1. **面向对象编程**
   1. **面向过程和面向对象的编程风格**

* **面向过程的编程风格**：程序是由模块构成的，一个模块就是一个过程，C++是C语言发展而来的，因此C++在某种意义上也是面向过程语言。同时面向过程语言是自顶向下设计的，也就是将一个大问题不断分解成小问题，然后每个小问题就是一个过程，或者说函数。因此这样的编程风格有一个问题就是不方便后期维护，一旦需求有变或者串联改变时，修改维护的工程量比较大。
* **面向对象的编程风格：**面向对象中构成模块的基本单元是**类**，不是过程。类可以理解为是对象的集合。一个类里面包含两个属性，分别是静态属性和动态属性（个人通俗的分类，方便记忆）。静态属性就是我们说的属性或者变量，动态属性是指方法或者过程或者操作。面向对象的编程中如何设计类是重点，也是基础。同时，类与类之间的关系、类与对象之间的关系也是面向对象编程的重要内容，我们统称为**关系**。

**1.2类和抽象数据类型（理解有点模糊）**

* **信息隐藏：**C++中由接口和实现两个概念，接口是公有（public）的，也就是用户可见的，实现是私有的（private），也就是用户不可见的。 这样做的好处是可以对部分信息进行私有隐藏，修改的话不影响公有接口的使用，这样可以保证维护时不影响用户的使用。
* **封装：**把数据和处理数据的过程封装在一起就形成了类，说白了就是将这些信息封装在一起，方便对象使用。
* **抽象数据类型：**当一个数据类型将接口暴露出来，将实现隐藏起来，我们就称这类数据类型为抽象数据类型。计算机中抽象的概念非常重要，这样的思想非常有利于面向对象编程。

**1.3客户/服务器模式和消息传递**

* **客户/服务器模式：**面向对象程序中，类和对象可以理解为服务器，而使用类和对象的模块理解为客户。
* **消息传递：**一般来说，使用类或者对象的模块在调用函数的时候采用的就是消息传递的机制，通过这个机制来获取客户所需的服务。而具体这个服务是如何实现的，客户是不关心的，它关心的只是它要的服务是否实现。

**1.4继承和多态**

* **继承：**继承的层次是由一些具有父子关系的类组成。所谓的继承，是子类继承父类所有的特点，包括父类的所有变量和操作，同时在此基础上，子类可以拥有一些只属于自己的变量和操作。同时在C++中支持多重继承，也就是说一个子类可以由多个父类。
* **多态：**多个不同的类中可以拥有相同的函数，比如window类中有display函数，messagewindow类中也可以有display函数，而使用的时候直接用w.display（）就可以实现对应的功能，其中w可以是window或者messagewindow。判读w是指向哪个对象不是用户的事情，而是系统来完成的。

**1.5接口和组件**

* **接口：**类暴露出来的部分称为接口，是用户可以进行调用的，通过接口可以实现对应的功能，而接口里面的东西是隐藏的，也是用户不需要关心的。
* **组件：**组件是接口的后端服务者。软件组件跟硬件组件类似，软件中的容器跟硬件中的总线类似。组件和容器的有机结合构成了程序。

1. **从C到C++**
2. **类**
3. **继承**
4. **多态**
5. **操作符重载**
6. **模板与标准模板库**
7. **C++输入输出类层次**
8. **MFC中的面向对象编程**