

# C++ 笔记

shiyu-hong

2025-04-04



# 目录

欢迎	5
第一章 未定义行为	7
1.1 未初始化变量 . . . . .	7
1.1.1 局部变量未初始化（最常见风险之一） . . . . .	7
1.1.2 数组未初始化 . . . . .	8
1.1.3 类成员变量未初始化 . . . . .	8
1.1.4 指针未初始化（最常见风险之一） . . . . .	9



# 欢迎

欢迎阅读这份 C++ 学习笔记!



# 第一章 未定义行为

C++ 未定义行为（Undefined Behavior, UB）是指代码中某些不符合语言规范的操作，其执行结果在 C++ 标准中未被明确定义。这类行为不会触发编译错误，但可能导致程序崩溃、输出错误结果，甚至表现出与硬件或编译器实现相关的不可预测现象。

## 1.1 未初始化变量

未初始化变量是指程序中声明了变量但未赋予初始化，直接使用其内存中的【残留值】的行为。这种行为属于未定义（Undefined Behavior, UB），可能导致程序输出不可预测的结果、崩溃或逻辑错误。

### 1.1.1 局部变量未初始化（最常见风险之一）

```
#include <iostream>

void unsafe_calculation();

int main(int argc, char **argv) {
    unsafe_calculation();
    return 0;
}

void unsafe_calculation() {
    int a; // 危险：未初始化局部变量
    int b = 10;
    int result = a + b; // 未定义行为（未初始化的变量 a 可能包含垃圾值）

    std::cout << result << std::endl; // 输出随机垃圾值
}
```

- **输出可能：** 10（MSVC）、521112049（GCC）等任意值。
- **风险：** 不同编译器/运行环境可能输出不同结果，甚至触发段错误。

### 1.1.2 数组未初始化

```
#include <iostream>

void process_array();

int main(int argc, char **argv) {
    process_array();
    return 0;
}

void process_array() {
    int buffer[3]; // 危险：未初始化数组

    for (auto i = 0; i < 3; ++i) {
        buffer[i] += 1; // 未定义行为（操作未初始化的值）
    }

    for (auto i = 0; i < 3; ++i) {
        std::cout << buffer[i] << " "; // 输出随机垃圾值
    }
}
```

- 输出可能：8 1 1（MSVC）、230 -1387049927 682（GCC）等任意值。
- 风险：不同编译器/运行环境可能输出不同结果，甚至触发段错误。

### 1.1.3 类成员变量未初始化

```
#include <iostream>

class Point {
public:
    void print() { std::cout << "(" << x_ << ", " << y_ << ")" << std::endl; }

private:
    int x_; // 未在构造函数中初始化
    int y_; // 未在构造函数中初始化
};

void log_point();

int main(int argc, char **argv) {
    log_point();
}
```



```
    return 0;
}

void log_point() {
    Point point;    // 危险：未显示初始化成员
    point.print(); // 输出随机垃圾值
}
```

- 输出可能：(0, 0) (MSVC)、(521112039, 32760) (GCC) 等任意值。
- 风险：不同编译器/运行环境可能输出不同结果，甚至触发段错误。

### 1.1.4 指针未初始化（最常见风险之一）

```
#include <iostream>

void dangerous_pointer_operation();

int main(int argc, char **argv) {
    dangerous_pointer_operation();
    return 0;
}

void dangerous_pointer_operation() {
    int *ptr; // 危险：未初始化指针
    *ptr = 3; // 未定义行为（可能覆盖随机内存，触发段错误）
}
```

- 风险：可能覆盖随机内存，触发段错误。