

面向对象程序设计技术X

主讲：陈笑沙

课程材料：[GitHub](#)

课程安排

课时安排

24课时教学内容

成绩分布

80%卷面成绩，20%平时成绩

平时成绩：回答问题、作业

课程资料

<https://github.com/pdcxs/cpp-class-material>

QQ群: 1034981027



第一课

1.1 软件与硬件

1.2 编程工具介绍

1.3 开发环境搭建

1.4 编程语言派系与历史

1.5 最简单的程序

1.1 软件与硬件

硬件

- 输入设备:
 1. 键盘
 2. 鼠标
 3. 摄像头
 4.
- 输出设备:
 1. 显示器
 2. 打印机
 3.

1.1 软件与硬件

硬件

- 计算机体系架构:
 1. 内存
 2. 硬盘
 3. CPU
 4. GPU
 5.

1.1 软件与硬件

软件

- 操作系统：
 1. Windows
 2. Linux
 3. Mac
- 编程相关：
 1. 编译器
 2. 解释器
 3. 编辑器
 4. IDE
- 其他（办公、通信、娱乐.....）

1.2 编程工具介绍

编程语言分类

1.2 编程工具介绍

编程语言分类

- 编译型语言

1.2 编程工具介绍

编程语言分类

- 编译型语言
- 解释型语言

1.2 编程工具介绍

编程语言分类

- 编译型语言
- 解释型语言
- 静态类型语言

1.2 编程工具介绍

编程语言分类

- 编译型语言
- 解释型语言
- 静态类型语言
- 动态类型语言

1.2 编程工具介绍

编程语言分类

- 编译型语言
- 解释型语言
- 静态类型语言
- 动态类型语言
- 强类型语言

1.2 编程工具介绍

编程语言分类

- 编译型语言
- 解释型语言
- 静态类型语言
- 动态类型语言
- 强类型语言
- 弱类型语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

编程语言分类

- 过程式语言
- 面向对象语言
- 函数式语言
- 声明式语言
- 标记语言

1.2 编程工具介绍

C++

C++是一种静态类型语言

C++是一种弱类型语言

C++是一种编译型语言

C++是一种多范式语言

1.3 开发环境搭建

参考视频:

环境搭建:

<https://www.bilibili.com/video/BV1beNfepExE>

1.3 开发环境搭建

1、使用 `xboxdownload` 访问 Github

1.3 开发环境搭建

2、使用 scoop 集中管理软件

环境变量设置

添加软件源

安装vscode、mingw、cmake、xmake

1.3 开发环境搭建

3、配置vscode

安装 C/C++ 插件

安装 Xmake

安装简体中文

1.3 开发环境搭建

传统编译过程

手动设置第三方库

```
g++ -c source.cpp
```

```
g++ source.o
```

手写 Makefile

1.3 开发环境搭建

现代化编译过程

cmake

xmake

.....

1.3 开发环境搭建

作业

在任意平台上搭建好C++开发环境，编译并运行Hello World程序。

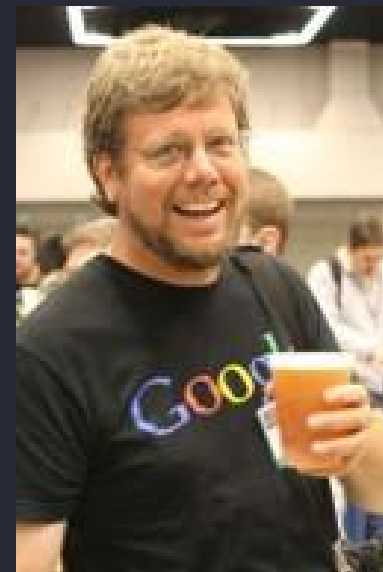
1.4 编程语言派系与历史



C++创始人
Bjarne Stroustrup



Java 创始人
James Gosling



Python 创始人
Guido van Rossum

1.4 编程语言派系与历史

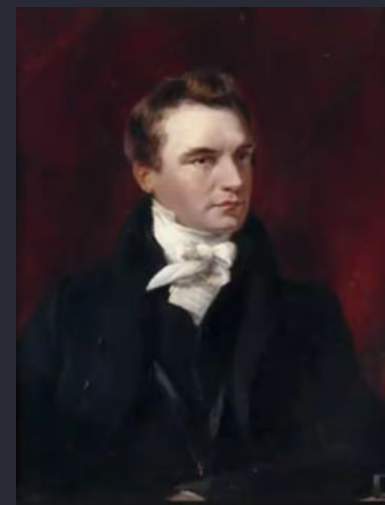
古早时期：机器语言



英国浪漫主义诗人
- 拜伦



埃达·洛夫莱斯



查尔斯·巴贝奇

1.4 编程语言派系与历史

古早时期：机器语言



分析机

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 of 799).

Number of Operations	Variable operating upon	Variable receiving results	Definition of change in the value of any Variable	Statement of Results	Data												Working Variables												Result Variables			
					x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}		
1	$x_1 \leftarrow x_1 + x_2$	x_1	$x_1 = x_1 + x_2$	$x_1 = 2x_2$	1	0	0	0	0	0	0	0	0	0	0	0																
2	$x_2 \leftarrow x_2 - x_1$	x_2	$x_2 = x_2 - x_1$	$x_2 = -x_1$	0	1	0	0	0	0	0	0	0	0	0	0																
3	$x_3 \leftarrow x_1 + x_2$	x_3	$x_3 = x_1 + x_2$	$x_3 = 0$	1	0	1	0	0	0	0	0	0	0	0	0																
4	$x_4 \leftarrow x_2 + x_3$	x_4	$x_4 = x_2 + x_3$	$x_4 = x_3$	0	1	0	1	0	0	0	0	0	0	0	0																
5	$x_5 \leftarrow x_3 + x_4$	x_5	$x_5 = x_3 + x_4$	$x_5 = 2x_3$	0	0	1	1	0	0	0	0	0	0	0	0																
6	$x_6 \leftarrow x_4 + x_5$	x_6	$x_6 = x_4 + x_5$	$x_6 = 3x_3$	0	0	1	2	0	0	0	0	0	0	0	0																
7	$x_7 \leftarrow x_5 + x_6$	x_7	$x_7 = x_5 + x_6$	$x_7 = 5x_3$	0	0	1	3	1	0	0	0	0	0	0	0																
8	$x_8 \leftarrow x_6 + x_7$	x_8	$x_8 = x_6 + x_7$	$x_8 = 8x_3$	0	0	1	4	2	0	0	0	0	0	0	0																
9	$x_9 \leftarrow x_7 + x_8$	x_9	$x_9 = x_7 + x_8$	$x_9 = 13x_3$	0	0	1	5	3	0	0	0	0	0	0	0																
10	$x_{10} \leftarrow x_8 + x_9$	x_{10}	$x_{10} = x_8 + x_9$	$x_{10} = 21x_3$	0	0	1	6	4	0	0	0	0	0	0	0																
11	$x_{11} \leftarrow x_9 + x_{10}$	x_{11}	$x_{11} = x_9 + x_{10}$	$x_{11} = 34x_3$	0	0	1	7	5	0	0	0	0	0	0	0																
12	$x_{12} \leftarrow x_{10} + x_{11}$	x_{12}	$x_{12} = x_{10} + x_{11}$	$x_{12} = 55x_3$	0	0	1	8	6	0	0	0	0	0	0	0																
13	$x_{13} \leftarrow x_{11} + x_{12}$	x_{13}	$x_{13} = x_{11} + x_{12}$	$x_{13} = 89x_3$	0	0	1	9	7	0	0	0	0	0	0	0																
14	$x_{14} \leftarrow x_{12} + x_{13}$	x_{14}	$x_{14} = x_{12} + x_{13}$	$x_{14} = 144x_3$	0	0	1	10	8	0	0	0	0	0	0	0																
15	$x_{15} \leftarrow x_{13} + x_{14}$	x_{15}	$x_{15} = x_{13} + x_{14}$	$x_{15} = 233x_3$	0	0	1	11	9	0	0	0	0	0	0	0																
16	$x_{16} \leftarrow x_{14} + x_{15}$	x_{16}	$x_{16} = x_{14} + x_{15}$	$x_{16} = 377x_3$	0	0	1	12	10	0	0	0	0	0	0	0																
17	$x_{17} \leftarrow x_{15} + x_{16}$	x_{17}	$x_{17} = x_{15} + x_{16}$	$x_{17} = 610x_3$	0	0	1	13	11	0	0	0	0	0	0	0																
18	$x_{18} \leftarrow x_{16} + x_{17}$	x_{18}	$x_{18} = x_{16} + x_{17}$	$x_{18} = 987x_3$	0	0	1	14	12	0	0	0	0	0	0	0																
19	$x_{19} \leftarrow x_{17} + x_{18}$	x_{19}	$x_{19} = x_{17} + x_{18}$	$x_{19} = 1597x_3$	0	0	1	15	13	0	0	0	0	0	0	0																
20	$x_{20} \leftarrow x_{18} + x_{19}$	x_{20}	$x_{20} = x_{18} + x_{19}$	$x_{20} = 2584x_3$	0	0	1	16	14	0	0	0	0	0	0	0																
21	$x_{21} \leftarrow x_{19} + x_{20}$	x_{21}	$x_{21} = x_{19} + x_{20}$	$x_{21} = 4181x_3$	0	0	1	17	15	0	0	0	0	0	0	0																
22	$x_{22} \leftarrow x_{20} + x_{21}$	x_{22}	$x_{22} = x_{20} + x_{21}$	$x_{22} = 6765x_3$	0	0	1	18	16	0	0	0	0	0	0	0																
23	$x_{23} \leftarrow x_{21} + x_{22}$	x_{23}	$x_{23} = x_{21} + x_{22}$	$x_{23} = 10946x_3$	0	0	1	19	17	0	0	0	0	0	0	0																
24	$x_{24} \leftarrow x_{22} + x_{23}$	x_{24}	$x_{24} = x_{22} + x_{23}$	$x_{24} = 17711x_3$	0	0	1	20	18	0	0	0	0	0	0	0																
25	$x_{25} \leftarrow x_{23} + x_{24}$	x_{25}	$x_{25} = x_{23} + x_{24}$	$x_{25} = 28657x_3$	0	0	1	21	19	0	0	0	0	0	0	0																
26	$x_{26} \leftarrow x_{24} + x_{25}$	x_{26}	$x_{26} = x_{24} + x_{25}$	$x_{26} = 46368x_3$	0	0	1	22	20	0	0	0	0	0	0	0																
27	$x_{27} \leftarrow x_{25} + x_{26}$	x_{27}	$x_{27} = x_{25} + x_{26}$	$x_{27} = 74477x_3$	0	0	1	23	21	0	0	0	0	0	0	0																
28	$x_{28} \leftarrow x_{26} + x_{27}$	x_{28}	$x_{28} = x_{26} + x_{27}$	$x_{28} = 12094551x_3$	0	0	1	24	22	0	0	0	0	0	0	0																
29	$x_{29} \leftarrow x_{27} + x_{28}$	x_{29}	$x_{29} = x_{27} + x_{28}$	$x_{29} = 19646102x_3$	0	0	1	25	23	0	0	0	0	0	0	0																
30	$x_{30} \leftarrow x_{28} + x_{29}$	x_{30}	$x_{30} = x_{28} + x_{29}$	$x_{30} = 315217x_3$	0	0	1	26	24	0	0	0	0	0	0	0																
31	$x_{31} \leftarrow x_{29} + x_{30}$	x_{31}	$x_{31} = x_{29} + x_{30}$	$x_{31} = 514232x_3$	0	0	1	27	25	0	0	0	0	0	0	0																
32	$x_{32} \leftarrow x_{30} + x_{31}$	x_{32}	$x_{32} = x_{30} + x_{31}$	$x_{32} = 809459x_3$	0	0	1	28	26	0	0	0	0	0	0	0																
33	$x_{33} \leftarrow x_{31} + x_{32}$	x_{33}	$x_{33} = x_{31} + x_{32}$	$x_{33} = 1291682x_3$	0	0	1	29	27	0	0	0	0	0	0	0																
34	$x_{34} \leftarrow x_{32} + x_{33}$	x_{34}	$x_{34} = x_{32} + x_{33}$	$x_{34} = 2036611x_3$	0	0	1	30	28	0	0	0	0	0	0	0																
35	$x_{35} \leftarrow x_{33} + x_{34}$	x_{35}	$x_{35} = x_{33} + x_{34}$	$x_{35} = 3205823x_3$	0	0	1	31	29	0	0	0	0	0	0	0																
36	$x_{36} \leftarrow x_{34} + x_{35}$	x_{36}	$x_{36} = x_{34} + x_{35}$	$x_{36} = 5042434x_3$	0	0	1	32	30	0	0	0	0	0	0	0																
37	$x_{37} \leftarrow x_{35} + x_{36}$	x_{37}	$x_{37} = x_{35} + x_{36}$	$x_{37} = 7841117x_3$	0	0	1	33	31	0	0	0	0	0	0	0																
38	$x_{38} \leftarrow x_{36} + x_{37}$	x_{38}	$x_{38} = x_{36} + x_{37}$	$x_{38} = 12094551x_3$	0	0	1	34	32	0	0	0	0	0	0	0																
39	$x_{39} \leftarrow x_{37} + x_{38}$	x_{39}	$x_{39} = x_{37} + x_{38}$	$x_{39} = 19646102x_3$	0	0	1	35	33	0	0	0	0	0	0	0																
40	$x_{40} \leftarrow x_{38} + x_{39}$	x_{40}	$x_{40} = x_{38} + x_{39}$	$x_{40} = 315217x_3$	0	0	1	36	34	0	0	0	0	0	0	0																
41	$x_{41} \leftarrow x_{39} + x_{40}$	x_{41}	$x_{41} = x_{39} + x_{40}$	$x_{41} = 514232x_3$	0	0	1	37	35	0	0	0	0	0	0	0																
42	$x_{42} \leftarrow x_{40} + x_{41}$	x_{42}	$x_{42} = x_{40} + x_{41}$	$x_{42} = 809459x_3$	0	0	1	38	36	0	0	0	0	0	0	0																
43	$x_{43} \leftarrow x_{41} + x_{42}$	x_{43}	$x_{43} = x_{41} + x_{42}$	$x_{43} = 1291682x_3$	0	0	1	39	37	0	0	0	0	0	0	0																
44	$x_{44} \leftarrow x_{42} + x_{43}$	x_{44}	$x_{44} = x_{42} + x_{43}$	$x_{44} = 2036611x_3$	0	0	1	40	38	0	0	0	0	0	0	0																
45	$x_{45} \leftarrow x_{43} + x_{44}$	x_{45}	$x_{45} = x_{43} + x_{44}$	$x_{45} = 3205823x_3$	0	0	1	41	39	0	0	0	0	0	0	0																
46	$x_{46} \leftarrow x_{44} + x_{45}$	x_{46}	$x_{46} = x_{44} + x_{45}$	$x_{46} = 5042434x_3$	0	0	1	42	40	0	0	0	0	0	0	0																
47	$x_{47} \leftarrow x_{45} + x_{46}$	x_{47}	$x_{47} = x_{45} + x_{46}$	$x_{47} = 7841117x_3$	0	0	1	43	41	0	0	0	0	0	0	0																
48	$x_{48} \leftarrow x_{46} + x_{47}$	x_{48}	$x_{48} = x_{46} + x_{47}$	$x_{48} = 12094551x_3$	0	0	1	44	42	0	0	0	0	0	0	0																
49	$x_{49} \leftarrow x_{47} + x_{48}$	x_{49}	$x_{49} = x_{47} + x_{48}$	$x_{49} = 19646102x_3$	0	0	1	45	43	0	0	0	0	0	0	0																
50	$x_{50} \leftarrow x_{48} + x_{49}$	x_{50}	$x_{50} = x_{48} + x_{49}$	$x_{50} = 315217x_3$	0	0	1	46	44	0	0	0	0	0	0	0																
51	$x_{51} \leftarrow x_{49} + x_{50}$	x_{51}	$x_{51} = x_{49} + x_{50}$	$x_{51} = 514232x_3$	0	0	1	47	45	0	0	0	0	0	0	0																
52	$x_{52} \leftarrow x_{50} + x_{51}$	x_{52}	$x_{52} = x_{50} + x_{51}$	$x_{52} = 809459x_3$	0	0	1	48	46	0	0	0	0	0	0	0																

1.4 编程语言派系与历史

20世纪30年代



Haskell Brooks
Curry
美国数理学家
组合逻辑



Alonzo Church
阿隆佐·邱奇
Lambda 演算

1.4 编程语言派系与历史



艾伦·麦席森·图灵

1937年 提出图灵机理论
1939年 破译德军密码

1.4 编程语言派系与历史



约翰·冯·诺依曼

1945年 提出冯诺依曼结构

1.4 编程语言派系与历史



凯瑟琳·布恩
1947年创造汇编语言

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Machine Language vs. Assembly Language

Objective: Multiply the value stored in R4 by **120**

Machine Language

```
0011 0000 0000 0000
0101 000 000 1 00000
0101 111 111 1 00000
0001 000 000 1 ?????
0000 110 000000011
0001 111 111 000 100
0001 000 000 1 11111
0000 111 111111100
1111 0000 0010 0101
```

Assembly Language

```
.ORIG x3000
AND R0, R0, #0
AND R7, R7, #0
ADD R0, R0, ???
TEST BRnz DONE
ADD R7, R7, R4
ADD R0, R0, #-1
BRnzp TEST
DONE HALT
.END
```

7-4

1.4 编程语言派系与历史



约翰巴科斯
IBM 公司

```
PROGRAM TRIVIAL
  INTEGER I
  I = 2
  IF(I .GE. 2) CALL PRINTIT
  STOP
END
SUBROUTINE PRINTIT
  PRINT *, 'Hola Mundo'
  RETURN
END
```

Fortran语言
1966年: Fortran66
Latest: Fortran 2023

1.4 编程语言派系与历史



约翰麦卡锡
IBM 公司

```
(defun all-pairs (M N)
  (if (null M) nil
      (append (dist1 (car M) N)
                (all-pairs (cdr M) N))))

(defun dist1 (x N)
  (if (null N) nil
      (cons (list x (car N))
              (dist1 x (cdr N)))))
```

Lisp语言

1.4 编程语言派系与历史

1. 1958年
2. 美国美国计算机协会、联邦德国应用数学和力学协会
3. ALGOL58
4. ALGOL60中，引入递归、作用域、代码块等概念

```
BEGIN
FILE F(KIND=REMOTE);
  EBCDIC ARRAY E[0:11];
  REPLACE E BY "HELLO WORLD!";
  WRITE(F, *, E);
END.
```

```
begin
  printf(($gl$, "Hello, World!"))
end
```

1.4 编程语言派系与历史

1958年



格蕾丝·赫柏
IBM 公司

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAIN.  
  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
    01 WS-STUDENT-ID PIC 9(4) VALUE 1000.  
    01 WS-STUDENT-NAME PIC A(15) VALUE 'Tim'.  
  
PROCEDURE DIVISION.  
    CALL 'UTIL' USING WS-STUDENT-ID, WS-STUDENT-NAME.  
    DISPLAY 'Student Id : ' WS-STUDENT-ID  
    DISPLAY 'Student Name : ' WS-STUDENT-NAME  
    STOP RUN.
```

COBOL语言, 宏、数据结构、Bug

1.4 编程语言派系与历史

1966年



艾伦·凯
面向对象编程概念

1.4 编程语言派系与历史

1970年

尼克劳斯维尔特: Pascal语言

肯·汤普逊: B语言、Unix操作系统

1.4 编程语言派系与历史

1972年

艾伦·凯
Smalltalk

```
m1 := MyClass new.  
m2 := MyClass new.  
(m1 equals: m2) ifTrue: [  
    Transcript show: 'They are equal'  
] else: [  
    Transcript show: 'They are not equal'  
]
```

1.4 编程语言派系与历史

1973年



肯·汤普逊
丹尼斯·里奇

```
#include <stdio>
int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

C 语言

1.4 编程语言派系与历史

1975年



杰拉德·杰伊·萨斯曼
麻省理工学院



盖伊·史提尔二世
麻省理工学院

```
(define my-counter
  (let ((count 0))
    (lambda ()
      (set! count (+ count 1))
      count)))
```

Scheme 语言 闭包

1.4 编程语言派系与历史

1983年



比雅尼·斯特劳斯特鲁普
贝尔实验室

C++语言

1.4 编程语言派系与历史

- 1992年
- 俄勒冈波特兰的函数式编程语言与计算机结构大会成立专门委员会，形成第一个Haskell版本
- 惰性函数式编程语言

```
1 primes = filterPrime [2..] -- 筛法求素数
2   where
3     filterPrime (p : xs) =
4       p : filterPrime [x | x <- xs, x `mod` p /= 0]
5
6 -- 快速排序
7 qsort [] = []
8 qsort (x : xs) = qsort (filter (<= x) xs) ++ [x] ++
9                   qsort (filter (> x) xs)
```

1.4 编程语言派系与历史

- 1992年
- 俄勒冈波特兰的函数式编程语言与计算机结构大会成立专门委员会，形成第一个Haskell版本
- 惰性函数式编程语言

```
1 primes = filterPrime [2..] -- 筛法求素数
2   where
3     filterPrime (p : xs) =
4       p : filterPrime [x | x <- xs, x `mod` p /= 0]
5
6 -- 快速排序
7 qsort [] = []
8 qsort (x : xs) = qsort (filter (<= x) xs) ++ [x] ++
9                   qsort (filter (> x) xs)
```

1.4 编程语言派系与历史

1995年

詹姆斯·高斯林

Sun微系统公司

1.5 最简单的程序

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, world!" << endl;
6     return 0;
7 }
```

C++ Hello World 程序

1.5 最简单的程序

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, world!" << endl;
6     return 0;
7 }
```

头文件从 `stdio.h` 变为了 `iostream`

1.5 最简单的程序

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, world!" << endl;
6     return 0;
7 }
```

使用命名空间避免命名冲突。

1.5 最简单的程序

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, world!" << endl;
6     return 0;
7 }
```

`main` 函数为返回整型的函数，返回0表示运行成功。

1.5 最简单的程序

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello, world!" << endl;
6     return 0;
7 }
```

新的输入/输出方法。

课后作业

简单，但是重要！

在任意平台上搭建好C++开发环境，编译并运行Hello World程序。