

# CCF 编程能力等级认证 C++&Python 认证标准

中国计算机学会编程能力等级认证（CCF GESP）为计算机和编程学习者提供学业能力验证的平台。GESP 旨在提升青少年计算机编程能力、培训机构编程教育水平，推广和普及计算机和编程教育。考察语言为图形化编程、Python 编程及 C++ 编程。C++ & Python 编程认证划分为一至八级，通过设定不同等级的认证目标，让学习者具备计算机使用的基础能力和通过编程思维解决生活问题的能力，激发其学习编程相关知识与技术的兴趣，提高编程科学技术素养，培养编程综合实践能力，为广大学员在进修等方面提供编程能力水平的证明。

## C++&Python 认证知识体系

级别	知识内容 (C++)	知识内容 (Python)	知识目标
一级	计算机基础与编程环境 计算机历史 变量的定义与使用 基本数据类型（整型、浮点型、字符型、布尔型） 控制语句结构（顺序、循环、选择） 基本运算（算术运算、关系运算、逻辑运算） 输入输出语句	计算机基础与编程环境 计算机历史 变量的定义与使用 基本数据类型（整型、浮点型、字符型、布尔型） 控制语句结构（顺序、循环、选择） 基本运算（算术运算、关系运算、逻辑运算） 输入输出语句 Turtle 绘图	掌握顺序、循环、分支的简单程序结构，可以使用集成开发环境进行编程与调试，通过编程基础知识的学习，完成单一功能的程序设计。
二级	计算机的存储与网络 程序设计语言的特点 流程图的概念与描述 ASCII 编码 数据类型的转换 多层分支/循环结构 常用数学函数（绝对值函	计算机的存储与网络 程序设计语言的特点 流程图的概念与描述 ASCII 编码 数据类型的转换 多层分支/循环结构 简单数学函数（不含三角、	掌握程序基本设计，能够使用简单数学函数。可以独立完成包含分支语句、循环语句等比较综合的案例，可以使用分支循环嵌套结构。

	数、平方根函数、 <b>max</b> 函数 数、 <b>min</b> 函数)	对数、指数等)	
三级	数据编码（原码、反码、补码） 进制转换（二进制、八进制、十进制、十六进制） 位运算（与（&）、或（ ）、非（~）、异或（^）、左移（<<）、右移(>>)） 算法的概念与描述（自然语言描述、流程图描述、伪代码描述） <b>C++</b> 一维数组基本应用； <b>Python</b> 列表、字典、元组、集合的基本应用、内置函数以及列表解析的使用 字符串及其函数 算法：枚举法 算法：模拟法	掌握数据编码、进制转换、位运算等知识，掌握一维数组、字符串及函数的使用，能够独立使用模拟法、枚举法解决对应的算法问题。	
四级	函数的定义与调用 形参与实参、作用域 <b>C++</b> 指针类型的概念及基本应用 函数参数传递的概念（ <b>C++</b> 值传递、引用传递、指针传递； <b>Python</b> 值传递、引用传递） <b>C++</b> 结构体 <b>C++</b> 二维数组与多维数组基本应用； <b>Python</b> 复合数据类型的嵌套 算法：递推 算法：排序概念和稳定性 算法：排序算法（冒泡排序、插入排序、选择排序） 简单算法复杂度的估算（含多项式、指数复杂度） 文件重定向与文件读写操作 异常处理	掌握函数的定义、调用及函数参数传递的方法；掌握二维数组与多维数组的使用技巧；掌握常用排序算法、文件读写和异常处理的使用。能够解决递推相关问题。	
五级	初等数论 ( <b>C++</b> ) 数组模拟高精度加法、减法、乘法、除法 单链表、双链表、循环链表	掌握初等数论，线性表的知识，二分法、分治法、贪心法的思想，完成指定功能的	

	辗转相除法（也称欧几里得算法） 素数表的埃氏筛法和线性筛法 唯一分解定理 二分查找/二分答案（也称二分枚举法） 贪心算法 分治算法（归并排序和快速排序） 递归 算法复杂度的估算（含多项式、指数、对数复杂度）	程序。 <b>C++</b> 掌握数组模拟高精度的运算。
六级	树的定义，构造与遍历 哈夫曼树 完全二叉树 二叉排序树 哈夫曼编码 格雷编码 深度优先搜索算法 宽度优先搜索算法（也称广度优先搜索算法） 二叉树的搜索算法 简单动态规划（一维动态规划、简单背包问题） 面向对象的思想 类的创建 栈、队列、循环队列	掌握树的基础知识，能够分辨不同的树，并根据不同的搜索算法进行遍历，掌握简单线性动态规划和简单背包问题。
七级	数学库常用函数（三角、对数、指数） 复杂动态规划（二维动态规划、动态规划最值优化） 图的定义及遍历 图论基本算法（图的深度优先遍历、广度优先遍历、泛洪算法） 哈希表	掌握图的定义与遍历相关算法，掌握图论基本概念及基础算法，能使用二维动态规划、动态规划最值优化的知识完成复杂的动态规划算法
八级	计数原理 排列与组合 杨辉三角	掌握组合数学中基本知识，通过算法的时间和空间效率分析，可以完成相对应的算

	倍增法 代数与平面几何（初中数学部分） 图论算法及综合应用（最小生成树、单源最短路） 较复杂算法的空间和时间效率分析 算法优化	法优化。
--	---	------

注：对于 C++ 语言的考题，如无特殊说明，均以【C++11】标准为准。如考题有特殊说明，以考题说明为准。考题不会涉及该标准下的未定义行为（undefined behavior）。

# C++编程一级标准

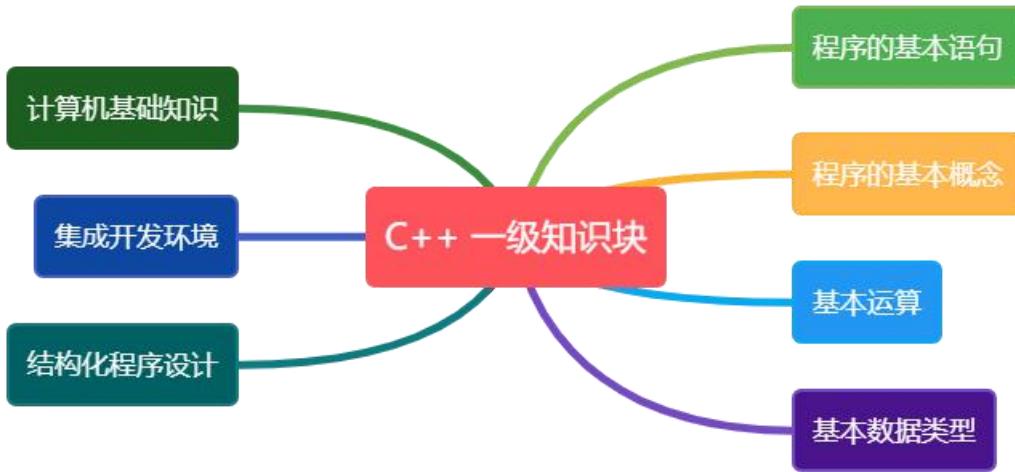
## (一) 知识点详述

- (1) 了解计算机的基本构成 (CPU, 内存, I/O 设备等)，了解 Windows、Linux 等操作系统基本概念和常见操作，了解计算机的历史及在现代社会中的常见应用。
- (2) 熟悉集成开发环境使用 (例如 Dev C++)：创建文件、编辑文件、保存文件、编译、解释、调试。
- (3) 掌握基础的 `cin` 语句、`scanf` 语句、`cout` 语句、`printf` 语句，赋值语句等。
- (4) 掌握标识符、关键字、常量、变量、表达式的概念。
- (5) 掌握常量与变量的命名、定义、作用、初始化与赋值以及变量的自加与自减运算。
- (6) 掌握基础算术表达式：加、减、乘、除、整除、求余。
- (7) 掌握逻辑运算与 (`&&`)、或 (`||`)、非 (`!`)。
- (8) 掌握关系运算：大于、大于等于、小于、小于等于、等于、不等于。
- (9) 掌握基础的数据类型的定义和使用 (整型、实数型、字符型、布尔型)。
- (10) 掌握顺序结构程序的编写。
- (11) 掌握分支结构程序的编写，掌握 `if` 语句、`if-else` 语句、`switch` 语句，了解三目运算。
- (12) 掌握循环结构程序的编写，掌握 `for`、`while`、`do-while` 循环语句的使用以及 `continue` 语句和 `break` 语句在循环中的应用。
- (13) 理解程序的注释和调试的概念。

## (二) 考核目标

学生通过计算机基础知识的学习，了解计算机的构成与操作，以及计算机的发展历程。通过编程基础知识以及语句的掌握，可以独立完成简单功能的顺序结构、分支结构、循环结构的程序。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	计算机基础知识	计算机的软硬件组成、常见操作、发展历程。
2	集成开发环境	创建文件、编辑文件、保存文件、编译、解释、调试。
3	结构化程序设计	顺序结构、分支结构、循环结构。
4	程序的基本语句	cin 语句、scanf 语句、cout 语句、printf 语句、赋值语句、复合语句、if 语句、switch 语句、for 语句、while 语句、do while 语句。
5	程序的基本概念	标识符、关键字、常量、变量、表达式的概念。 常量与变量的命名、定义、作用。 程序的注释。
6	基本运算	算术运算、逻辑运算、关系运算、变量自增与自减运算、三目运算。
7	基本数据类型	整数型: int, long long 实数型: float, double 字符型: char

		布尔型: bool
--	--	-----------

## (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# Python 编程一级标准

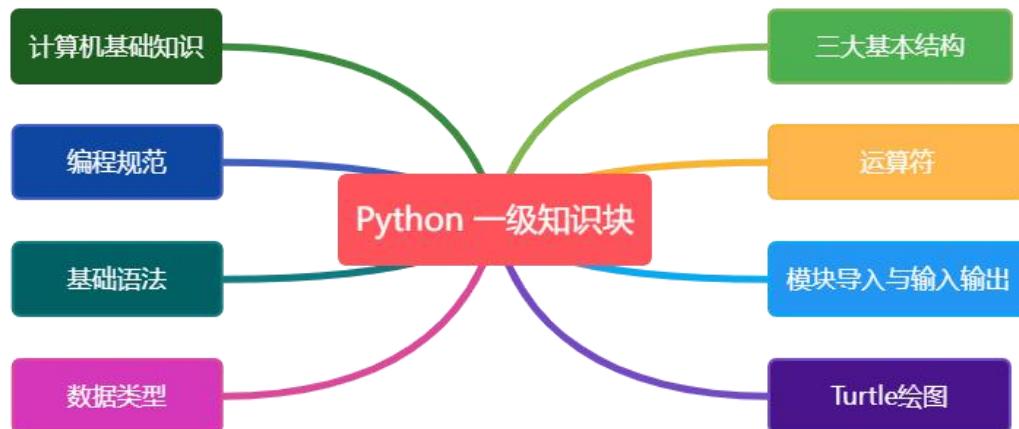
## (一) 知识点详述

- (1) 了解 Windows、Linux 等操作系统的 basic 概念及常见操作，了解计算机硬件的基本组成结构。
- (2) 了解计算机网络协议和互联网的基本概念。
- (3) 了解计算机语言的基本概念与转换，文件存储的类型与大小的概念，掌握编程文件创建、复制、粘贴、删除、移动程序和调试的基本操作。
- (4) 掌握编程语言开发环境的使用（如 DEV C++、PyCharm、IDLE、Visual Studio 等）。
- (5) 理解并掌握“输入、处理、输出”程序编写方法，掌握 Python 语言编写的基本格式：如缩进、空格、括号、注释等编码规范。
- (6) 掌握标识符、关键字、常量、变量的命名规则和使用方法。
- (7) 了解程序的顺序结构、选择结构、循环结构。
- (8) 了解数字类型、字符串类型和布尔类型的初级使用。
- (9) 掌握比较运算符、算术运算符、逻辑运算符的基本概念及基础应用。
- (10) 掌握变量的创建及使用。
- (11) 掌握输入输出语句 `input` 和 `print`。
- (12) 掌握图形库 `turtle` 的主要功能，使用 `turtle` 进行绘图。
- (13) 掌握模块的导入方法。

## (二) 考核目标

学生对计算机系统的编程软件的界面认识和基本操作，能够独立创建完整的编程文件并运行通过，并实现通过导入 `turtle` 绘图模块学会图像绘制并掌握数据类型的使用，实现编程入门，同时针对参加一级考试的学生将进行简单的逻辑推理能力的考查。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	计算机基础知识	运行 Python 环境 鼠标、键盘等硬件设备的操作及软件的打开与操作、 计算机文件类型（文本，视频，音频）创建、复制、 粘贴、删除、移动保存编程文件
2	编程规范	缩进、空格、括号、注释、换行的使用
3	基础语法	标识符、关键字、常量、变量
4	数据类型	数字、字符串、布尔类型
5	三大基本结构	顺序、分支、循环
6	运算符	算术运算符：+、-、*、/ 、% 逻辑运算符：and 、or、 not 比较运算符：==、 !=、 >、 <、 >=、 <=
7	模块导入与输入输出	import、from、input()和 print()
8	Turtle 绘图	Turtle 绘图指令（前进、转弯、填色、抬笔等）

#### (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# C++ 编程二级标准

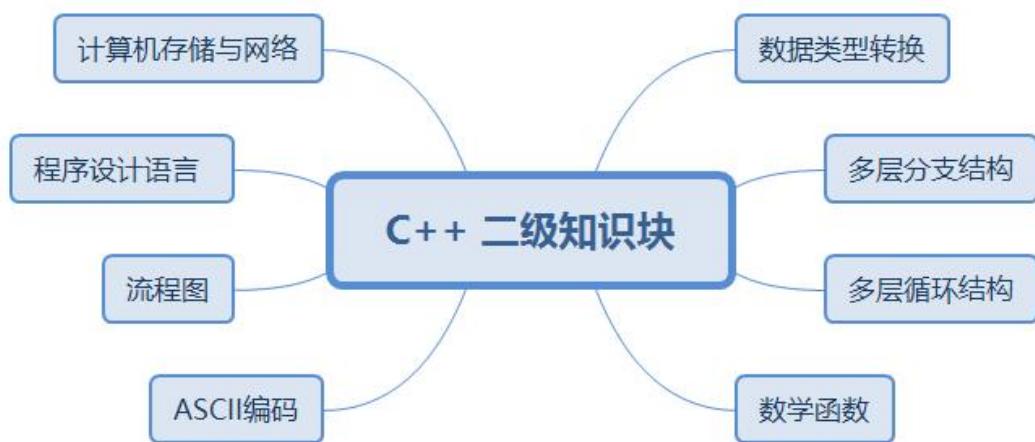
## (一) 知识点详述

- (1) 了解计算机存储的基本概念及分类，了解随机存储器（RAM）、只读存储器（ROM）和高速缓冲存储器（Cache）的功能及区别。
- (2) 了解计算机网络的概念，了解计算机网络的分类（广域网（WAN）、城域网（MAN）、局域网（LAN）），了解计算机网络的层级结构及作用（TCP/IP 四层模型与 OSI 七层模型），了解不同层级的重要协议，了解 IP 地址及子网划分。
- (3) 了解程序设计语言的几大分类及特点（机器语言、汇编语言、高级语言），了解常见的高级语言（C++、Python 等）。
- (4) 了解流程图的概念及基本表示符号，掌握绘制流程图的方法，能正确使用流程图描述程序设计的三种基本结构。
- (5) 了解编码的基本概念，了解 ASCII 编码原理，能识别常用字符的 ASCII 码（空格：32、“0”：48、“A”：65、“a”：97），并掌握 ASCII 码和字符之间相互转换的方法。
- (6) 掌握数据类型的转换：强制类型转换和隐式类型转换。
- (7) 掌握多层分支结构，掌握 if 语句、if...else 语句、switch 语句，及相互嵌套的方法。
- (8) 掌握多层循环结构，掌握 for 语句、while 语句、do...while 语句，及相互嵌套的方法。
- (9) 掌握常用的数学函数：绝对值函数、平方根函数、最大值函数、最小值函数、随机数函数理解相应的算法原理。

## (二) 考核目标

通过计算机基础知识的学习，了解计算机的存储与网络知识、程序设计语言分类及特点、常见的编程语言和绘制流程图的方法。通过 C++ 知识的学习，掌握数据类型的转换方法及数学库函数的使用，可以独立完成多分支结构与循环结构的程序。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	计算机存储与网络	ROM、RAM、CACHE 计算机网络分类 TCP/IP 四层模型与 OSI 七层模型 IP 地址及子网划分
2	程序设计语言	程序设计语言分类 常见的高级语言
3	流程图	流程图的概念、绘制流程图、描述流程图
4	ASCII 编码	常见字符的 ASCII 编码、字符编码之间的相互转换
5	数据类型转换	强制类型转换 隐式类型转换
6	多层分支结构	if 语句、if...else 语句、switch 语句的嵌套
7	多层循环语句	while 循环、do...while 循环、for 循环的嵌套
8	数学函数	绝对值函数: abs() 平方根函数: sqrt()

		最大值函数: <code>max()</code> 最小值函数: <code>min()</code> 随机数函数: <code>rand()/srand()</code> 及相关
--	--	--

## (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# Python 编程二级标准

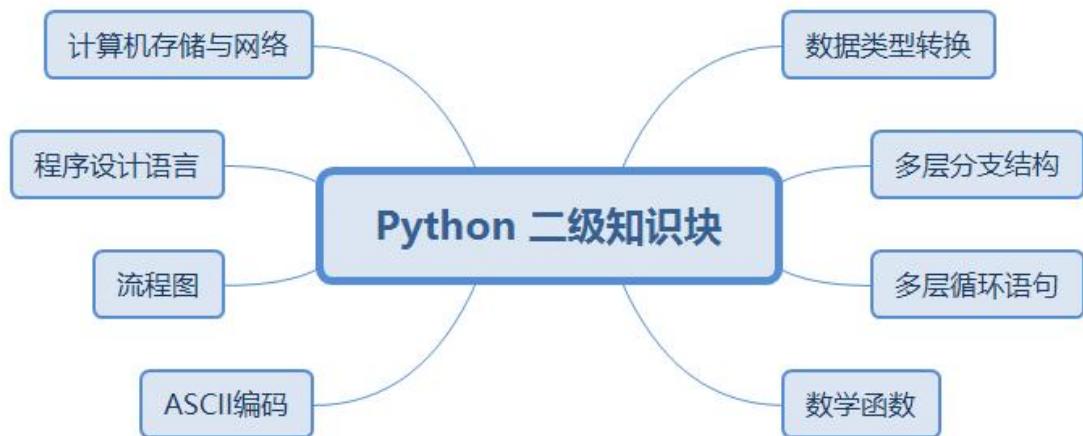
## (一) 知识点详述

- (1) 了解计算机存储的基本概念及分类，了解随机存储器（RAM）、只读存储器（ROM）和高速缓冲存储器（Cache）的功能及区别。
- (2) 了解计算机网络的概念，了解计算机网络的分类（广域网（WAN）、城域网（MAN）、局域网（LAN）），了解计算机网络的层级结构及作用（TCP/IP 四层模型与 OSI 七层模型），了解不同层级的重要协议，了解 IP 地址及子网划分。
- (3) 了解程序设计语言的几大分类及特点（机器语言、汇编语言、高级语言），了解常见的高级语言（C++、Python 等）。
- (4) 了解流程图的概念及基本表示符号，掌握绘制流程图的方法，能正确使用流程图描述程序设计的三种基本结构。
- (5) 了解编码的基本概念，了解 ASCII 编码原理，能识别常用字符的 ASCII 码（空格：32、“0”：48、“A”：65、“a”：97），并掌握 ASCII 码和字符之间相互转换的方法。
- (6) 掌握数据类型的转换：强制类型转换和隐式类型转换。
- (7) 掌握多层分支结构，掌握 if 语句、if...else 语句、elif 语句及相互嵌套的方法。
- (8) 掌握多层循环结构，掌握 for 语句、while 语句及相互嵌套的方法。
- (9) 掌握简单的数学函数，如绝对值函数、平方根函数、最大值函数、最小值函数、随机数函数、round()函数等，理解其算法原理（不含三角、对数、指数等）。

## (二) 考核目标

通过计算机基础知识的学习，了解计算机的存储与网络知识、程序设计语言分类及特点、常见的编程语言和绘制流程图的方法。通过 Python 知识的学习，掌握数据类型的转换方法及相关数学库函数的使用，可以独立完成多分支结构与循环结构的程序。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	计算机存储与网络	ROM、RAM、CACHE 计算机网络分类 TCP/IP 四层模型与 OSI 七层模型 IP 地址及子网划分
2	程序设计语言	程序设计语言分类 常见的高级语言
3	流程图	流程图的概念、绘制流程图、描述流程图
4	ASCII 编码	常见字符的 ASCII 编码、字符编码之间的相互转换
5	数据类型转换	强制类型转换 隐式类型转换
6	多层分支结构	if 语句、if...else 语句、elif 语句的嵌套
7	多层循环语句	while 循环、for 循环的嵌套
8	数学函数	绝对值函数: abs() 平方根函数: sqrt() 最大值函数: max() 最小值函数: min()

	四舍五入函数: <code>round()</code> , 相关随机函数
--	--

## (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# C++&Python 编程三级标准

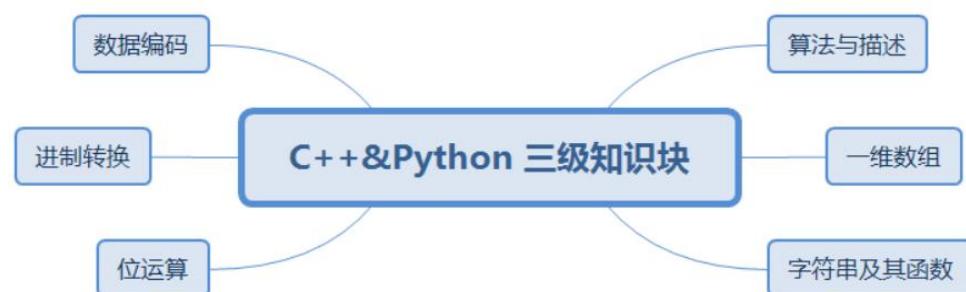
## (一) 知识点详述

- (1) 了解二进制数据编码:原码、反码、补码。
- (2) 掌握数据的进制转换: 二进制、八进制、十进制、十六进制。
- (3) 掌握位运算: 与(&)、或(|)、非(~)、异或(^)、左移(<<)、右移(>>)的基本使用方法及原理。
- (4) 了解算法的概念与描述, 熟练运用自然语言、流程图、伪代码方式来描述算法。
- (5) C++一维数组基本应用 (不包括变长数组); Python 列表、字典、元组、集合的基本应用、内置函数以及列表解析的使用.
- (6) 掌握字符串及其函数的使用包括但不限于大小写转换、字符串搜索、分割、替换。
- (7) 理解枚举算法、模拟算法的原理及特点, 可以解决实际问题。
- (8) 理解模拟算法、模拟算法的原理及特点, 可以解决实际问题。

## (二) 考核目标

掌握计算机中常用进位制、位运算及数据编码的知识, 掌握一维数组、字符串类型及其函数的使用, 掌握枚举法、模拟法的原理和运用技巧, 对于较简单的实际问题能构造算法、描述算法、实现算法并调试程序。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	数据编码	原码、反码、补码
2	进制转换	二进制、八进制、十进制、十六进制
3	位运算	与 ( <code>&amp;</code> )、或 ( <code> </code> )、非 ( <code>~</code> )、异或 ( <code>^</code> )、左移 ( <code>&lt;&lt;</code> )、右移 ( <code>&gt;&gt;</code> )
4	算法与描述	枚举法、模拟法 自然语言描述、流程图描述、伪代码描述
5	数据结构	C++一维数组 (不包括变长数组); Python 列表、字典、元组、集合、列表解析
6	字符串及其函数	大小写转换、字符串搜索、分割、替换等

#### (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# C++&Python 编程四级标准

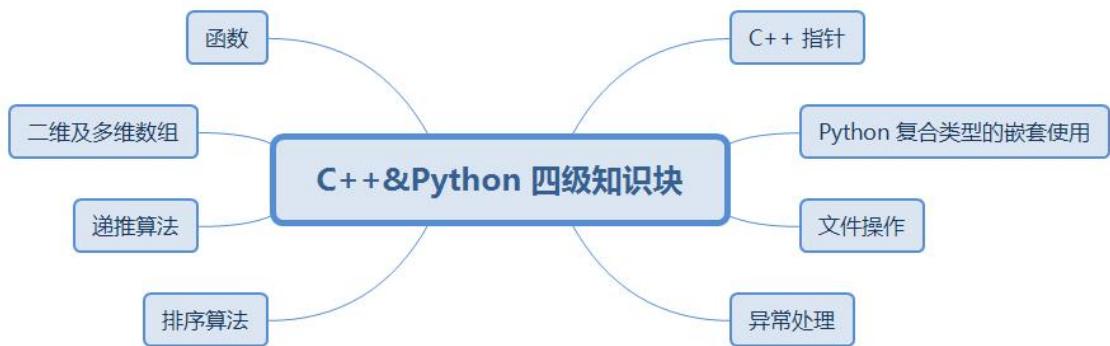
## (一) 知识点详述

- (1) 理解 C++指针类型的概念，掌握指针类型变量的定义、赋值、解引用。
- (2) 掌握 C++结构体、二维及多维数组（不包括变长数组）的基本概念及使用；掌握 Python 复合数据类型的嵌套使用。
- (3) 理解模块化编程思想，掌握函数（不包含匿名函数、类的括号运算符重载）的声明、定义及调用，掌握形参与实参的概念及区别。
- (4) 掌握变量作用域的概念，理解全局变量与局部变量的区别。
- (5) 掌握函数参数的传递方式：C++值传递、引用传递、指针传递；Python 值传递、引用传递。
- (6) 掌握递推算法基本思想、递推关系式的推导以及递推问题求解。
- (7) 掌握排序算法的概念，了解内排序和外排序的概念及差别，理解排序算法的时间复杂度、空间复杂度、使用场景以及稳定性。
- (8) 掌握排序算法中的冒泡排序、插入排序、选择排序的算法思想、排序步骤及代码实现。
- (9) 简单算法复杂度的估算，含多项式、指数复杂度。
- (10) 掌握文件操作中的重定向，实现文件读写操作，了解文本文件的分类，掌握写操作、读操作、读写操作。
- (11) 了解异常处理机制，掌握异常处理的常用方法。

## (二) 考核目标

掌握 C++指针类型、二维及多维数组（不包括变长数组）的基本使用；掌握 Python 复合类型的嵌套使用。通过函数相关知识的学习，掌握模块化设计思想，具备编写自定义函数程序的能力。掌握文件读写操作，并通过对排序算法、递推法的学习，可以根据不同的使用场景，合理选择最优的算法。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	指针	指针类型, 指针类型定义变量, 指针类型变量的赋值、解引用
2	二维及多维数组	C++二维及多维数组的定义、使用 Python 复合类型的嵌套使用
3	结构体	结构体定义和使用, 结构体数组, 结构体指针, 结构体嵌套结构体, 结构体做函数参数, 结构体中 <code>const</code> 使用场景
4	函数	函数的定义、调用、声明 (不包含匿名函数、类的括号运算符重载) 形参、实参 全局作用域、局部作用域 值传递、引用传递
5	递推算法	递推算法基本思想、递推关系式推导
6	排序算法	冒泡排序、插入排序、选择排序 时间复杂度、空间复杂度、算法稳定性 简单算法复杂度的估算, 含多项式、指数复杂度
7	文件操作	文件重定向, 读操作、写操作、读写操作

8	异常处理	异常处理机制和常用方法
---	------	-------------

### (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 120 分钟

# C++&Python 编程五级标准

## (一) 知识点详述

- (1) 掌握初等数论相关知识的概念和应用，包括素数与合数、最大公约数与最小公倍数、同余与模运算、约数与倍数、质因数分解、奇偶性等。
- (2) 掌握 C++ 数组模拟高精度加法、减法、乘法和除法的相关知识。
- (3) 掌握链表的创建、插入、删除、遍历和反转操作，理解单链表、双链表、循环链表的区别。
- (4) 掌握辗转相除法（也称欧几里得算法）、素数表的埃氏筛法和线性筛法、唯一分解定理的原理和应用。
- (5) 掌握算法复杂度估算方法（含多项式、对数）。
- (6) 掌握二分查找和二分答案算法（也称二分枚举法）的基本原理，能够在有序数组中快速定位目标值。
- (7) 掌握递归算法的基本原理，能够应用递归解决问题，能够分析递归算法的时间复杂度和空间复杂度，了解递归的优化策略。
- (8) 掌握贪心算法的基本原理，理解最优子结构，能够使用贪心算法解决相关问题。
- (9) 掌握分治算法的基本原理，能够使用归并排序和快速排序对数组进行排序。

## (二) 考核目标

掌握初等数论知识点，能够使用辗转相除法（也称欧几里得算法）、素数表的埃氏筛法和线性筛法、唯一分解定理等相关知识解决相应的问题。掌握单链表、双链表、循环链表的基本操作方法。掌握算法复杂度估算方法（含多项式、对数），熟悉二分法、分治法、贪心算法和递归算法的算法思想，能够根据实际情况选择合适的算法并完成解决相应的问题。C++ 掌握使用数组模拟高精度加法、减法、乘法和除法的知识。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	初等数论	素数与合数、最大公约数与最小公倍数、同余与模运算、约数与倍数、质因数分解、奇偶性 欧几里得算法 唯一分解定理 素数表的埃氏筛法和线性筛法
2	算法复杂度估算方法	含多项式的算法复杂度 含对数的算法复杂度
3	C++高精度运算	C++数组模拟高精度加法、减法、乘法、除法
4	链表	单链表、双链表、循环链表的创建、插入、删除、遍历、查找的基本操作
5	二分算法	二分查找算法 二分答案算法（也称二分枚举法）
6	递归算法	递归算法的相关概念 递归算法的时间复杂度和空间复杂度 递归的优化策略
7	分治算法	归并排序算法 快速排序算法

8	贪心算法	贪心算法的相关概念 最优子结构
---	------	--------------------

## (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间：180 分钟

# C++&Python 编程六级标准

## (一) 知识点详述

- (1) 掌握树的基本概念，掌握其构造与遍历的相关算法。
- (2) 掌握哈夫曼树、完全二叉树、二叉排序树的相关概念和应用。
- (3) 理解哈夫曼编码、格雷编码相关原理并能进行简单应用。
- (4) 掌握深度优先搜索算法（DFS）、宽度优先搜索算法（也称广度优先搜索算法，BFS）、二叉树的搜索算法的概念及应用，能够根据现实问题，选择合适的搜索算法。
- (5) 掌握简单动态规划的算法思想，能够使用代码解决相应的一维动态规划问题和简单背包问题。
- (6) 掌握面向对象的思想，了解封装、继承、多态的基本概念，并掌握类的创建和基本的使用方法。
- (7) 掌握栈、队列、循环队列的基本定义，应用场景和常见操作。

## (二) 考核目标

掌握树的基础知识，并能够分辨和使用哈夫曼树、完全二叉树、二叉排序树。掌握搜索算法，可以根据不同的实际问题选择最优的搜索算法。掌握动态规划的思路和步骤，能够解决一维动态规划问题和简单背包问题。掌握面向对象的概念和特性，了解与面向过程思想的不同之处，并掌握类的创建及其基本使用方法。掌握栈、队列、循环队列的基本定义和常见操作，并可根据实际情况选择合适的数据结构。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	树	树的基本概念 哈夫曼树 完全二叉树 二叉排序树
2	基于树的编码	格雷编码 哈夫曼编码
3	搜索算法	深度优先搜索算法（DFS） 宽度优先搜索算法（也称广度优先搜索算法，BFS） 二叉树的搜索算法
4	简单动态规划	一维动态规划 简单背包
5	面向对象	面向对象思想 类的创建和初始化 类的特性:继承、封装、多态
6	栈和队列	栈 队列 循环队列

#### (五) 题型分布

单选题	判断题	编程题
15道（2分/道）	10道（2分/道）	2道（25分/道）

考试时间：180分钟

# C++&Python 编程七级标准

## (一) 知识点详述

- (1) 掌握数学库常用函数（三角、对数、指数），三角函数包括  $\sin(x)$ ,  $\cos(x)$  等；对数函数包括  $\log_{10}(x)$ : 返回  $x$  以 10 为底的对数,  $\log_2(x)$ : 返回  $x$  以 2 为底的对数；指数函数包括  $\exp(x)$ : 计算指数函数，返回  $x$  的以  $e$  为底的指数函数。
- (2) 掌握复杂动态规划（二维动态规划、动态规划最值优化）。包括区间动态规划、最长上升子序列（LIS）、最长公共子序列（LCS）等内容，理解基于滚动数组等降低动态规划空间复杂度的方法。
- (3) 图的定义及基本图论算法。包括图的定义、图的种类（有向图、无向图），图节点的度的概念。掌握编程时图的数据结构表示，以及基于深度优先搜索（DFS）和广度优先搜索（BFS）的图搜索与遍历方法，图的泛洪（flood fill）算法。
- (4) 掌握哈希表的概念与知识及其应用。

## (二) 考核目标

掌握常用数学库函数，了解相关函数概念与定义。掌握复杂动态规划，包括二维动态规划、求 LIS、LCS 等内容，并掌握利用滚动数组等的优化方法。了解图的定义与广搜和深搜的算法，泛洪算法。了解哈希表的概念和知识。

## (三) 知识块



## (四) 知识点描述

编号	知识块	知识点
1	数学库函数	三角函数 对数函数 指数函数
2	复杂动态规划	二维动态规划、动态规划最值优化 区间动态规划 求最长上升子序列（LIS） 求最长公共子序列（LCS） 基于滚动数组的动态规划空间复杂度优化
3	图的定义及遍历	图的概念 图的广度优先遍历 图的深度优先遍历
4	图论算法	图的泛洪算法（flood fill）
5	哈希表	哈希表的概念与知识及其应用

## （五）题型分布

单选题	判断题	编程题
15道（2分/道）	10道（2分/道）	2道（25分/道）

考试时间：180分钟

# C++&Python 编程八级标准

## (一) 知识点详述

- (1) 掌握计数原理。包括加法原理和乘法原理。
- (2) 掌握排列与组合基础知识。包括排列、组合的基本概念，及能实现基础排列和组合编程问题的一般方法。
- (3) 掌握杨辉三角形（又称帕斯卡三角形）的概念。
- (4) 掌握倍增法概念。了解倍增法的时间复杂度。
- (5) 掌握代数与平面几何基础知识（初中数学部分）。包括方程的概念及一元一次方程、二元一次方程的基本求解技巧，求基础平面几何概念、求基本图形（如长方形、三角形、圆形等）的面积等。
- (6) 掌握图论算法及综合应用技巧。包括最小生成树的概念、kruskal 算法、prim 算法，掌握最短路径的概念、单源最短路径的 dijkstra 算法、Floyd 算法等。理解实现同一功能的不同算法的比较，并可以灵活解决相关问题。
- (7) 算法的时间和空间效率分析。能够掌握较为复杂算法的时间和空间复杂度分析方法，能够分析各类算法（包括排序算法、查找算法、树和图的遍历算法、搜索算法、分治及动态规划算法等）的时间和空间复杂度。
- (8) 算法优化。理解不同方法求解一个问题在时间复杂度和空间复杂度上的差异，理解使用数学知识辅助求解问题的技巧（如可以用循环求出等差数列的和，也可以用数学公式求出等差数列的和），掌握一般的算法优化技巧。

## (二) 考核目标

掌握基本计数原理，理解加法原理和乘法原理的区别与使用。掌握排列组合概念，能够实现常见排列组合问题的编程求解方法。掌握杨辉三角形的概念和应用，了解杨辉三角形与组合之间的关系。掌握代数与平面几何的基本知识（限初中数学），能够求解一元一次方程、二元一次方程并掌握平面几何基本知识。掌握较为复杂算法的时间复杂度和空间复杂度分析方法，及其一般的算法优化技巧，能根据数学知识优化算法。

## (三) 知识块



#### (四) 知识点描述

编号	知识块	知识点
1	计数原理	加法原理 乘法原理
2	排列与组合	排列 组合
3	杨辉三角	杨辉三角的定义 杨辉三角形的实现
4	倍增法	倍增的概念
5	代数与平面几何	一元一次方程 二元一次方程 三角形、圆形、长方形面积
6	图论算法及综合应用	最小生成树的概念、kruskal 算法、prim 算法 最短路径的概念、dijkstra 算法、Floyd 算法 图论算法的综合应用与问题求解技巧
7	算法的时间和空间效率分析	算法时间和空间复杂度的一般分析方法 各类算法（包括排序算法、查找算法、树和图的遍历算法、搜索算法、分治及动态规划算法等）的时间和空间复杂度
8	算法优化	不同算法求解问题的差异分析 算法优化的一般方法 根据数学知识优化算法的一般方法（包括但不限于等差、等比数列的求和公式等）

## (五) 题型分布

单选题	判断题	编程题
15 道 (2 分/道)	10 道 (2 分/道)	2 道 (25 分/道)

考试时间: 180 分钟