
- Used in critical systems and performance-heavy applications
- **Strong foundation** for learning other languages



Where C++ is Used

- Game development (Unreal Engine)
- Operating systems (Windows components, Linux parts)
- Embedded systems (IoT devices, firmware)
- High-frequency trading systems
- Scientific simulations



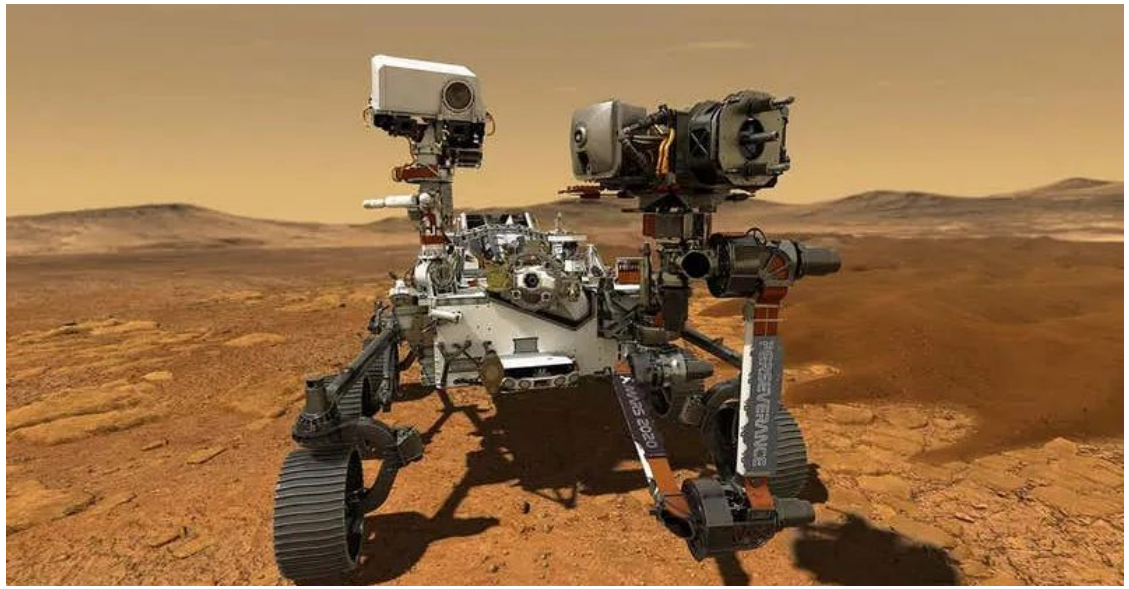


Strengths & Weaknesses

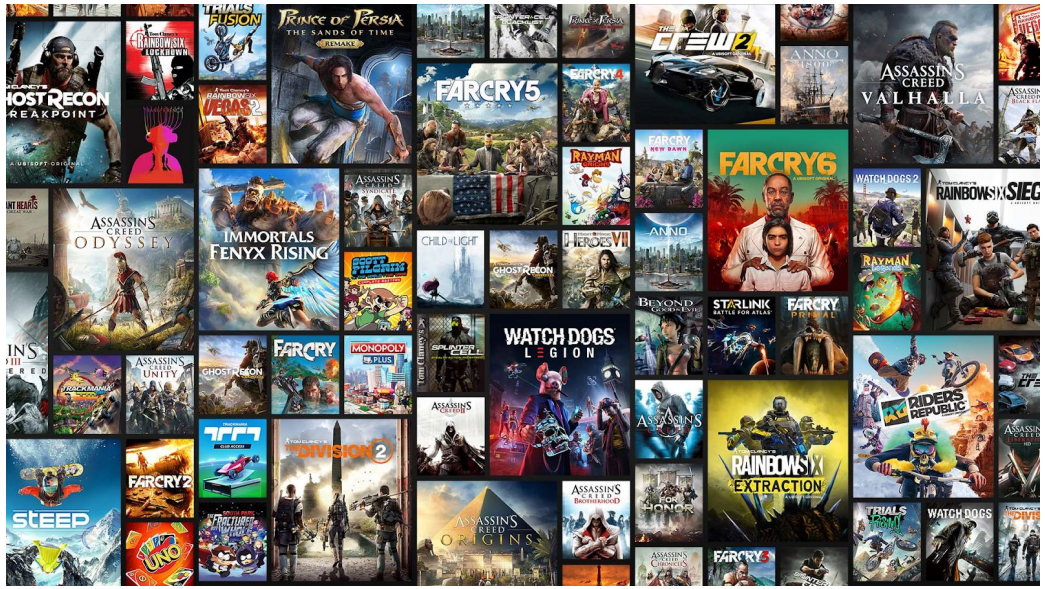
UNIVERSITAS GADJAH MADA

Strengths	Weaknesses
Fast execution	Steeper learning curve
Hardware control	Manual memory management risks
Versatile	More complex syntax
Large community	Slower development compared to higher-level languages

Fun Fact



NASA’s Mars Rovers use C++ for control systems



Many AAA games are built with C++

UNIVERSITAS
GADJAH MADA

IDEs and Text Editors

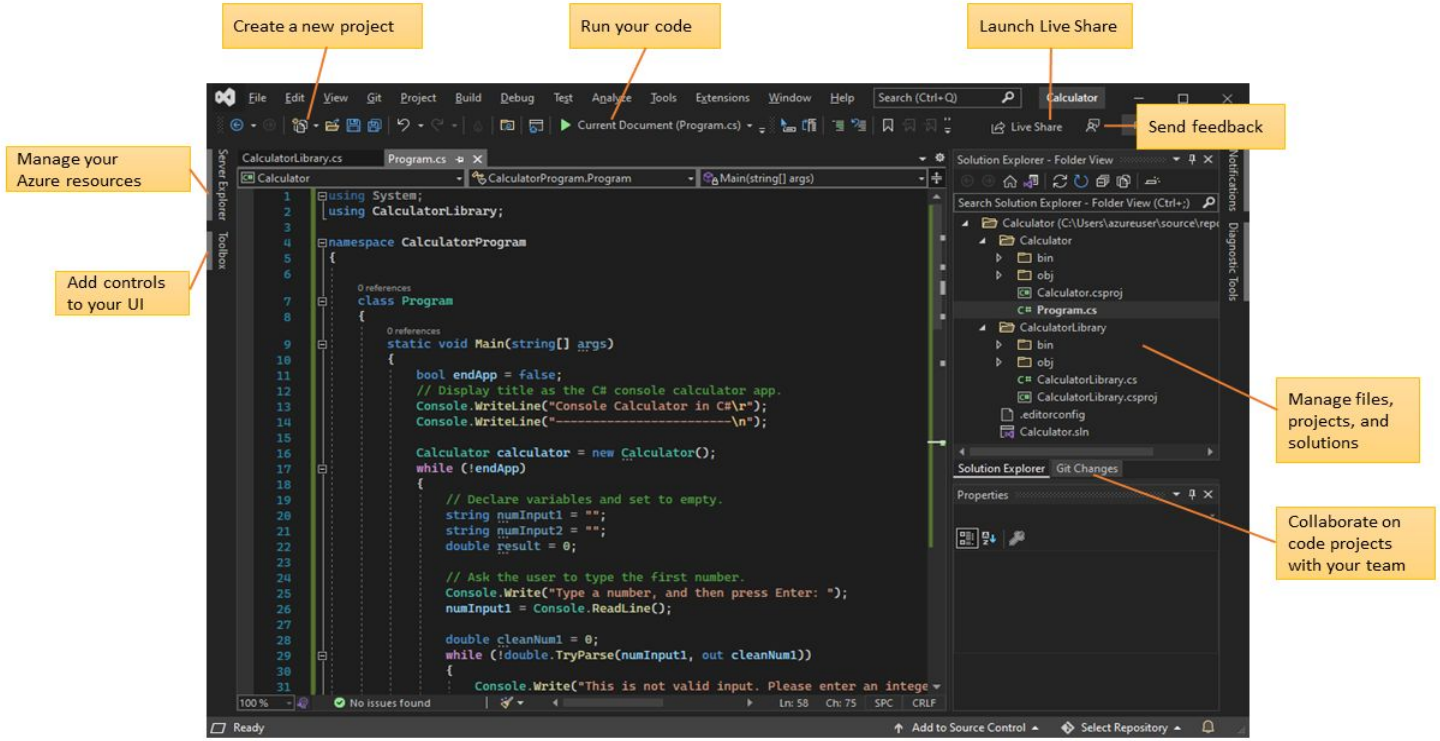
What is an IDE?

UNIVERSITAS GADJAH MADA

Integrated Development Environment

Combines multiple tools in one software:

- Code editor
- Compiler/interpreter
- Debugger
- Build automation



IDE vs Text Editor (1)

UNIVERSITAS GADJAH MADA

Visual Studio

vs.

VS Code

IDE vs Text Editor (2)

UNIVERSITAS GADJAH MADA

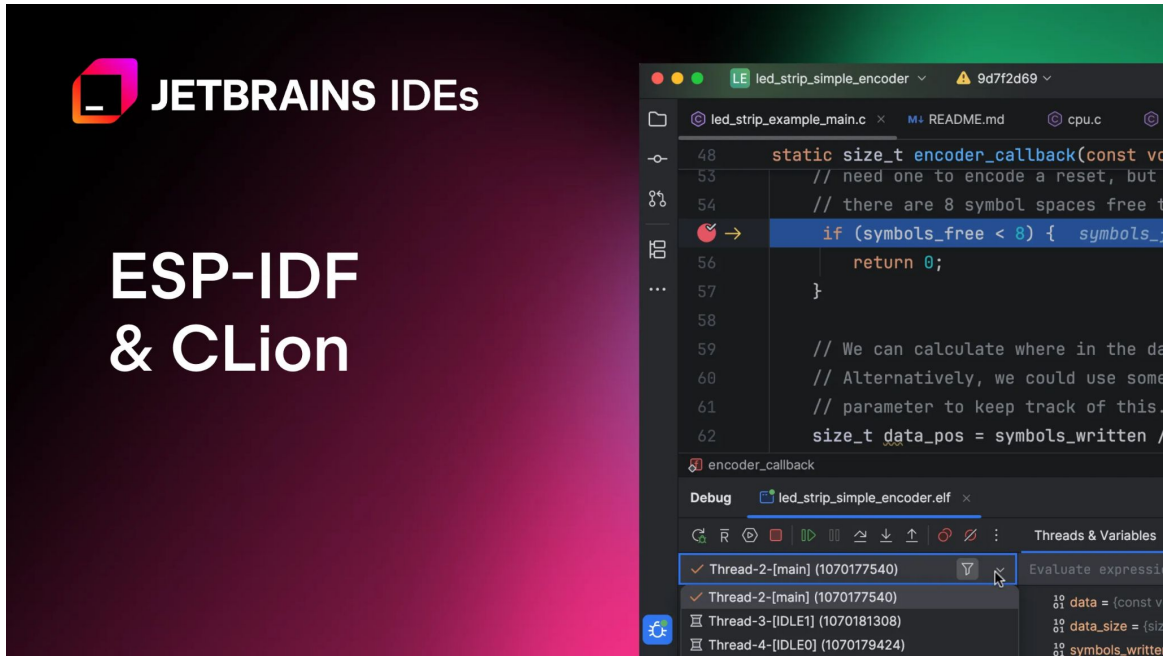
Feature	IDE	Text Editor
Tools	Complete suite (compiler, debugger)	Basic code editing
Setup	Ready for development	Needs manual setup
Speed	Heavier	Lightweight
Use case	Large projects	Quick edits, scripting

Popular C++ IDEs

UNIVERSITAS GADJAH MADA

- **Visual Studio** (Windows, full-featured)
- **CLion** (Cross-platform, smart code analysis)
- **Code::Blocks** (Lightweight, open-source)

Visual Studio

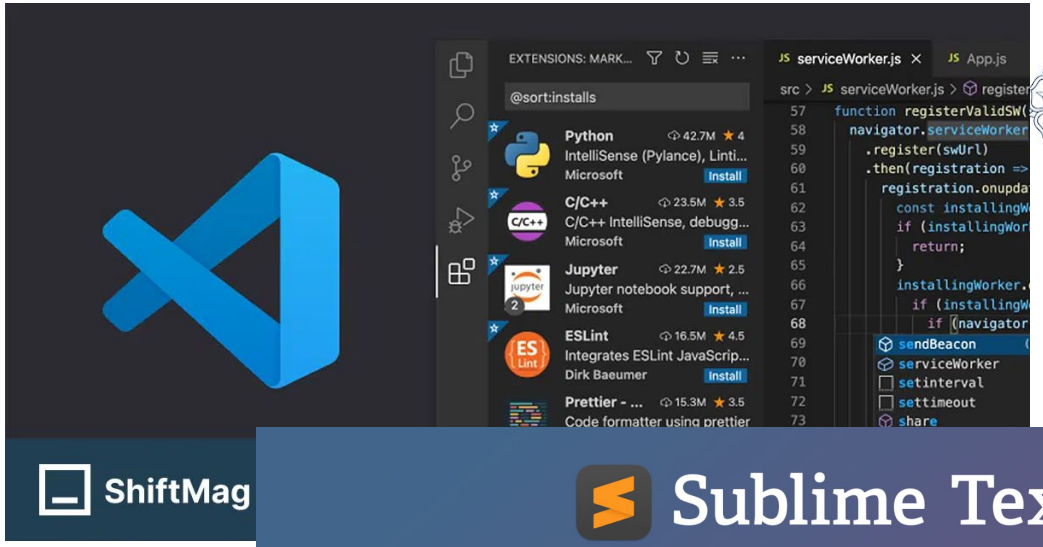


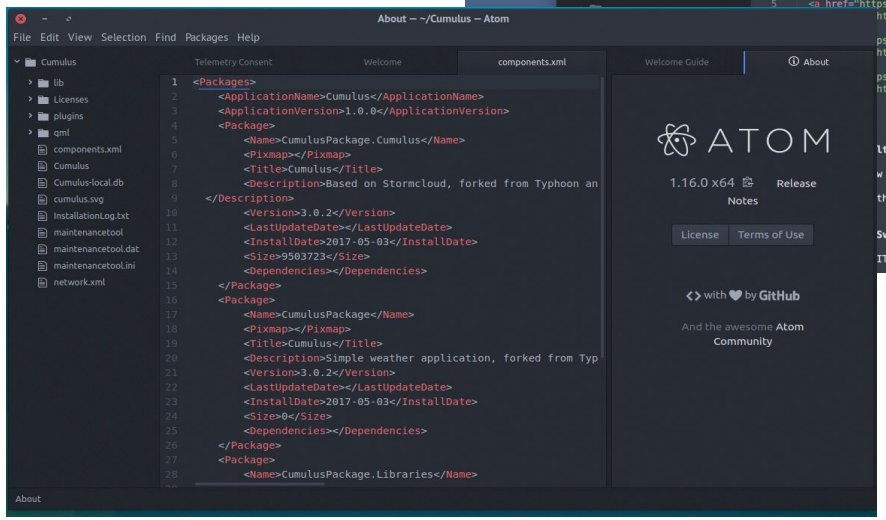
Code::Blocks

The open source, cross-platform IDE

Popular Text Editors

- **Visual Studio Code** (Extensible, cross-platform)
- **Sublime Text** (Fast, lightweight)
- **Atom** (Customizable, open-source)

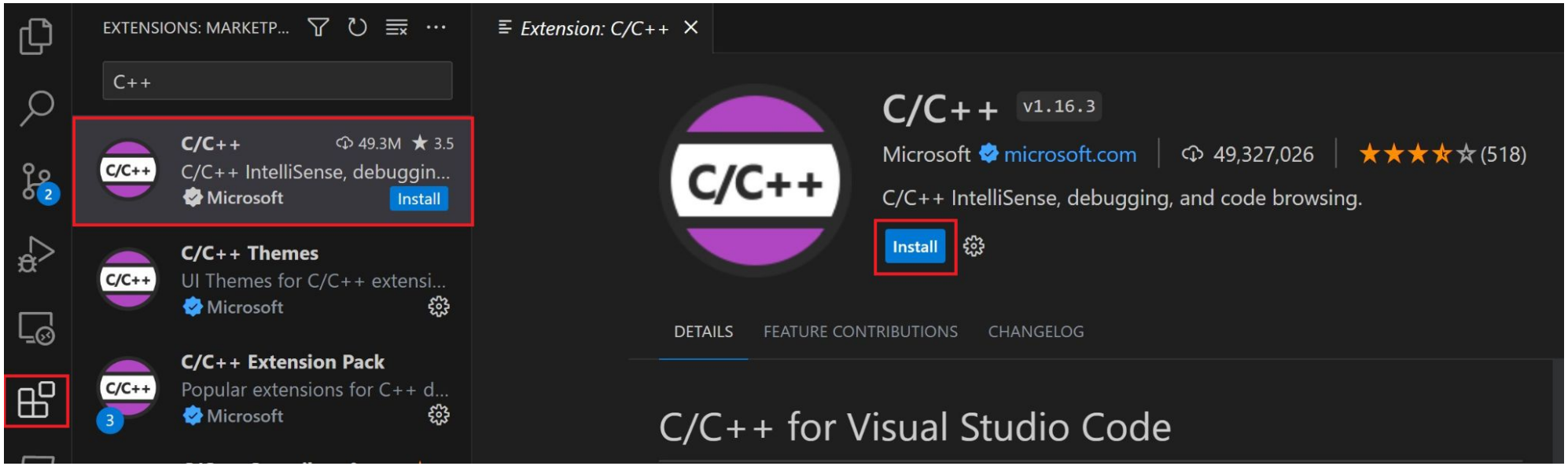




Using VS Code for C++

UNIVERSITAS GADJAH MADA

- Install C++ extension (C/C++ by Microsoft)
- Install compiler (GCC/MinGW)
- Configure tasks for build and run
- Use integrated terminal for execution



UNIVERSITAS
GADJAH MADA

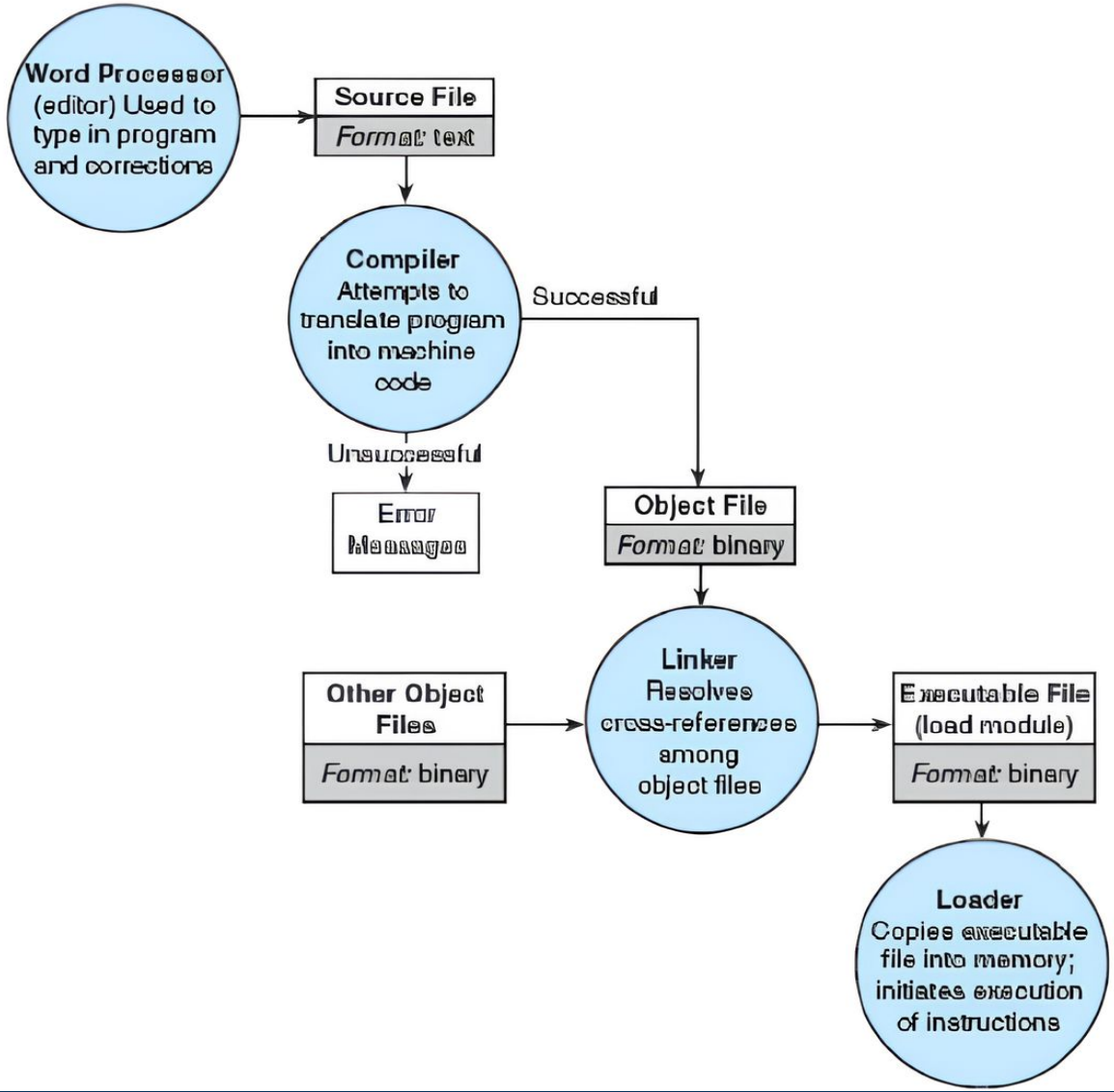
Basic Structure of C++ Program

ugm.ac.id

LOCALLY ROOTED, GLOBALLY RESPECTED

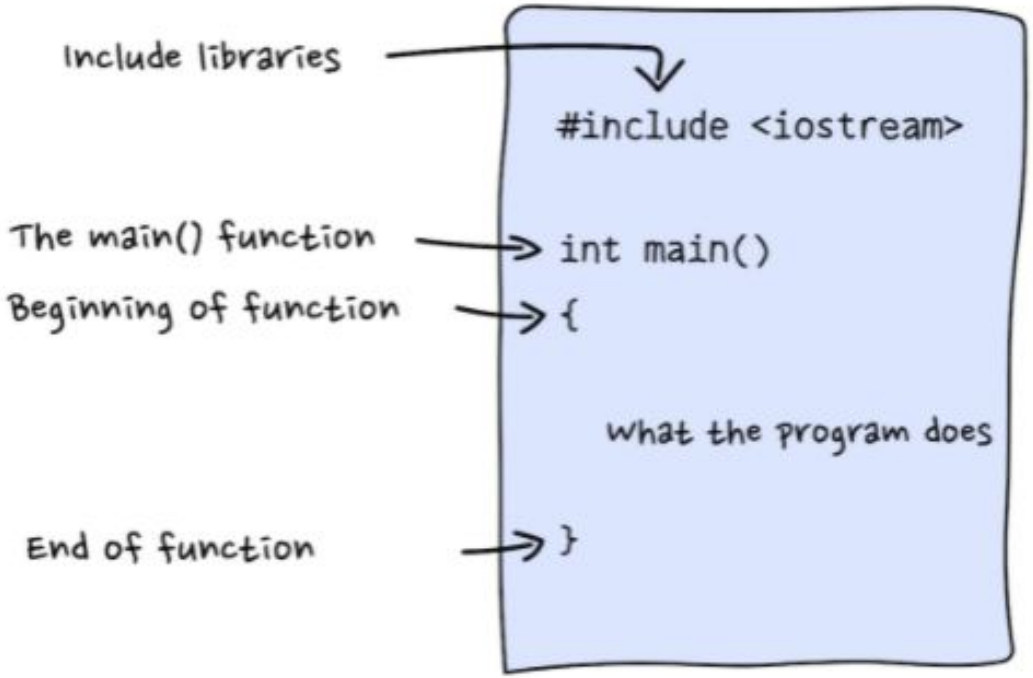
Flow of a C++ Program

Entering, Translating, and Running a High-Level Language Program



The General Structure of a C++ Program

UNIVERSITAS GADJAH MADA



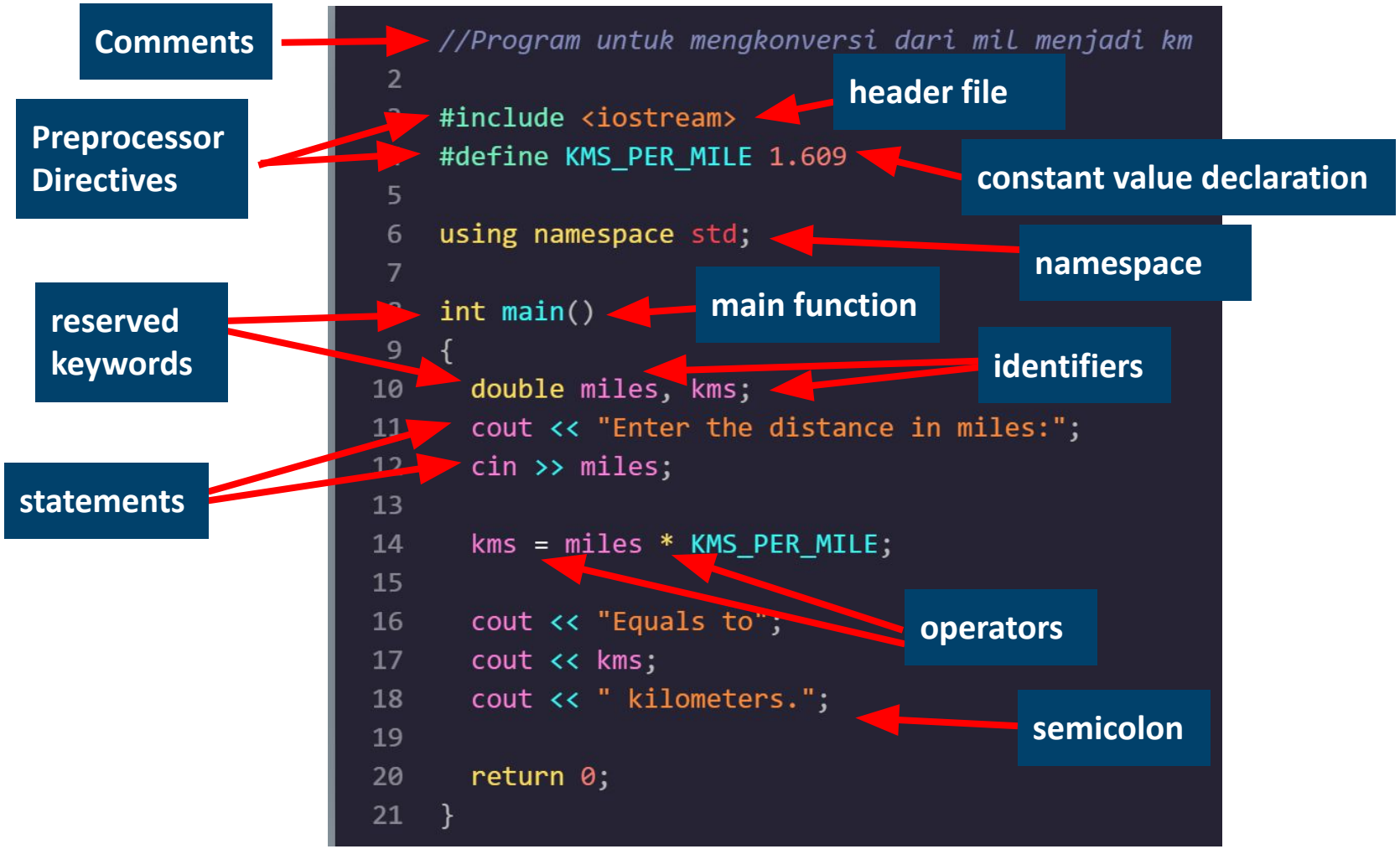

```cpp
#include <iostream>  // Preprocessor directive

using namespace std;  // Namespace declaration

int main() {  // Main function
    // Statements
    cout << "Hello, World!" << endl;  // Output statement
    return 0;  // Return statement
}
```

Elements of a Program in C++

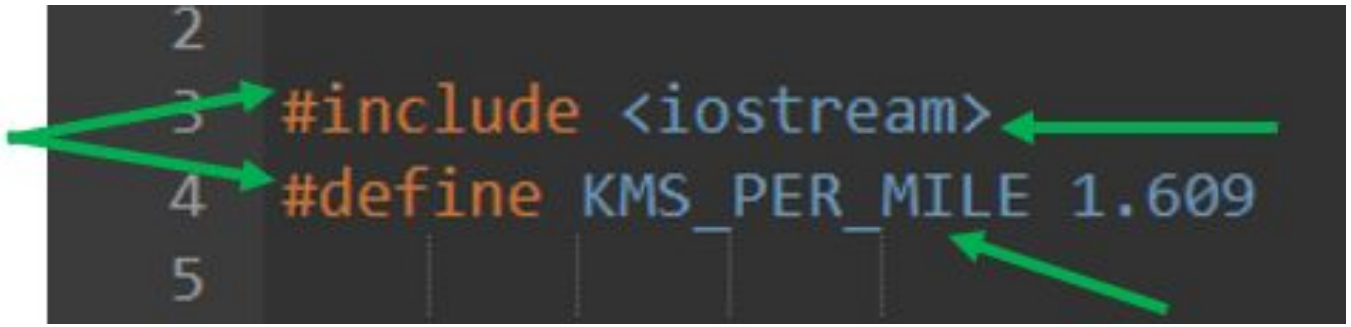
UNIVERSITAS GADJAH MADA



Preprocessor Directives

UNIVERSITAS GADJAH MADA

- A preprocessor directive is a command that gives instructions to the C++ preprocessor to modify the C++ program text before it is compiled.
- It begins with the # character.
- Common examples:
  - #include – include libraries
  - #define – define constants or macros



Namespace

UNIVERSITAS GADJAH MADA

- A namespace is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it.
- Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when your code base includes multiple libraries.
- All C++ standard library types and functions are declared in the std namespace.

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, World!";
    return 0;
}
```

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

int main() {
    cout << “Hello, World!” << endl;
    return 0;
}
```

without vs with using namespace std

Main Function

UNIVERSITAS GADJAH MADA

- Every C++ program always have a main function
- **‘int main()’** [line 8] marks the beginning of the main function where program execution starts. **‘int’** indicates that the main function will return an integer value (0) to the OS if the program executes successfully.
- No value inside ‘()’ can also be written as ‘(void)’ -> indicating that the main function does not receive data from the OS when starting execution.
- Body of the function [line 9-15] is enclosed within ‘**{ }**’ symbols. It contains declarations and executable statements.

```
8  int main( )
9  {
10     double miles, kms;
11     cout << "Enter the distance in miles:";
12     ...
13
14     return 0;
15 }
```

Reserved Keyword

- All programming languages have "reserved keywords/words".
- All reserved keywords appear in lowercase; they have special meanings in C++ and cannot be used for other purposes.
- A complete list of C++ reserved keywords can be seen in Table 1."

TABLE 1

and	double	not_eq	throw
and_eq	dynamic_cast	operator	true
asm	else	or	try
auto	enum	or_eq	typedef
bitand	explicit	private	typeid
bitor	extern	protected	typename
bool	false	public	union
break	float	register	unsigned
case	for	reinterpret-cast	using
catch	friend	return	virtual
char	goto	short	void
class	if	signed	volatile
compl	inline	sizeof	wchar_t
const	int	static	while
const-cast	long	static_cast	xor
continue	mutable	struct	xor_eq
default	namespace	switch	
delete	new	template	
do	not	this	

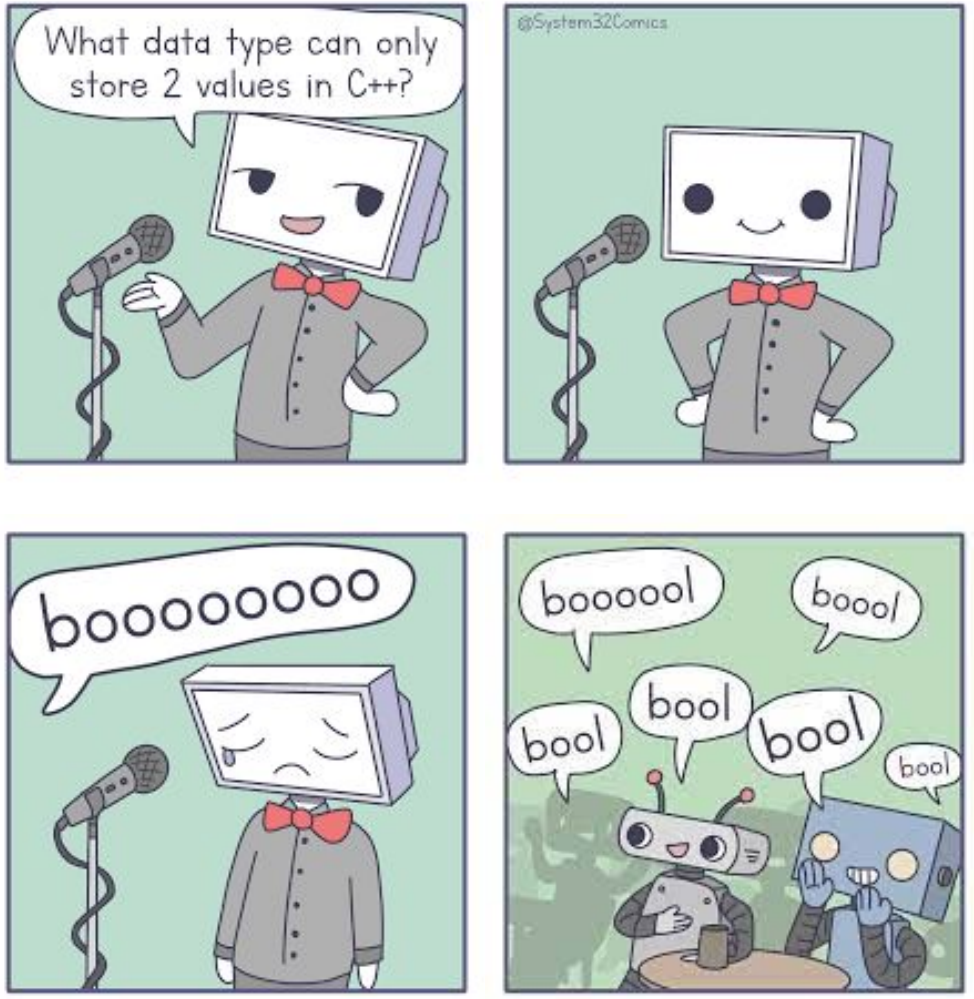
Data Types

UNIVERSITAS GADJAH MADA

Fundamental Types:

- int → integers
- float, double → real numbers
- char → single characters
- bool → true/false values

string → sequence of characters (text)



Why important?

- Determines **kind of data** stored.
- Defines **memory size** and **operations allowed**.

Identifiers (1)

UNIVERSITAS GADJAH MADA

- Identifiers are names used to represent variables, functions, constants, etc
- Two types of identifiers:
  - **Standard Identifiers** → predefined in C++ (e.g., `cin`, `cout`).
  - **User-defined Identifiers** → created by programmers (e.g., `miles`, `KMS_PER_MILE`).
- Rules for valid identifiers:
  - Can use letters, digits, underscore.
  - Cannot start with a digit.
  - Cannot use reserved keywords.

Identifiers (2)

UNIVERSITAS GADJAH MADA

- Invalid identifiers

TABLE 2.2 Invalid Identifiers

Invalid Identifier	Reason Invalid
`1Letter`	begins with a letter
`double`	reserved word
`int`	reserved word
`TWO*FOUR`	character * not allowed
`joe's`	character ' not allowed

- Reserved words and Identifiers

TABLE 2.3 Reserved Words and Identifiers in Fig. 2.1

Reserved Words	Standard Identifiers	User-Defined Identifiers
`int, void,`	`printf, scanf`	`KMS_PER_MILE, main,`
`double,`		`miles, kms`
`return`		

Operators

UNIVERSITAS GADJAH MADA

An **operator** in C++ is a special symbol that tells the compiler to perform specific mathematical, relational, or logical operations on one or more operands.

- **Arithmetic Operators** → **+, -, *, /, %**
- **Relational Operators** → **==, !=, <, >, <=, >=**
- **Logical Operators** → **&&, ||, !**
- **Assignment Operators** → **=, +=, -=, *=, /=**
- **Increment/Decrement** → **++, --**

```
int a = 5, b = 2;
cout << "a + b = " << a + b << endl;     // 7
cout << "a > b = " << (a > b) << endl;   // 1 (true)
cout << "a && b = " << (a && b) << endl; // 1 (true)
```

Input & Output (I/O)

UNIVERSITAS GADJAH MADA

- **Standard Library (iostream)**
  - cin → input from keyboard
  - cout → output to screen
- Example

```
int age;
cout << "Enter your age: ";
cin >> age;
cout << "You are " << age << " years old.";
```

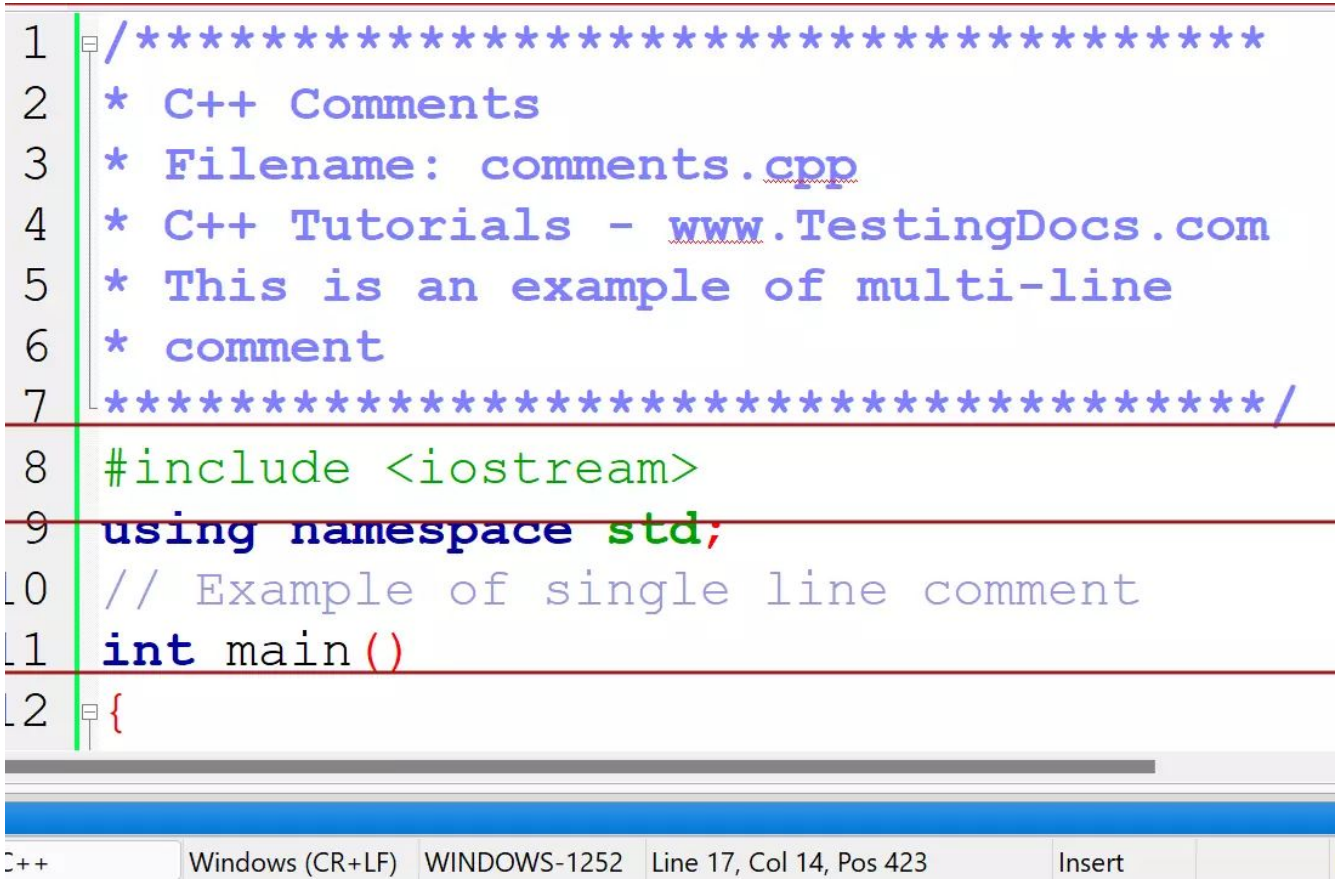
- Sample Output

```
Enter your age: 20
You are 20 years old.
```

Comments

UNIVERSITAS GADJAH MADA

- The text to the right of the // symbol is comments. Comments provide additional information to make it easier for us to understand the program, but comments are ignored by the compiler.
- Single line comments begin with //, multiple line comments enclosed between /* and */



Common Errors (Syntax & Type Errors)

UNIVERSITAS GADJAH MADA

- **Syntax Errors**
  - Occur when code violates the C++ grammar rules.
  - Example: Missing semicolon ;

```
int x = 10
cout << x;    // Missing semicolon
```

- **Type Errors (Type Mismatch)**

  Occur when assigning incompatible data types.

```
int x = "Hello";    // Error: incompatible types
```

Common Errors (Undeclared Variables)

UNIVERSITAS GADJAH MADA

- Happen when using variables before declaring them.
- The compiler cannot recognize the identifier.

```
cout << x;   // Error: ‘x’ was not declared in this scope
```

Common Errors (Return Statement & Case Sensitivity)

UNIVERSITAS GADJAH MADA

- **Missing return in main()**
  - Function int main() must return an integer (commonly return 0;).

```
int main() {
    cout << "Hello";
    // Missing return 0;
}
```

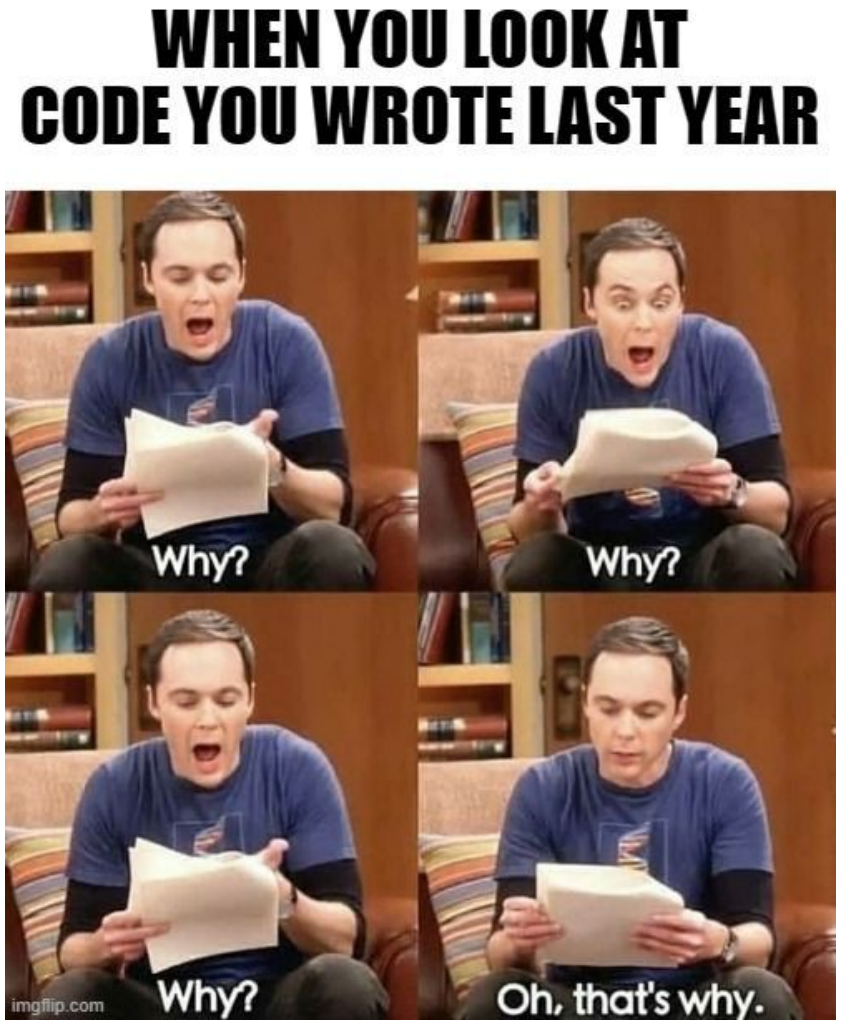
- **Case Sensitivity**
  - C++ treats uppercase and lowercase letters as different identifiers.

```
int main() {
    int Miles = 5;
    miles = 10;    // Error: ‘miles’ was not declared
}
```

Program Writing (1)

UNIVERSITAS GADJAH MADA

- A program that "looks good" is easier to read and understand.
- Most programs will be checked or studied by others.
- In industry, programmers spend more time on program maintenance (i.e., updating and modifying the program) than they do on the original design or coding.
- A well-organized program whose meaning is clear makes everyone's job simpler.



Program Writing (2)

UNIVERSITAS GADJAH MADA

- Meaningful names for *identifiers* are essential
  - Example: **salary** is better than **s** or **sal**.
- If the identifier consists of two or more words, placing an underscore character (_) between the words will improve the readability of the name.
  - Example: **dollars_per_hour** is better than **dollarsperhour**.
- Choose *identifiers* that are long enough to convey your meaning, but avoid names that are too long (as they increase the chance of typing errors)
  - Example: **lbs_per_sq_in** is better than **pounds_per_square_inch.**
- Avoid two names that differ only in the use of uppercase and lowercase letters.
  - Example: **LARGE** and **large**
- Do not use two names that are only slightly different, for example only distinguished by the presence/absence of an underscore.
  - Example: **xcoord** and **x_coord**.

UNIVERSITAS
GADJAH MADA

Any Question?

UNIVERSITAS
GADJAH MADA

Sneak Peak Demo Week 3

ugm.ac.id

LOCALLY ROOTED, GLOBALLY RESPECTED

What You'll Install

UNIVERSITAS GADJAH MADA

- **VS Code** (editor)
- **C++ compiler** (g++/clang or MSVC)
- **Debugger** (gdb or lldb)
- **VS Code extensions:**
  - *C/C++* (Microsoft) — IntelliSense, code navigation
  - *CodeLLDB* (for macOS lldb debugging)
  - *(Optional)* CMake Tools, C/C++ Extension Pack

You can follow the instructions from **ELOK**


KECEWA
BERAT


To Be Continued

UNIVERSITAS
GADJAH MADA

Thank You!
See you, next week, stay safe!