

# 数据结构与算法实验说明

## 实验安排

实验总学时：24学时

实验一：线性结构 4学时

实验二：二叉树 6学时

实验三：树 2学时

实验四：图 4学时

实验五：二叉搜索树与平衡二叉搜索树 4学时

实验六：散列 2学时

实验七：排序 2学时

每次实验结束后一周内，提交电子版实验报告。

## 实验报告要求

实验报告的主要内容是报告自己在实验过程中所做的主要工作，包括算法设计、程序设计及测试、实验结果分析等。实验报告应简单明了，语言通顺，图表数据齐全规范。实验报告的主要内容包括：

1. 实验目的及内容：参照实验指导书撰写，也可以结合自己的实际情况撰写。
2. 基本原理及算法设计：根据实验要求写出算法设计思想，要求用伪码描述，最好不要复制大段代码。
3. 测试及调试：实验中遇到了哪些问题，是怎样解决的。
4. 实验结果分析：对原始记录进行必要的分析、整理。包括程序运行结果与预期结果的比较，产生错误的原因及修改方法等。
5. 总结本次实验的体会和收获，例如对算法的理解及算法分析，程序测试与调试方法，以及实验中所获得的经验和教训等。

## 实验环境

使用符合C++标准的编译环境进行实验。

搭建编程环境是程序员的基本技能。请掌握如何自己搭建C++编译环境，不要依赖于直接使用实验室已经搭建好的平台。

使用C++编程理论上只需要一个C++编译器和一个文本编辑软件。文本编辑软件编写cpp源文件，C++编译器负责将源文件编译成可执行文件。由于C++编译器是没有用户界面的命令程序，一般都会使用文本编辑器及其相关插件配合工作，同时某些高级文本编辑软件也提供代码格式关键字高亮、代码自动生成、代码跳转等功能。

也可以直接使用集成开发环境来进行C++编程，所谓集成开发环境就是软件商将编译器以及一个专门为C++编程设计的文本编辑软件打包在一起发行的软件。

推荐使用以下环境之一进行实验：

## Windows

### VSCode + Mingw

VSCode 是微软提供的一个免费文本编辑工具，支持各种编程语言的关键字高亮、代码自动生成、代码跳转等功能。

官方网站为：<https://code.visualstudio.com>

Mingw 是开源软件 GNU编译器集合（GCC）在 Windows 平台下的发行版，提供了多种语言的编译器、调试器以及用于编译 windows 平台程序的标准库文件。最新版的 MingW 支持 C++17 语言标准。

推荐通过 MSYS2 进行安装：<https://www.msys2.org>

关于如何在 VSCode 进行 C++ 环境配置，参见：

<https://code.visualstudio.com/docs/cpp/config-mingw>

### CodeBlocks

CodeBlocks 是一个开源免费的简单基础C++集成开发环境。

CodeBlocks 本身不限定 C++ 编译器，可以配合任意编译器使用。

官方提供的下载链接中，有包含 Mingw 编译器的版本，请注意下载此版本。

官方网站为：<https://www.codeblocks.org>

**注意：** Codeblocks 安装包中所带 MingW 版本较老，支持 C++11 标准。

### 最新版本的Visual Studio

Visual Studio 是微软提供的用于开发 windows 平台上应用程序的集成开发环境，支持 C++ 和 C# 语言。最新版本的 Visual Studio 支持 C++20 语言标准。

官方网站为：<https://visualstudio.microsoft.com/zh-hans/vs/>

不要使用 VC++6.0。VC++6.0 不支持任何标准的C++。

# MAC

## VSCode + Clang

VSCode 是跨平台软件，macOS 中也可以使用。

可以在官网下载 mac 版 VSCode。

官方网站为：<https://code.visualstudio.com>

Clang 是苹果公司开发的 C++ 编译器，支持完整的 C++17 语言标准，部分 C++20 语言标准。

Clang 可以在系统终端中使用命令安装：

```
xcode-select --install
```

关于如何在 macOS 中配置 VSCode C++ 编程环境，参见：

<https://code.visualstudio.com/docs/cpp/config-clang-mac>

## XCode

XCode 是苹果公司发布的用于开发所有苹果平台上应用程序的集成开发环境。支持 C、C++、Objective-C 以及 Swift 语言。它内部的编译器实际为 Clang。

XCode 可以通过 macOS 上的 App store 进行下载安装。

# Linux

## VSCode + GCC

可以在官网下载 Linux 版 VSCode。

官方网站为：<https://code.visualstudio.com>

GCC 是开源组织 GNU 提供的编译器集合，支持的语言有 C、C++、Objective-C、Fortran、Ada、Go、以及 D，支持完整 C++17 语言标准，部分 C++20 语言标准。

不同发行版的 Linux 上 GCC 安装可能不同，请自行查找相关资料。

关于如何在 Linux 中配置 VSCode C++ 编程环境，参见：

<https://code.visualstudio.com/docs/cpp/config-linux>

## 扩展阅读：C++ 标准

ISO C++ 的全称为 ISO International Standard ISO/IEC 14882:2020(E) – Programming Language C++。

C++ 标准委员会成立于 1990-91 年，由来自 ISO/IEC JTC1/SC22 成员国认可的对 C++ 工作感兴趣的专家组成。从2011年起，C++ 标准委员会每四年公布一次新的 C++ 标准以指导各编译器开发厂商进行编译器的升级改进，以保证 C++ 语言在各平台上的一致性。每次 C++ 标准更新的各项特性，是投票决定的。目前已公布的最新标准为 C++20。

关于 C++ 标准会议的信息可参见：

<https://isocpp.org>

由于 C++ 语言编译器的设计极为复杂，且每个编译器厂商所采用的技术各有差异，并不能保证在新标准公布后所有厂商都能马上推出完全符合标准的编译器更新。同时，绝大部分 C++ 程序员都只使用 C++ 语言的某些特性子集进行工作，且更关系编译器的稳定正确性。所以，大部分编译器开发商在新标准提出后，并不会马上跟随，而是更多地在保证当前版本稳定正确性的前提下，逐步添加新标准中的特性。

目前基本所有主流 C++ 编译器均已完全支持 C++14 以上标准。编译本课程中的代码需要编译器支持 C++11 标准。

关于主流编译器对 C++ 的支持可参见：

[https://en.cppreference.com/w/cpp/compiler\\_support](https://en.cppreference.com/w/cpp/compiler_support)