# C++基础

1. 输入输出合法检验
   1. 吃掉回车问题-std::cin.ignore(100,’\n’);
   2. 定义宏的方法
   3. C/C++不分行的
   4. cin的状态
2. 数组问题
   1. cin 不接受字符串空格问题std::getline(std::cin,str);
   2. 数组的名字的问题
3. 指针
   1. 相关说法(取地址浮&解引用\*)
   2. 指针的\*位置
   3. 指针的\*的两种作用(创建指针和解引用)
   4. 无类型指针
   5. 图说指针
   6. 指针与数组
   7. 指针++===+sizeof(指针类型)
   8. 进阶思考之指针变量与整形变量
4. 结构问题
   1. 结构的赋值(点运算符和{})
   2. 结构与指针(.和->)
5. 传值、传址、传引用
6. 联合枚举类型别名typedef

# C++高级

1. 函数重载
   1. 定义(同名不同参)
   2. 注意事项(谨慎重载，减少重载，区分覆盖，返回值重载，真正目的)
   3. 常识(<<,不行的返回值式重载函数)
2. 类与对象
   1. 作用域操作符::
   2. 构造器
      1. 与方法的三大区别(与类名同名、自动调用、无返回值)
      2. 构造对象数组
      3. 默认为空构造器
   3. 析构器
      1. 析构器与构造器的四大区别(一申一清、同名加~、无返回值无参数、内存泄露)
      2. 析构器就是虚方法
   4. this指针
      1. 使用this的基本原则
      2. 隐含的this指针
      3. this传的是对象的首地址
   5. 类的继承
      1. 基类父类超类
      2. 写法：class SubClass::public SuperClass{}
      3. 继承机制下的构造器和析构器(前者直接，后者先子后父)
   6. 访问控制
      1. C++的访问级别public,protected,private
      2. 继承机制中的访问控制public，protected
   7. 友元关系friend class \*\*\*\*
   8. 静态属性和方法
      1. 静态方法两个好处(非创建调用、全体对象共享)
      2. 五个特性(静态成员不访静态元素、非静态方法访问、无法访问this、内存分配、静态成员调用)
   9. 虚方法
      1. new 和 delete
      2. 虚方法解决方案
      3. 声明方法virtual void play();
      4. 虚方法的继承(父虚子不虚)
      5. 习惯(拿不准则为虚、虚速度慢、多层则顶级全虚)
   10. 抽象方法(纯虚方法)
       1. 语法=0
       2. 与虚方法的区别
       3. 多态性(编译重载、运行虚函数)
   11. 运算符重载
       1. 写法：Complex operator+(Complex &d);
       2. 不允许重载的运算符(. .\* :: ?: sizeof)
       3. 隐含的参数
       4. 类友元函数
       5. 重载<<操作符std::ostream& operator<<(std::ostream& os,Complex f);
   12. 多继承
   13. 虚继承：virtual public
3. 副本构造器
   1. 逐位复制
   2. MyClass &operator=(const MyClass &rhs);
   3. 副本构造器解决方案MyClass(const MyClass &chs);
4. 强制类型转换
   1. 静态对象的强制类型转换
   2. 动态对象强制类型转换dynamic\_cast<new\_type>(expression)
5. 调试与异常
   1. return式处理异常
   2. assert函数
   3. trycatch
6. 动态内存管理
   1. C++内存分配方式(堆、栈、自由存储区、全局/静态存储区和常量存储区)
   2. 动态内存
   3. 动态数组
   4. 函数返回内存(函数指针、指针函数)
   5. 避免内存泄漏
7. 模块化编程
   1. 头文件.h
      1. 作用：保证C++可移植性
      2. <和”
      3. 不保存实现
      4. 注释习惯
      5. 路径问题
   2. 实现文件.cpp
      1. 接口和实现
      2. Linux下的多个编译g++ -o rational.cpp main.cpp
   3. C预处理器
      1. 预处理器注释#if0#endif
      2. 避免重复#ifndef #define #endif
   4. 命名空间
      1. 作用域{}
      2. 命名空间
         1. 作用：
         2. 使用方法(std::cout,using namespace,using std::cout)
8. 链接
   1. 储存类(auto,static,extern,register)
   2. 链接
      1. 编译器构建程序3步骤
      2. 3种链接(外链接external、内链接interal、无链接none
9. 模板
   1. 函数模板
   2. 类模板
   3. 内联模板