

一. 绪论

1. 程序：计算机的工作都是用程序来控制的，没有程序，计算机将一事无成
指令：计算机可以识别的命令。也就是由0, 1组合的命令，一般是固定的
程序是指令的集合
2. 什么是机器语言，汇编语言和高级语言？
机器语言：由计算机硬件系统可以识别的二进制指令组成的语言成为机器语言，
汇编语言：将机器指令映射为一些可以被人读懂的助记符，例如add,sub等可以用来表示算术运算中的加减
机器语言与汇编语言与人类自然语言都相去甚远，而且不同的机器所能识别的汇编语言与机器语言存在一定的差距，程序员在编译程序时需要考虑大量的机器细节，且编译好的代码不具有泛用性
高级语言：它屏蔽了机器细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得程序书写时可以对现实中的具体事物
3. 什么是面向对象的语言？
面向对象的编程语言将客观事物看作具有属性和行为的对象，通过抽象找出同一类对象的共同属性和行为，形成类。通过类的继承和多态可以很方便地实现代码重用，使得软件风格统一。
4. 简要说明一下对象，类和封装的概念
对象：现实中的对象是现实生活中的一个事物，既可以是有形的也可以是无形的。面向对象中的对象是构成系统的一个基本单位，用来描述客观世界的一个实体，由一组属性和行为构成。
类：从同类型的对象中抽象出共性，就形成了类。换种方式说，类像一个表头，包含了某一类对象的共同属性和行为，而对象是一个表项，填入了具体的属性值。
封装：是C++的一个核心原则，就是将对象的属性和行为封装成一个整体，在保留操作接口的同时并尽可能的向外部屏蔽细节。
5. 为什么几乎所有的计算机都采用二进制？
 - a. 便于物理实现，可以用高压电平表示0, 1
 - b. 二进制运算简单
 - c. 机器可靠性高，抗干扰能力强
 - d. 通用性强
6. 位，字节，字，机器字长有什么区别？
 - ①. 位：度量数据的最小单位，表示1位二进制信息
 - ②. 字节：一个字节有8个二进制数据构成。是信息存储的基本单位
 - ③. 字：是计算机存储器的基本单位
 - ④. 机器字长：参与运算的寄存器的二进制的位数
7. 不同二进制编码方法的实质是什么？对负数表示的不同编码

二. 基本程序概念

1. 常量与符号常量

常量：程序运行过程中不可改变的量，直接使用符号（文字）表示的值。包括整型常量，实型常量，字符常量和字符串常量
符号常量：使用符号为常量命名

2. 符号常量

使用之前必须实现声明

```
const 数据类型说明符 常量名=常量值  
const float PI = 3.141592653589793
```

3. 逻辑运算——异或

两个操作数对应的每一个位进行异或，若对应位相同，则运算结果为0，若对应位不同，则运算结果为1

4. 枚举类型

```
enum 枚举类型名 {变量值列表}
```

```
Example: enum Weekday {SUN,MON,TUE,WED,THU,FRI,SAT}
```

枚举类型不能赋值，具有默认值，上述例子中分别为0, 1, 2..., 6

5. 作用域？局部变量？全局变量

作用域：作用域是一个标识符在程序正文中的有效区域。

- ①. 包括函数原型作用域，例如函数原型声明时的形参，只在形参的左右括号之间起作用。
- ②. 局部作用域，函数形参列表中形参的作用域，就是从形参列表的声明处开始，到整个函数的结束为止。
函数体内部声明的变量，作用范围就是从声明处开始，一直到这个声明所在的大括号为止。
- ③. 类作用域
- ④. 命名空间作用域。

局部变量：具有局部作用域的变量称为局部变量。

全局变量：具有命名空间作用域的变量称为全局变量，作用范围是整个文件。

三. 函数基本概念

1. 函数？

能够完成某一特定功能的一段程序，可以被重复使用，使用者不需要关心函数的内部细节，而只需要关注函数的功能。

2. 值调用和引用调用的相同点和不同点

值调用：当发生函数调用时，首先为形参分配相应的内存空间，然后用将实参的值复制给形参，这是一个参数值单项传递的过程，一旦形参得到了实参的所有值，就与实参脱离关系，相当于是实参的一个副本，对形参的任何操作都不会影响到实参。

引用调用：当发生函数调用时，系统用实参来初始化形参的引用，相当于直接对实参进行操作，形参是实参的一个别名，它们指向内存中的同一块数据单元

3. 内联函数

用关键字inline声明的函数是内联函数，编译器在编译时将函数体直接嵌入在每一个调用处，节省了参数传递和控制转移的开销。一般是调用频率高且规模较小的函数声明为内联函数。

内联函数的特点：

- ①. 内联函数不能包含循环和switch等复杂结构
- ②. 内联函数的声明必须在第一次调用内联函数之前
- ③. 对内联函数不能进行异常接口声明

4. 什么是函数的重载？

有两个以上的同名函数，它们的形参类型或者形参的个数不同，编译器在编译时将根据实参和形参的类型以及形参个数找到最佳匹配的函数，这就是函数的重载。

四. 类与对象

1. 构造函数？析构函数？

构造函数：构造函数就是在对象被创建的时候用一组特定的值来构造对象，将对象初始化为一个特定状态，与该类的其他对象进行区分，构造函数将在创建对象时系统自动调用。特点：函数名与类名相同，没有返回值，可以重载

析构函数：析构函数主要在对象被删除前进行一些清理工作，在对象的生存周期结束时系统自动调用，释放对象所占用的内存空间。特点：析构函数不接受任何参数。一个类可以有多个构造函数，但是只能由1个析构函数。

2. 拷贝构造函数？何时被调用

拷贝构造函数是一种特殊的构造函数，具有构造函数的全部特性，其形参是本类对象的引用，用来实现利用一个已经存在的对象，创建一个同类的新对象。

三种调用：用类的一个对象初始化该类的另一个对象；函数的形参是类对象，函数形参和实参结合时将调用拷贝构造函数；函数的返回值是类对象，函数调用完成返回时需要调用拷贝构造函数。

五. 数据的共享与保护

1. 作用域？局部变量？全局变量

作用域：作用域是一个标识符在程序正文中的有效区域。

- ①. 包括函数原型作用域，例如函数原型声明时的形参，只在形参的左右括号之间起作用。
- ②. 局部作用域，函数形参列表中形参的作用域，就是从形参列表的声明处开始，到整个函数的结束为止。
函数体内部声明的变量，作用范围就是从声明处开始，一直到这个声明所在的大括号为止。
- ③. 类作用域
- ④. 命名空间作用域。

局部变量：具有局部作用域的变量称为局部变量。

全局变量：具有命名空间作用域的变量称为全局变量，作用范围是整个文件。

2. 静态数据成员？

静态数据成员主要解决一个类的不同对象之间数据共享问题的。静态数据成员用static关键字进行声明，与普通数据成员不同的是，普通数据成员在每一个对象中都有一份拷贝，这也是同类的不同对象之间进行区分的主要特征，而静态数据成员在所有对象中只有一份拷贝，由该类的所有对象共同维护。可以说，静态数据成员不属于任何一个对象，而是属于类的，可以通过类名直接进行访问。

3. 友元函数？友元类？

友元函数是通过关键字friend声明的非成员函数，该函数可以直接调用该类的私有成员和保护成员。如果一个类被声明为友元类，则它的全部成员函数都是友元函数。

六. 数组，指针和字符串

1. 指针型函数和指向函数的指针？

指针型函数是函数，函数的返回值是指针类型。

指向函数的指针是指针，指针中存放的地址是函数代码的起始地址，也就是函数名。当函数指针指向某个具体的函数后，就可以使用调用函数的方法调用指针。

2. 深拷贝与浅拷贝？

浅拷贝：也就是默认复制构造函数，只是简单的把两个对象的数据元素——复制，如果数据元素中存在指针等数据成员，浅拷贝可能会造成两个对象的数据成员指向同一个地址，并没有创造真正的副本，在析构函数释放内存空间时会产生错误。

深拷贝：为新对象的数据成员开辟新的地址空间，与已有对象的地址空间区分开来。

3. 常量指针与指针常量

常量指针：指向常量的指针，不能通过指针修改所指向对象的值，但是可以改变指针本身指向另外的值

指针常量：指针类型的常量，指针本身的值不能被改变

七. 继承与派生

1. 什么是继承？派生？

类的继承：新的类从已有类那里得到已有的特性

类的派生：从一个已有类产生新类

2. 派生类生成过程？

吸收基类成员、改造基类成员、增加新的成员

3. 三种继承方式的区别？

public:使得基类中的public和protected成员在派生类中的访问权限不变，但private成员在派生类中不可访问

private:使得基类中的public和protected成员在派生类中变为private成员，而基类中的private成员在派生类中不可访问

protected:使得基类中的public和protected成员在派生类中变为protected成员，而基类中的private成员在派生类中不可访问

private与protected的主要区别在于：通过private方式继承的派生类，如果继续派生，那么新产生的派生类不能访问间接基类的任何数据元素。而protected方式继承的派生类，如果继续派生，则新产生的派生类有可能访问间接基类中的public和protected成员

4. 派生类构造函数的执行顺序

①. 如果该类由直接或者间接的虚基类，则先执行虚基类的构造函数

②. 该类还有其他基类，则按照这些基类在继承声明的顺序依次调用构造函数

③. 按照类定义中出现的顺序，对新增数据成员对象进行初始化

④. 执行构造函数的函数体

5. 虚基类？有什么作用

当某类的部分或者全部基类都是从另一个基类中继承而来时，这些直接基类中，从上一层基类中继承的成员就具有相同的名称，派生类的对象在同名成员在内存中拥有多个拷贝，此时，可以使用作用域分辨符来唯一标识它们。也可以将共同基类声明为虚基类，这样虚基类的成员在内存中就只有一个副本，解决了同名成员的唯一标识问题。

6. 组合和继承有什么共同点和差异？

组合相当于一个类的数据成员是另一个类的对象，是整体与部分之间的关系。例如一辆汽车由轮子和发动机组成，汽车是整体，而轮子和发动机都是汽车的一部分。

继承是从一般到特殊的过程，一般来说，越底层的类越抽象，包含了某类物体的共同特征，比如学生是一个抽象类，而高中生，大学生是具体的类，这就是从一般到特殊的过程。

八、多态性

1. 多态？

同样的消息被不同对象接受后产生不同的行为。简单来说，就是同一段程序代码所传递的实参不同，最后得到的结果也不同

2. 多态的类型？

①. 重载多态：普通函数以及类的成员函数的重载都属于重载多态。运算符重载也属于函数重载，是重载多态

②. 强制多态：将一个变量的类型强制改变，以满足程序或者操作的要求

③. 包含多态：定义于不同类的同名函数的多态行为，主要通过虚函数来实现。

④. 参数多态：程序所处理的对象的类型参数化，使得同一段程序可以处理不同类型的对象

3. 从实现的角度，多态可以分为哪几类？

①. 编译时的多态：在程序编译的过程中就已经确定了同名函数的具体操作对象

②. 运行时的多态：在程序运行过程中才动态确定了函数的具体操作对象

4. 静态绑定？动态绑定？

绑定：计算机程序自身彼此关联的过程，也是一条消息和一个对象的方法相结合的过程。

静态绑定：绑定工作在编译连接阶段就完成了的情况称之为静态绑定。即在编译、链接的过程中，系统根据类型匹配及最佳个数匹配等原则确定某一个同名标识符到底调用哪一段代码。例如重载、强制和参数多态都属于静态绑定

动态绑定：绑定工作在程序运行阶段才完成的情况就是动态绑定。包含多态的操作对象的确定就是通过动态绑定完成的。

5. 运算符重载？

对已有运算符赋予新的含义，使得同一个运算符面对不同类型的对象时产生不同的结果。

6. 运算符重载的形式？

- ①. 重载为类的非静态成员函数。如果是重载为成员函数，则函数的参数要比原来的操作数的个数少一个，因为函数体把该类对象作为第一个操作数，隐藏了。后置运算符的形参列表有一个整型形参，用来区分前置++，--与后置的++，--
- ②. 重载为类的非成员函数，如果需要访问类的私有成员，可以把该函数声明为友元函数。例如重载输出流运算符，就必须将该重载函数声明为友元函数，因为该函数的左操作数是输出流对象，而不是本类的对象。

7. 虚函数的实现方式？

根据赋值兼容的规则，可以用基类类型的指针指向派生类的对象，如果派生类中有与基类同名的函数，那么通过基类类型的指针访问该同名函数，只能访问到基类的函数。但是如果将基类的同名函数设置为虚函数，就可以使用基类类型的指针就可以访问到指针指向的派生类对象同名函数。因此，通过基类指针指向不同的派生类，就可以实现不同的操作，从而实现了动态绑定。

8. 抽象类？纯虚函数？

纯虚函数：在基类中声明的虚函数，它没有定义具体的操作内容，需要派生类根据实际需求自己给出定义。纯虚函数的函数体由派生类给出。

抽象类：带有纯虚函数的类是抽象类。抽象类的作用是它为一个类群的派生类提供一个公共的接口，使它们能够有效的发挥多态特性。

9. 为什么构造函数不能是虚函数？而析构函数可以

虚函数是动态绑定的基础，主要是针对对象的，而构造函数是在创建对象之前调用的，所以虚构造函数没有存在的意义。

但是析构函数是在对象消亡前做一些清理工作，如果一个类的析构函数是虚函数，那么由这个类派生的所有类的构造函数也都是虚函数，这样就能保证使用基类指针指向不同的派生类对象进行不同的清理工作。

九、模板

1. 模板？

采用类型作为参数的程序设计方式，主要包括函数模板和类模板，用户可以编写与类型无关的代码，更好的实现代码重用。

十、STL及相关概念

1. 什么是STL

支持C++泛型编程的模板库，可以利用已经编译好的模板实现自定义程序功能

2. 什么是容器

存储一些列数据元素的对象

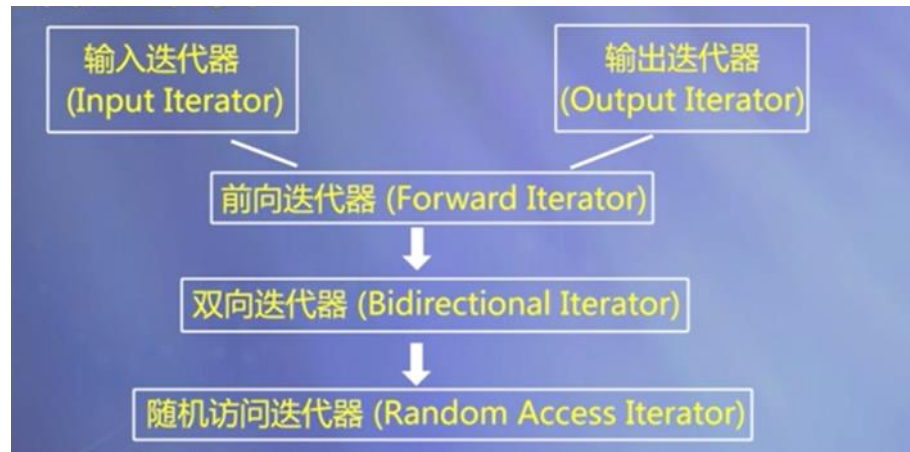
3. 什么是迭代器

迭代器是一种泛化的指针，可以对容器中的元素序列进行遍历，而不需要知道这个容器中的元素在内存中的具体位置

迭代器的分类：

- 1) 输入迭代器：可以从序列中读取数据

- 2) 输出迭代器：可以向序列中写入数据
- 3) 前向迭代器：既包含输入迭代器，有包含输出迭代器，但是只允许对序列进行单向遍历
- 4) 双向迭代器：在前向迭代器的基础上，可以对序列进行双向遍历
- 5) 随机迭代器：在双向迭代器的基础上，支持向前或者向后移动n个数据元素，可以做到和指针一样的功能



4. 关联容器

关联容器每个元素都有一个键，可以根据键高效的查找对应的元素

按照键值是否唯一:单重关联容器与多重关联容器

按照键和元素之间的关系:简单关联容器与二元关联容器

5. 集合

集合可以用来存储没有重复的元素，而且这些元素在集合中是有序排列的，可以高效的查找到某个指定的元素，同时，也可以查找到指定范围大小的元素所处的区间

6. 什么是函数对象