

Lecture 1:

Welcome to C++!

C++ Programming, Fall 2026

Congqing He (何从庆)

Today's Agenda

- Introduction
- Course Logistics
- The pitch: why C++ and what we will build
- First program: source code → compile → run

Introductions



Congqing He
何从庆

Lecturer, Huzhou Normal
University

Research in NLP, LLM, IR,
and AI for Law

♥ Huzhou, China

✉ hecongqing@hotmail.com

🐙 Github

🎓 Google Scholar

🆔 ORCID

I am currently a Lecturer at the School of Information Engineering, Huzhou Normal University, and a Part-time Postdoctoral Fellow at The Hong Kong Polytechnic University. If you are interested in academic collaboration, please feel free to contact me at hecongqing@hotmail.com.

I received my Ph.D. from the School of Computer Science, Universiti Sains Malaysia. Before that, I earned my M.S. from Hunan University and my B.S. from Hefei Normal University.

My research interests include deep learning, natural language processing, information retrieval, and AI for Law. I have published 10+ papers in SCI journals and top international AI conferences such as *Computer Science Review*, *ESWA*, *TIM*, *ACL*, *EMNLP*, and *ECAI*. More details can be found on my [Google Scholar](#).

📖 Educations

- **Mar. 2021 - Oct. 2025**, Ph.D. in School of Computer Science, Universiti Sains Malaysia.
- **Sep. 2016 - Jun. 2019**, M.S. in College of Computer Science and Electronic Engineering, Hunan University.
- **Sep. 2012 - Jun. 2016**, B.S. in School of Computer Science and Technology, Hefei Normal University.

👛 Experience

- **Dec. 2025 - present**, Lecturer, Huzhou Normal University, Huzhou, China.
- **Oct. 2025 - present**, Part-time Postdoctoral Fellow, The Hong Kong Polytechnic University, Hong Kong, China.
- **Jul. 2025 - Oct. 2025**, Research Assistant, The Hong Kong Polytechnic University, Hong Kong, China.
- **Jun. 2019 - Oct. 2020**, Algorithm Engineer, JD, Beijing, China.

🔥 News

- **Apr 2026**. Our paper, *Tree-CoT-RT: An Explainable Multi-Path Tree-Guided Chain-of-Thought and Reinforcement Learning Framework for Aspect Sentiment Quad Prediction*, was accepted by *Findings of ACL 2026*.
- **Jan 2026**. Our survey paper, *A Survey of Large Language Models for Legal Applications: Progress, Prospects and Challenges*, was published in *Computer Science Review*.
- **Dec 2025**. I joined Huzhou Normal University as a Lecturer in the School of Information Engineering.
- **Oct 2025**. I started serving as a Part-time Postdoctoral Fellow at The Hong Kong Polytechnic University.

<https://hecongqing.github.io>

ABOUT ME

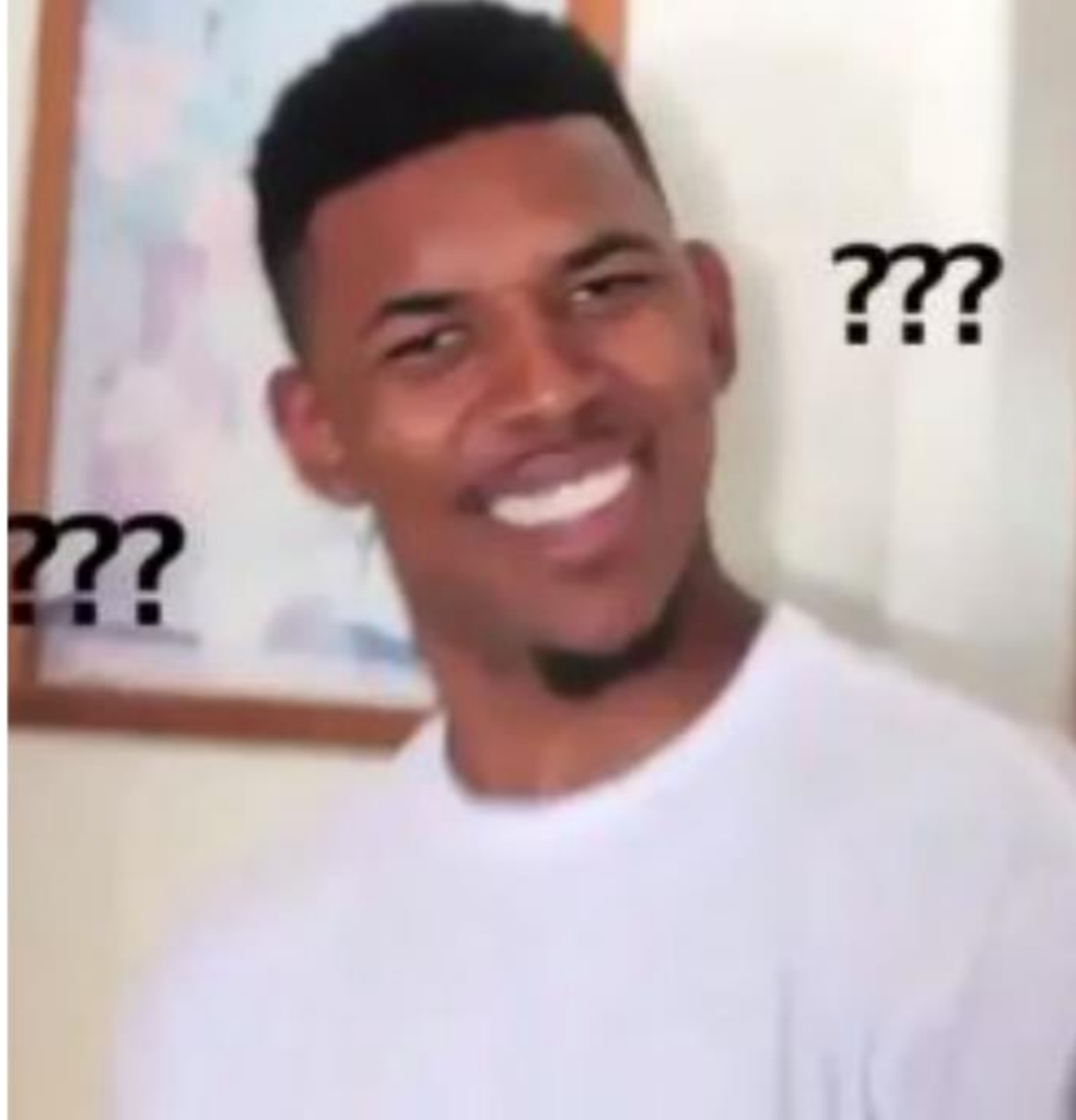
Name	Congqing He (何从庆)
Role	Instructor · C++ Programming
Email	hecongqing@huznu.edu.cn

Ask early—bring your code, output, and first error message.

Course Logistics

Asking Questions

- We welcome questions!
- Feel free to raise your hand at any time with a question
- We'll also pause periodically to solicit questions and check understanding



Learn C++ by Building and Testing

- C++ · 16 weeks · international undergraduate course
- Start with small programs, then build toward classes and projects.
- Predict → run → explain → modify → test.
- Keep lectures and practical tasks connected.
- Explain and change your own code—not just make it run.

Run it · explain it · change it · test it

A C++ Program

```
#include <iostream>

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

OUTPUT

Hello, world!

One source file → one visible result

Start with the Course Schedule

Open

<https://hecongqing.github.io/cpp-course/>

Check each week

[topic](#) · [class mode](#) · [materials](#) · [assignment](#)

Use the links

[Slides](#) · [Code](#) · [Assignment](#)

Also find

[Grading](#) · [Resources](#) · [EN](#) / [中文](#)

Bookmark this page before we begin.

START HERE

Introduction to C++

-  C++ is a powerful and widely used programming language that combines high-level abstraction with precise control over data and memory. This course introduces the core ideas of C++ through clear explanations, worked examples, and regular programming practice.
-  We begin with variables, expressions, input/output, selection, repetition, and functions, then move to strings, vectors, pointers, classes, streams, and inheritance. Students learn to read, write, test, and debug C++ programs throughout the course.
-  Lectures and practical tasks are built around small, meaningful programs, with an emphasis on program design, problem decomposition, code organization, and debugging. The course concludes with a complete, well-structured project.

Congqing He (何从庆)

 hecongqing@huznu.edu.cn

 Tuesday · Building 33, Room 33-B606

16-WEEK PLAN

Course Schedule

Week	Mode	Topic	Materials	Assignments
1	Mixed	Introduction	 Slides  Code	 Assignment
2	Theory	Types	 Slides	 Assignment



Course Assessment & Grading

Your final grade is calculated from two main components:

- 40%:** Continuous assessment
- 60%:** Final open-book written exam

Continuous Assessment (40%)

- **Attendance:** Regular check-ins during every session.
- **In-class engagement:** Participate in code walkthroughs, questions, and short practice tasks.
- **Assignments and labs:** Submit runnable programs with normal, boundary, and error tests.
- **Lab reports:** Six reports, one per lab session. Include code, test results, and brief explanations.
- **Submission:** Send PDFs to the class representative, who will collect and forward them to the instructor.

Note: Continuous assessment reflects your work throughout the semester. Keep your programs runnable, record your test results, and be ready to explain your code and design choices.

Final Open-Book Written Exam (60%)

Objective: Apply C++ fundamentals and object-oriented design.

Exam focus:

- **Code reading:** Trace variables, functions, classes, and program output.
- **Error analysis:** Identify compile-time, run-time, and logic errors.
- **OOP design:** Complete a short class, inheritance, or polymorphism task.

Note: Exam details will be shared mid-semester.

The Pitch

Why C++?

“The invisible foundation of everything”

Valorant



[source]

CS:GO / CS2



[source]

...and many more!



High Frequency Trading

Optiver 



CITADEL | Securities



[source]

Self Driving



TESLA



WAYMO

[source]

GPU Programming



NVIDIA®

CUDA®

And much, much more!

- Databases (MySQL, MongoDB)
- Web Browsers (Chrome, Safari, Edge)
- Virtual Reality (Quest)
- Low level ML (PyTorch, TensorFlow)
- Compilers and virtual machines (JVM, LLVM, GCC)
- Operating Systems (Windows, macOS, Linux)

“The invisible foundation of everything”

C++ is great for...

- Handling lots of data
- And handling it very efficiently
- And doing it in an elegant, readable way

C++ in Industry

amazon.com[®]

Meta



Google

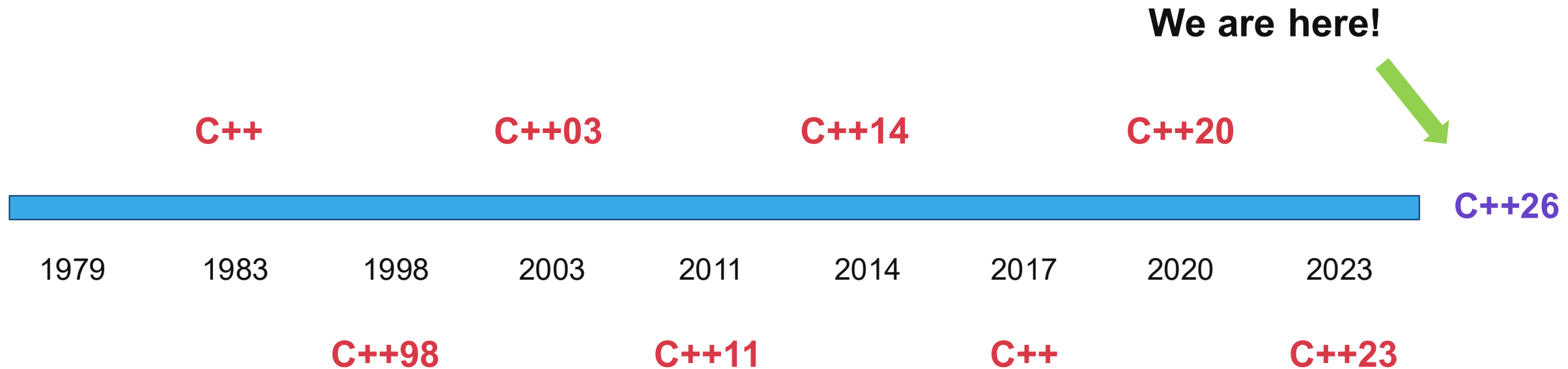
C++ was first released in 1985, still #3!

Mar 2026	Mar 2025	Change	Programming Language		Ratings	Change
1	1			Python	21.25%	-2.59%
2	4	▲		C	11.55%	+2.02%
3	2	▼		C++	8.18%	-2.90%
4	3	▼		Java	7.99%	-2.37%
5	5			C#	6.36%	+1.49%
6	6			JavaScript	3.45%	-0.01%
7	9	▲		Visual Basic	2.50%	-0.02%
8	8			SQL	2.00%	-0.57%
9	16	▲▲		R	1.88%	+0.94%
10	10			Delphi/Object Pascal	1.80%	-0.36%

[TIOBE Index, March 2026]

The C++ Community

- C++ has a MASSIVE user base
- The C++ Standard continues to be revised every three years
- A standards committee decides new changes to the language



What is C++?

A C++ Program

```
#include <iostream>

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

OUTPUT

Hello, world!

One source file → one visible result

Read the program in four moves

```
#include <iostream>

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

- 1 include**
bring in the output tool
- 2 main**
the program starts here
- 3 {} and ;**
group steps and end statements
- 4 cout**
send text to the console

std::cout sends a sequence of values to the console

```
std::cout << "Name: " << "Ada" << "\n";  
std::cout << "Year: " << 1 << "\n";
```

OUTPUT

```
Name: Ada  
Year: 1
```

Read the line as a left-to-right stream of values.

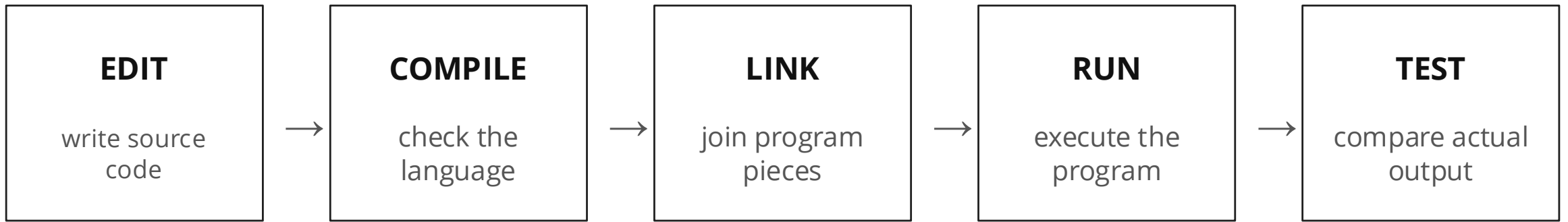
Text is data, and `\n` controls the next line

```
std::cout << "Hello";  
std::cout << "\n";  
std::cout << "Welcome";
```

```
Hello  
Welcome
```

A string is text between quotes.
A newline moves the cursor down.

The computer follows a build-and-test path



Edit → Compile → Link → Run → Test

OnlineGDB: Code in Your Browser

OnlineGDB

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>
```

```
int main()
```

```
{
```

```
    std::cout<<"Hello World";
```

```
    return 0;
```

```
}
```

Use OnlineGDB in Four Steps

- 1 · Open** https://www.onlinegdb.com/online_c++_compiler
- 2 · Choose** Language: C++
- 3 · Write** Edit main.cpp · use stdin when the program needs input
- 4 · Run & check** Run/F9 → output · Debug/F8 → step through

Read the first error · fix one thing · run again

Your First OnlineGDB Program

```
#include <iostream>

int main() {
    std::cout << "Hello, my name is
    _____.\\n";
    std::cout << "I study _____.\\n";
    std::cout << "I want to learn _____.\\n";
    return 0;
}
```

Done when...

- your own name appears
- each line is readable
- the program compiles
- the output matches your plan

A Creeper from the game Minecraft is shown in a cave setting, surrounded by dirt and stone blocks. The Creeper is green with a pixelated texture and a black, jagged mouth. It is positioned in the center of the frame, facing forward. The background consists of brown and grey pixelated blocks, typical of a Minecraft cave environment.

Open OnlineGDB. Run It. Change It. Test It.

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

Three common types of errors

Syntax error

The code breaks a language rule.

Example: a missing semicolon.

Read the first compiler message.

Runtime error

A problem occurs while the program runs.

Example: an uncaught exception stops the program.

Check which operation failed.

Logic error

The program runs, but the result is wrong.

Example: a missing newline joins two messages.

Compare expected and actual output.

Make one small change, run again, and check the result.

A Creeper from the game Minecraft is shown from the chest up, standing in a dirt pit. The Creeper has its characteristic green, pixelated skin and a black, blocky face with a wide, toothy grin. It is positioned in the center of the frame, with its body partially obscured by a large black text box. The background consists of brown dirt blocks and a green grassy floor.

What questions do we have?

creeper_abt_to_explode_with_questions