哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

高级语言程序设计
High-level Language Programming

Lecture 1 Introduction

Yitian Shao (shaoyitian@hit.edu.cn)
School of Computer Science and Technology

Instructor Information

Prof. Dr. **Yitian Shao** (Chinese name: **邵奕天**)

- School of Computer Science and Technology
- Research interests: **haptic interfaces**, **robotic tactile sensing**, **wearable technologies**, and **extended reality**

Academic Background

- Junior professor (W1) of Electrical and Computer Engineering, TU Dresden, Germany (2022-2023)
- Postdoctoral Researcher, Max Planck Institute for Intelligent Systems, Stuttgart, Germany (2021-2022)
- Ph.D., Electrical and Computer Engineering, UC Santa Barbara, U.S. (2015-2020)

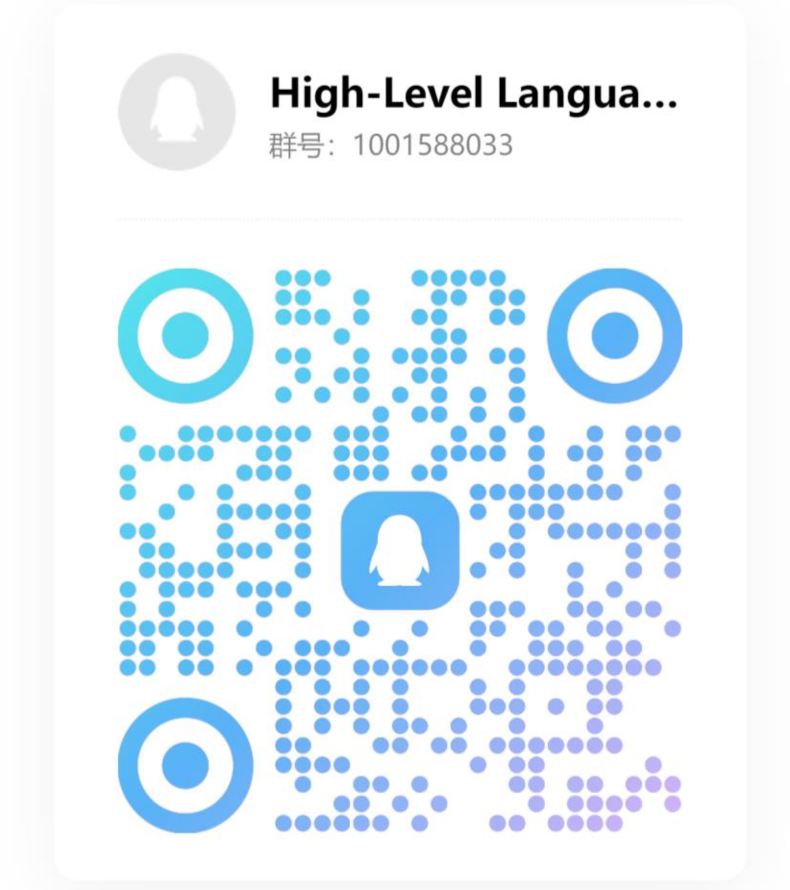
Email: shaoyitian@hit.edu.cn

Teaching Assistant: **Shih Ying-Lei**

Email： 23s151045@stu.hit.edu.cn

Let’s Get Connected!

- Course related discussion on QQ
  - Posting homework
  - Sharing course materials
  - Course related Q&A



High-Level Langua...

群号：1001588033

- **Step 1**: Download and install QQ on your mobile/PC: im.qq.com/index

iOS

Android

- **Step 2**: Inside your QQ, scan the QR code above
- Group number **1001588033**

Course Objectives

- Master the fundamental approaches to **solve problems with computers**
- Master the basic knowledge of **high-level programming language** (C++)
- Master the fundamental methods of program design and implementation
- Acquire the ability to solve practical problems using computers
- Acquire the basic skills for programming and debugging

Assessment & Grades

- 6 weeks of courses for the Computer Science (**CS**) part
  - Monday/Wednesday 4pm-5:45pm, T5-406
  - Lab (4xTuesday: **9.24, 10.22, 11.5, 11.12**) 6:45pm-8:30pm, T2-210

- Course assignments and exams
  - **Homework** (2% each) x 10 = **20% points**
  - **Laboratory Assignments** (10% each) x 4 = **40% points**
  - **Final Examination** x 1 **= 40% points**

- Homework & Lab submission and grading
  - Teaching Assistant: **Shih Ying-Lei**
  - Email：23s151045@stu.hit.edu.cn

Class Attendance

- Class attendance is required and will be checked randomly.

If you miss over 1/3 class (miss > 4 lectures), you will be disqualified to attend the Final Examination (Loss of 40% points)

- **Recommended**: Bring your laptop to class, for in-class exercises and try it yourself examples.

Expectations

- Attend lectures on time
- Complete assignments on your own and submit on time
- Engage in class discussion
- Communicate with instructor if need help

- All lectures are given in **English**
- All assignments and exams must be completed in **English**

Textbook

Paul Kelly, Xiaohong Su, ***Learning C++ through English and Chinese (2nd Edition)***, Publishing House of Electronics Industry, 2016

Other online references and materials

- Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, *C++ Primer (Fifth Edition)*, Addison-Wesley, 2013
- Mark Allen Weiss, *Data Structures and Algorithm Analysis in C++ (Fourth Edition)*, Pearson Education, 2014

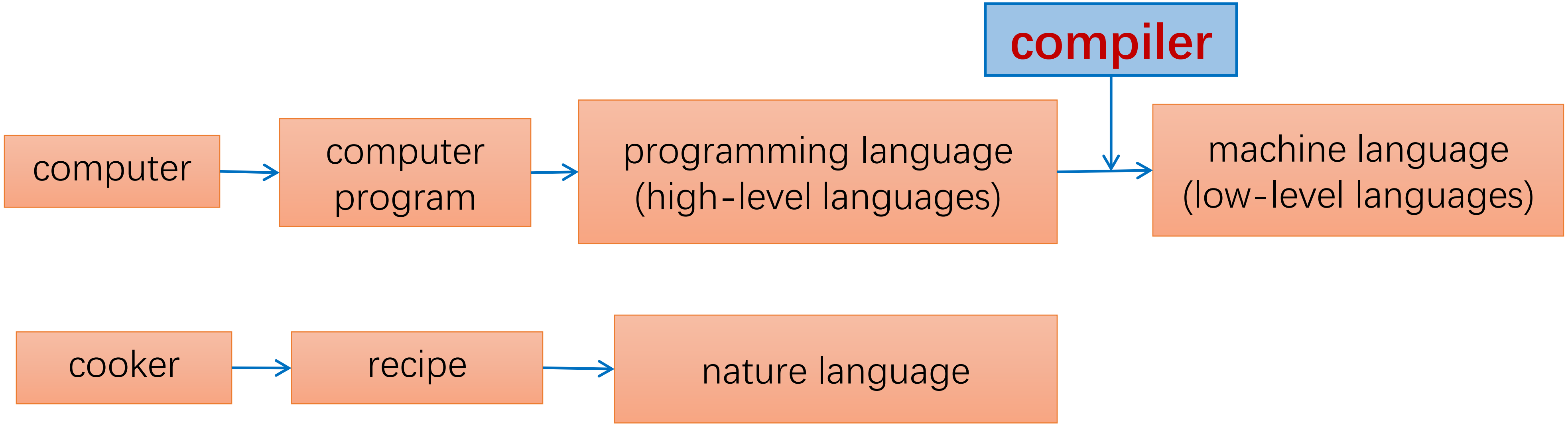
Introduction

Course Overview

- 1.1 What is a computer program?
- 1.2 How to develop a computer program?
- 1.3 The brief history of the C++ language
- 1.4 The common basic standards in the C++
- 1.5 How to use the online programming tool, e.g. "onlinegdb“
- 1.6 How to use the C++ programming tool
- 1.7 How to learn C++

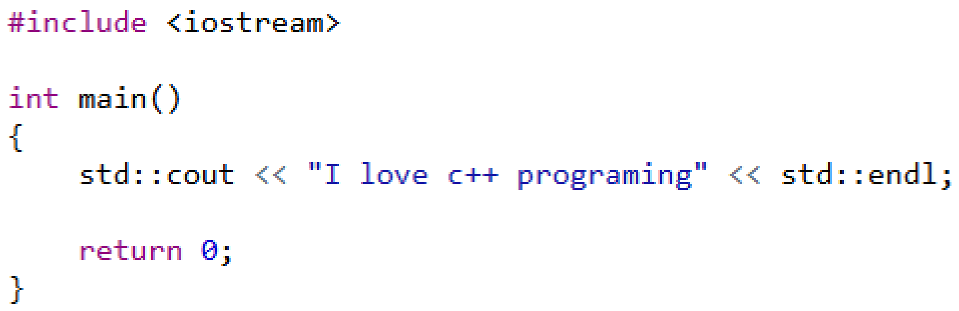
1.1 What is a computer program?

- **Computer program**： A computer performs all of these tasks by following a predefined set of instructions.


compiler
computer
computer program
programming language (high-level languages)
machine language (low-level languages)
cooker
recipe
nature language

1.1 What is a computer program?

- **Programming language**：



```
#include <iostream>

int main()
{
    std::cout << "I love c++ programing" << std::endl;

    return 0;
}
```

- **Natural language**：

Please print “I love c++ programing” onto the screen.

1.2 Developing a computer program

- Define and understand the problem to be solved----analysis phase

- How is it to be done? -----design phase

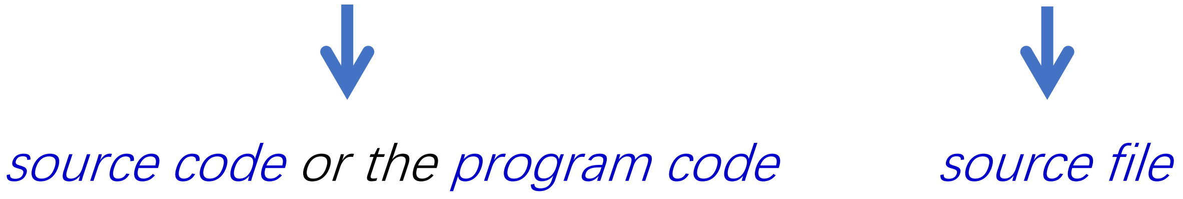
- Writing, compiling and testing C++ programs

1.2 Developing a computer program

Program development cycle

- Step 1: Design the program.
  - Each program has to be individually designed to implement the overall solution developed at the analysis and design phases.
  - Check its logic before starting to write the program

- Step 2: Write the program.
  - The C++ program instructions are typed into a file using a *text editor*



source code or the program code

source file

1.2 Developing a computer program

- Step 3: Compile the program.
  - Translate the C++ program instructions to machine instructions through a compiler--*object code, and stores the object code in an object file.*
  - Errors ---- *compile-time syntax error:* missing punctuation , misspellings,⋯
  - Warning message: not as serious as a syntax error, not prevent the program from being compiled, may be a problem and should be investigated

- Step 4: Link the program.
  - Combining the object file of the program with other object files from the C++ run-time library to form an *executable file*

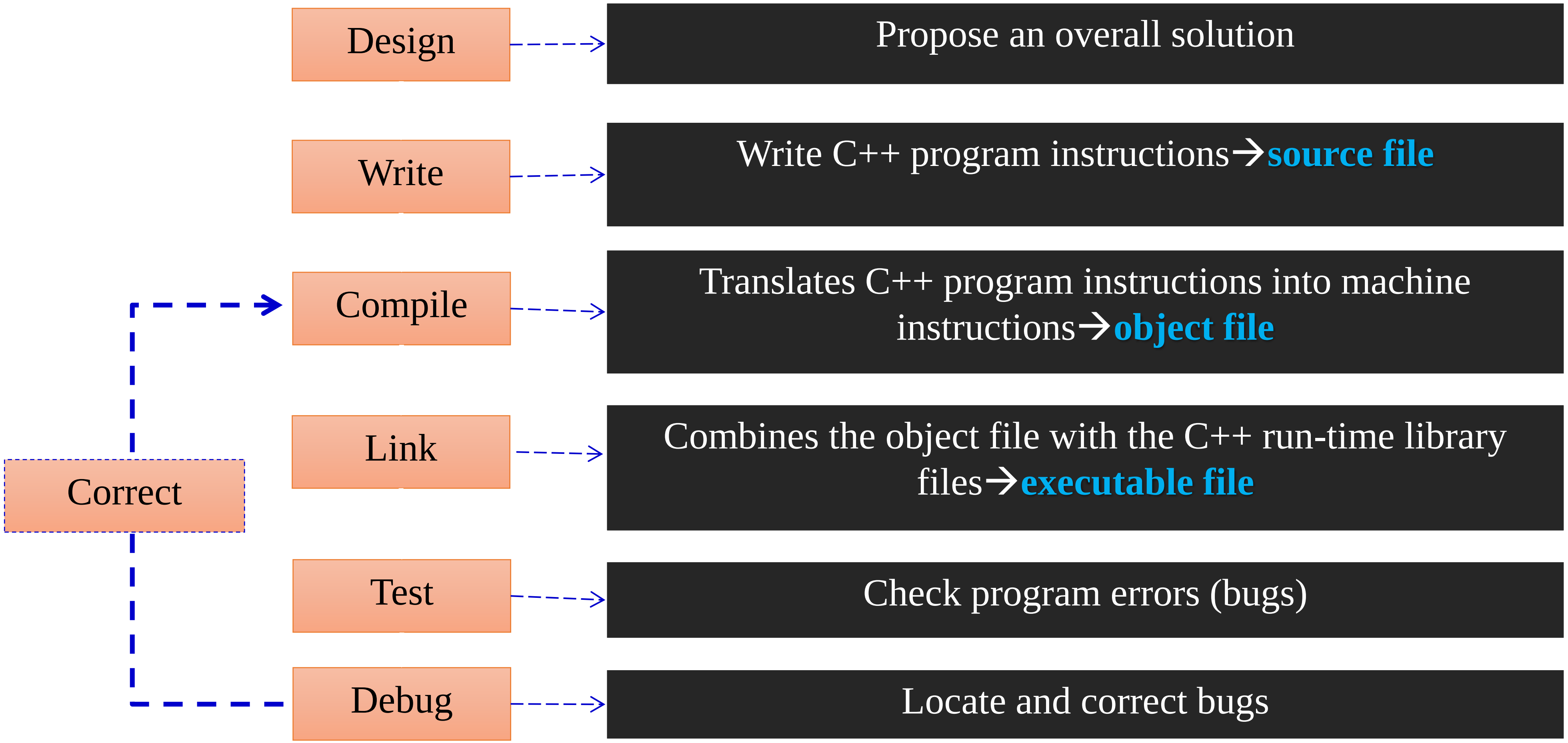
1.2 Developing a computer program

- Step 5: Test the program.
  - *run-time error*—causes the program to stop before it has completed its task
  - *logic errors(**bugs**)*---The program may complete its task but produce incorrect results

- Step 6: Debug the program.
  - The process of locating and correcting program errors is called *debugging*
  - Many compilers have tools that can be used to help locate bugs
  - Correcting bugs involves going back to step 2, try to catch errors as early as possible

1.2 Developing a computer program

Program development cycle



Design → Propose an overall solution

Write → Write C++ program instructions→**source file**

Compile → Translates C++ program instructions into machine instructions→**object file**

Link → Combines the object file with the C++ run-time library files→**executable file**

Test → Check program errors (bugs)

Debug → Locate and correct bugs

Correct

1.3 Brief history of C++

- 1972,   AT&T, Bell Lab. Dennis Ritche, C programming language
- 1980, Bell Lab. Bjarne Stroustrup, “C with Classes”
- 1983, C++ was developed by Bjarne Strostrup at AT&T Bell Laboratories
- 1985, the first commercial version of C++
- 1997, ANSI (American National Standards Institute) C++ (standard C++)

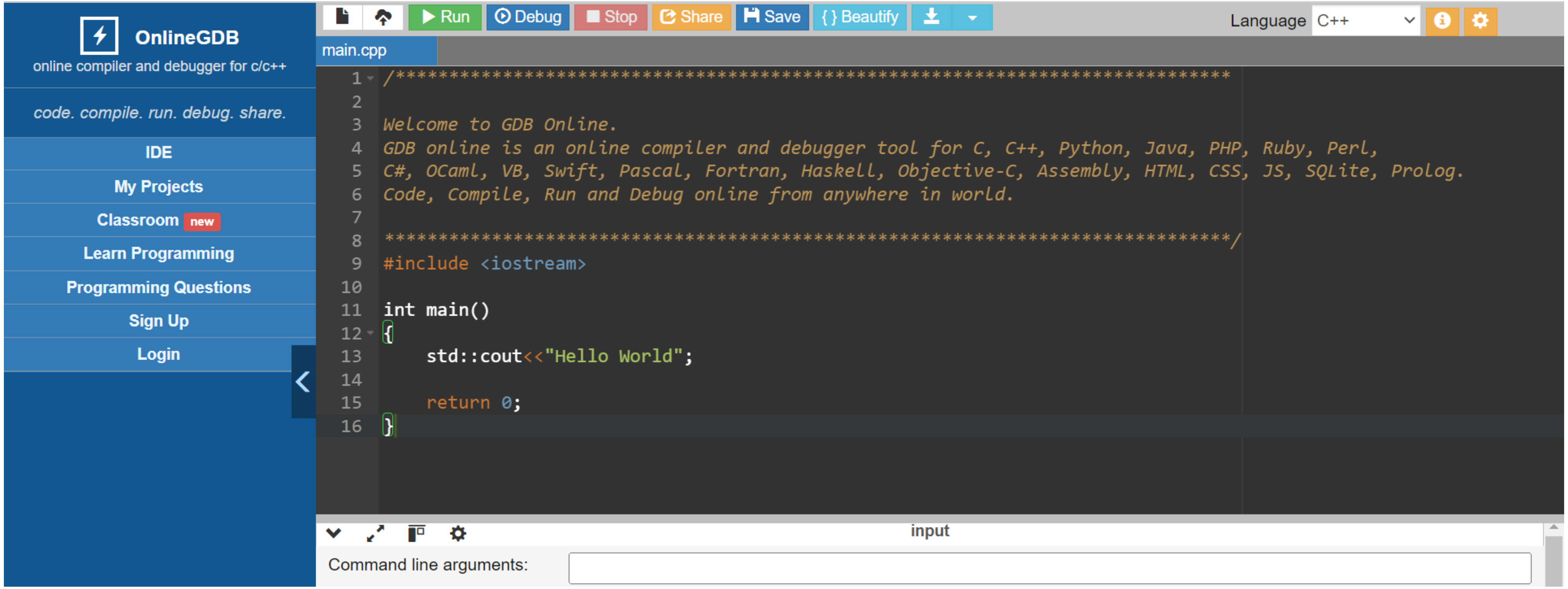
C++ is always evolving !!!

1.4 Common standards in the C++

- The example programs used in this book conform to the ANSI/ISO C++ standard.
    - Not all compilers conform to this standard, so some compilers may not correctly compile the example programs
    - Some of the example programs may have to be modified for use with these other compilers
    - See the website for more details

1.5 Online Programming

onlinegdb: https://www.onlinegdb.com/

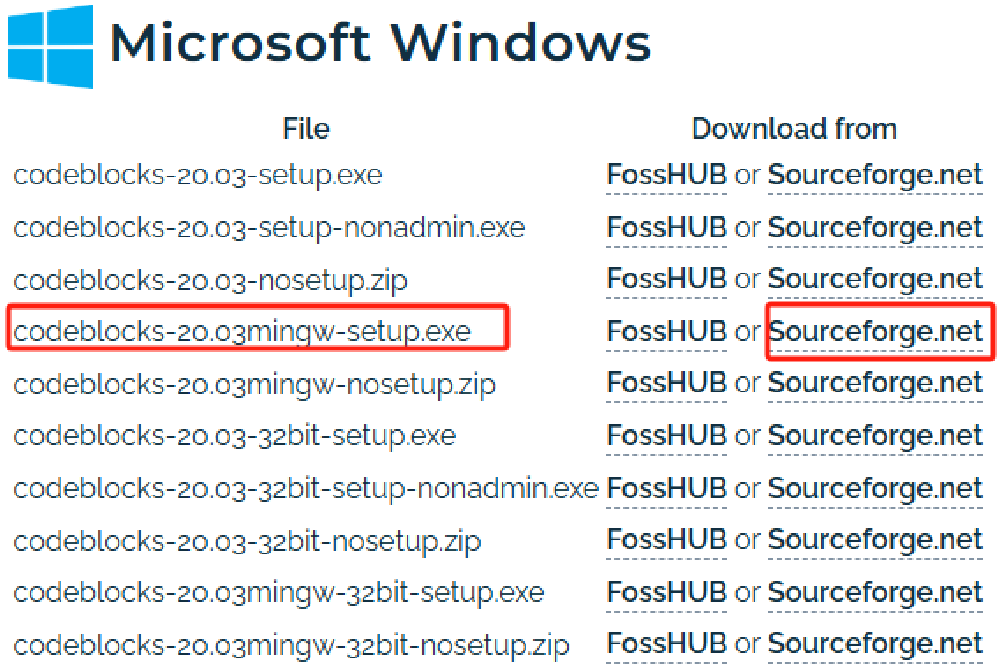


```
/******************************************************************************

Welcome to GDB Online.
GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,
C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.
Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************************/
#include <iostream>

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

    return 0;
}
```

1.6 Offline Programming

CodeBlocks

download CodeBlocks and install it



Microsoft Windows

File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

1.6 Offline Programming

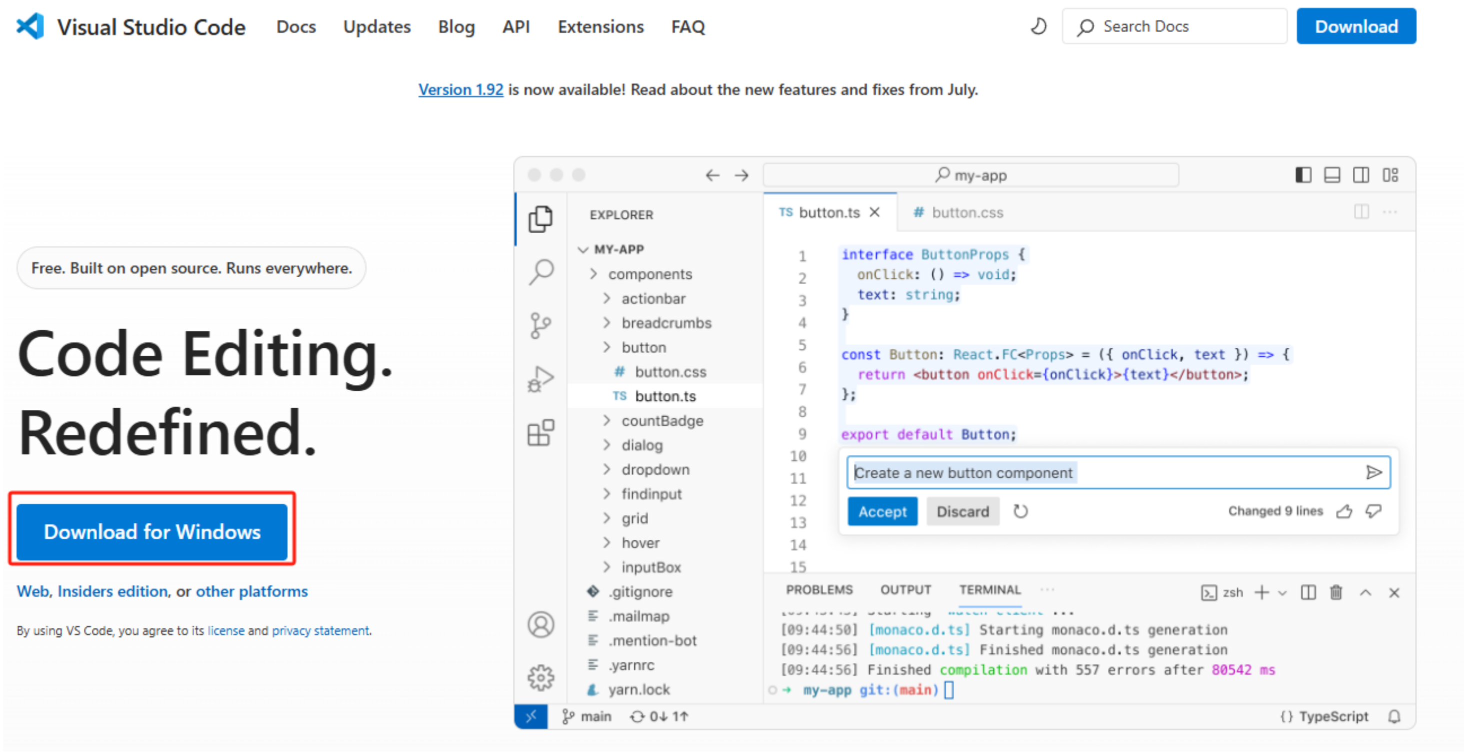
CodeBlocks

The installation package includes the compilation environment, and the installation will automatically find the environment path. Therefore, you can directly create new files to write and run C++ programmes.

1.6 Offline Programming

Visual Studio Code

- 1. download Visual Studio Code and install it



Version 1.92 is now available! Read about the new features and fixes from July.

Free. Built on open source. Runs everywhere.

Code Editing. Redefined.

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

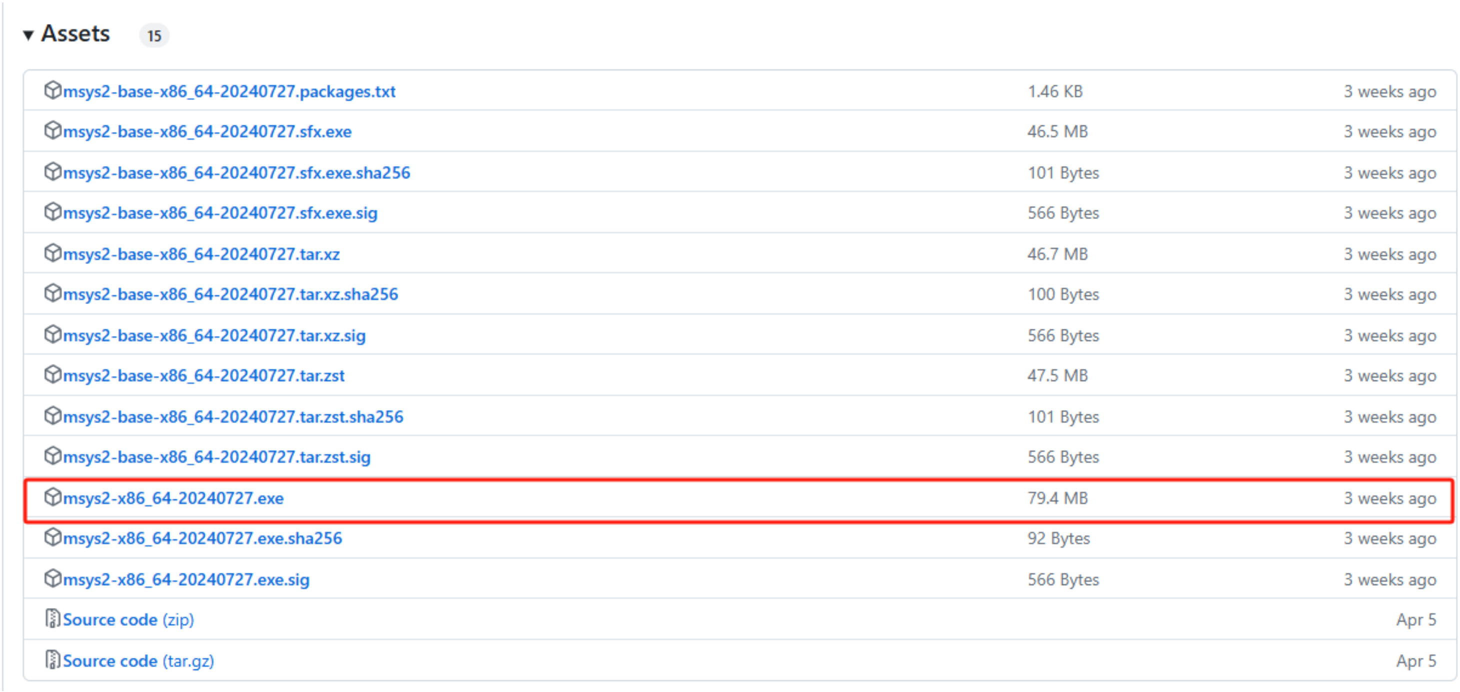
```
interface ButtonProps {
  onClick: () => void;
  text: string;
}

const Button: React.FC<Props> = ({ onClick, text }) => {
  return <button onClick={onClick}>{text}</button>;
};

export default Button;
```

Create a new button component

```
[09:44:50] [monaco.d.ts] Starting monaco.d.ts generation
[09:44:56] [monaco.d.ts] Finished monaco.d.ts generation
[09:44:56] Finished compilation with 557 errors after 80542 ms
○ → my-app git:(main)
```

1.6 Offline Programming

- 2. download MinGW-W64 and install it



▾ Assets 15

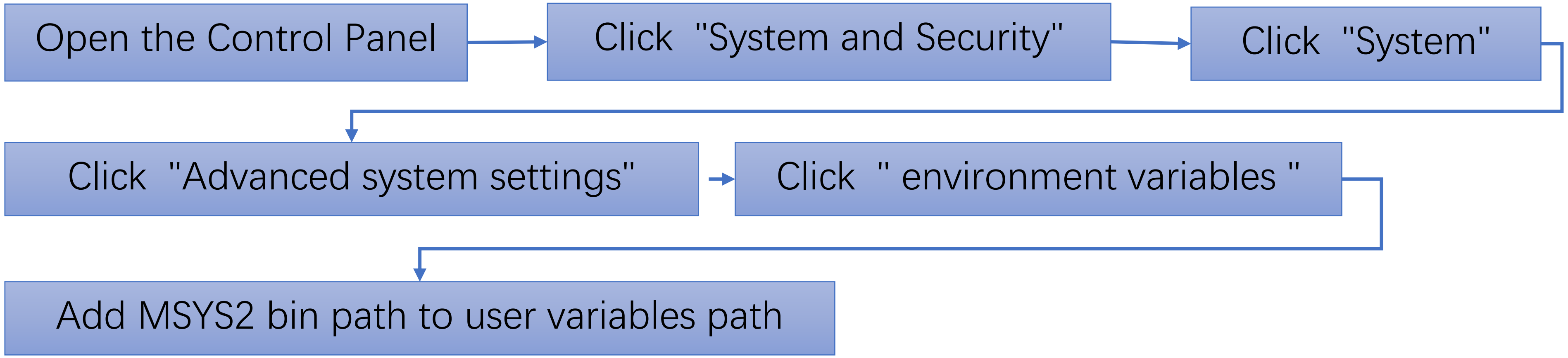
msys2-base-x86_64-20240727.packages.txt	1.46 KB	3 weeks ago
msys2-base-x86_64-20240727.sfx.exe	46.5 MB	3 weeks ago
msys2-base-x86_64-20240727.sfx.exe.sha256	101 Bytes	3 weeks ago
msys2-base-x86_64-20240727.sfx.exe.sig	566 Bytes	3 weeks ago
msys2-base-x86_64-20240727.tar.xz	46.7 MB	3 weeks ago
msys2-base-x86_64-20240727.tar.xz.sha256	100 Bytes	3 weeks ago
msys2-base-x86_64-20240727.tar.xz.sig	566 Bytes	3 weeks ago
msys2-base-x86_64-20240727.tar.zst	47.5 MB	3 weeks ago
msys2-base-x86_64-20240727.tar.zst.sha256	101 Bytes	3 weeks ago
msys2-base-x86_64-20240727.tar.zst.sig	566 Bytes	3 weeks ago
msys2-x86_64-20240727.exe	79.4 MB	3 weeks ago
msys2-x86_64-20240727.exe.sha256	92 Bytes	3 weeks ago
msys2-x86_64-20240727.exe.sig	566 Bytes	3 weeks ago
Source code (zip)		Apr 5
Source code (tar.gz)		Apr 5

1.6 Offline Programming

- 2. After finishing the installation, open MSYS2 and enter the following command to install the toolchain:

```
pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

- 3. Configure system environment variables



Open the Control Panel

Click "System and Security"

Click "System"

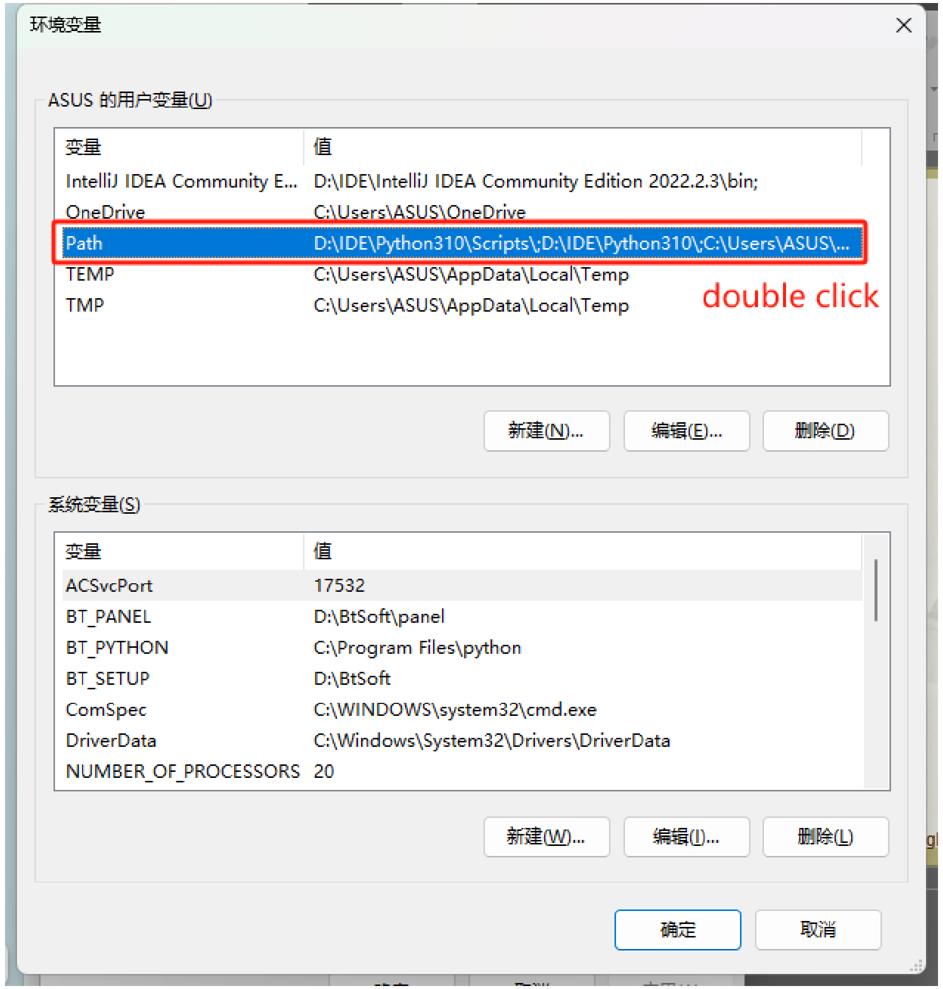
Click "Advanced system settings"

Click " environment variables "

Add MSYS2 bin path to user variables path

MSYS2 bin path : installation path of MSYS2 + ucrt64\bin

1.6 Offline Programming



环境变量

ASUS 的用户变量(U)

变量	值
IntelliJ IDEA Community E...	D:\IDE\IntelliJ IDEA Community Edition 2022.2.3\bin;
OneDrive	C:\Users\ASUS\OneDrive
Path	D:\IDE\Python310\Scripts\;D:\IDE\Python310\;C:\Users\ASUS\...
TEMP	C:\Users\ASUS\AppData\Local\Temp
TMP	C:\Users\ASUS\AppData\Local\Temp

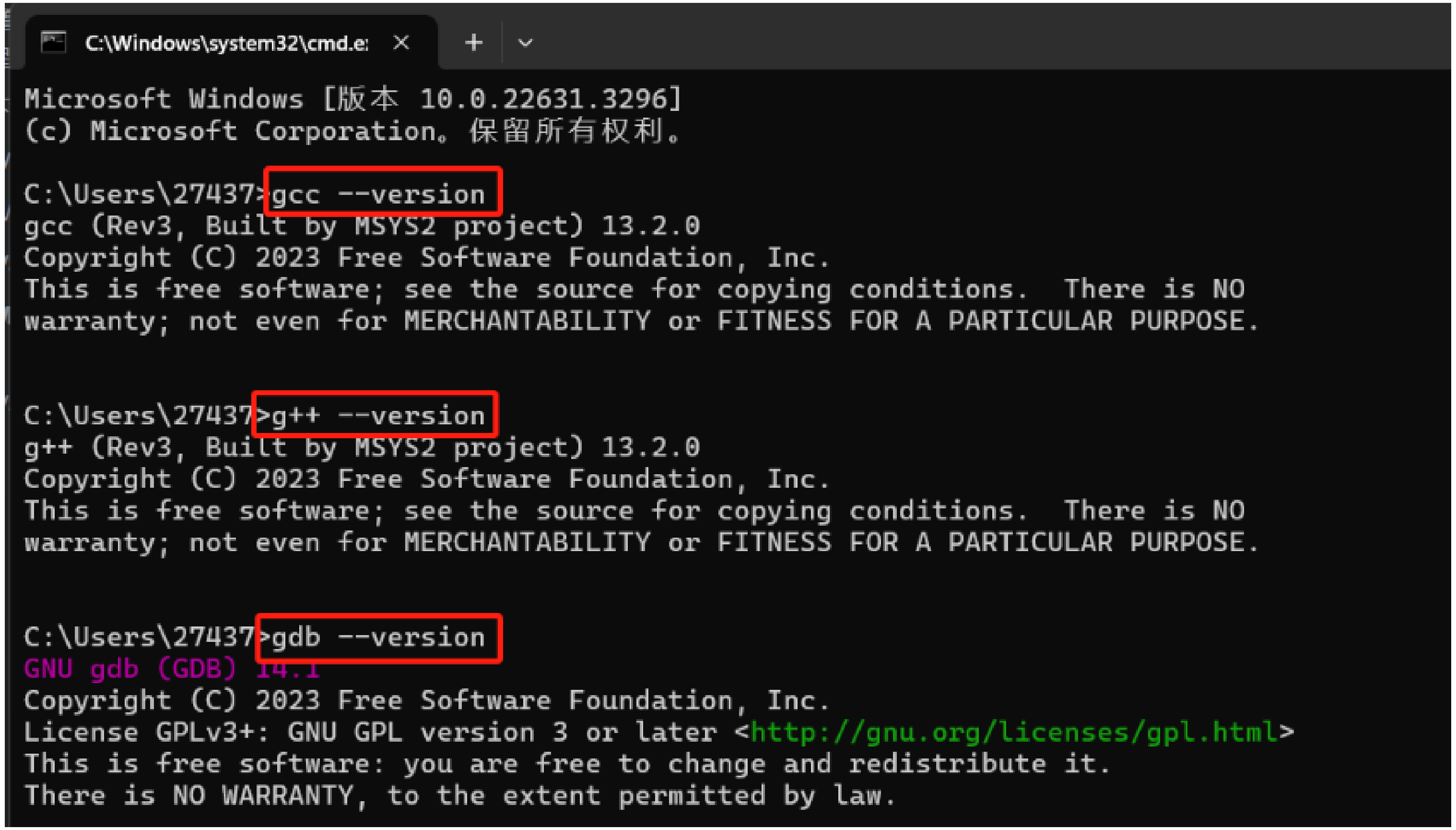
double click

系统变量(S)

变量	值
ACSvcPort	17532
BT_PANEL	D:\BtSoft\panel
BT_PYTHON	C:\Program Files\python
BT_SETUP	D:\BtSoft
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	20

1.6 Offline Programming

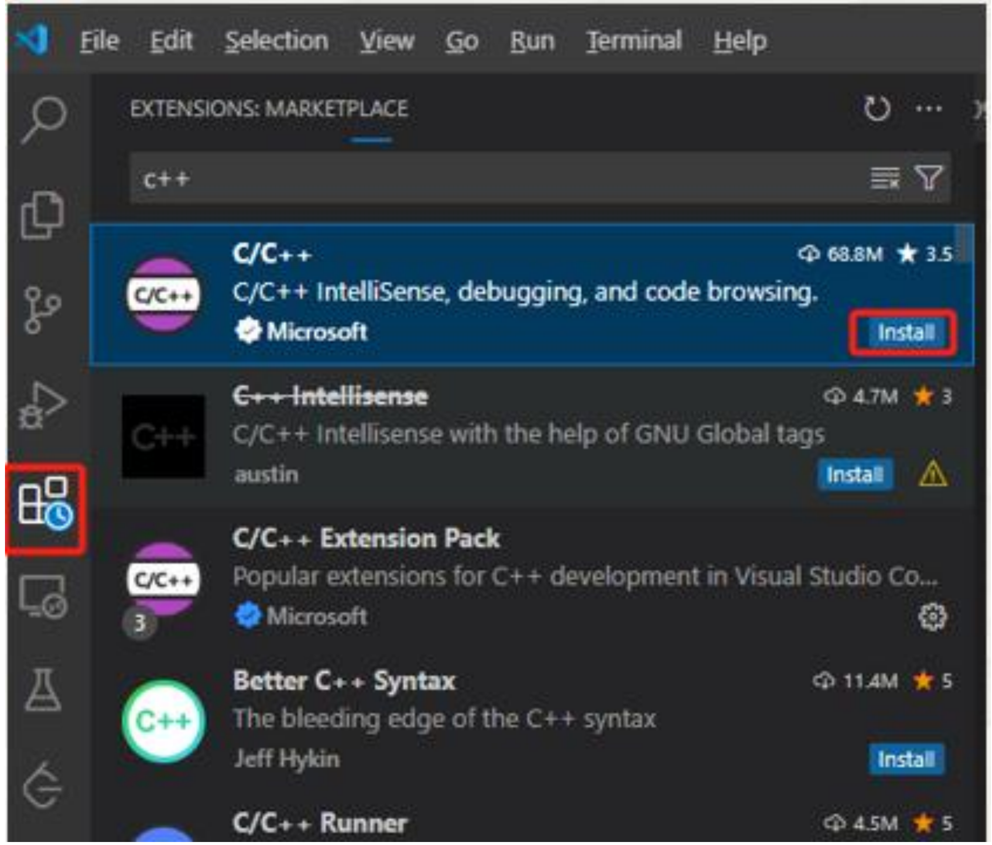
- 4. use the following commands to test the installation of environment:



```
Microsoft Windows [版本 10.0.22631.3296]
(c) Microsoft Corporation。保留所有权利。

C:\Users\27437>gcc --version
gcc (Rev3, Built by MSYS2 project) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\27437>g++ --version
g++ (Rev3, Built by MSYS2 project) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\27437>gdb --version
GNU gdb (GDB) 14.1
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

1.6 Offline Programming

- 5. open Visual Studio Code, search for "C++" in extensions and install it:



C/C++ 68.8M ★ 3.5
C/C++ IntelliSense, debugging, and code browsing.
Microsoft

~~C++ Intellisense~~ 4.7M ★ 3
C/C++ Intellisense with the help of GNU Global tags
austin

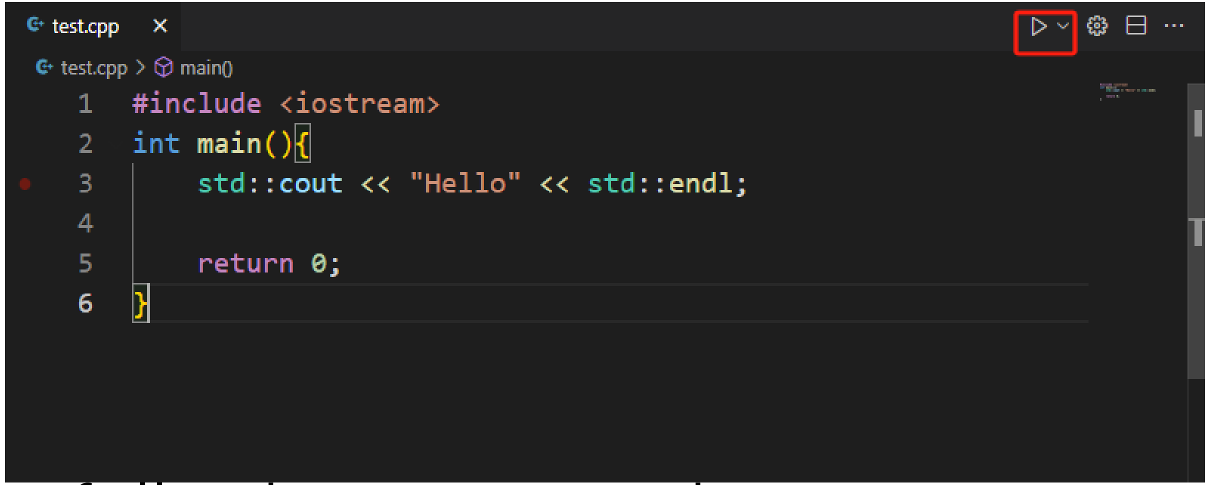
C/C++ Extension Pack
Popular extensions for C++ development in Visual Studio Co...
Microsoft

Better C++ Syntax 11.4M ★ 5
The bleeding edge of the C++ syntax
Jeff Hykin

C/C++ Runner 4.5M ★ 5

1.6 Offline Programming

- 6. test whether vscode can run c++ files



```cpp
#include <iostream>
int main(){
    std::cout << "Hello" << std::endl;

    return 0;
}
```

or use following commands:

```
g++ -o test test.cpp
./test
```

1.7 Learning C++

- A successful approach to learning to program in C++ depends on large amounts of **practice**.
  - exercises at the end of each chapter.
  - solutions to selected exercises are at the web site for this book

- Other learning materials

Learn C++ – Skill up with our free tutorials (learncpp.com)

https://www.bilibili.com/video/BV1oD4y1h7S3/?spm_id_from=333.337.search-card.all.click&vd_source=1e2cfa9e6c262e6d189fe4a48f2c1aac

Learn C++ | Codecademy

Homework 1

- 1. How to develop a C++ program?
- 2. Use CodeBlocks, Visual Studio Code or online programming to write a C++ program:

  Display the following two sentences on the screen:

    Hello C++.

    I like programming.

  The output is formatted as two lines.