* **考试题型及分值分布**

期末考试共四大类型题目：填空题（10题，其中包含计算题和名词理解题）、选择题（15道单项选择题）、程序填空题（3道题，每题约4~8分）、编程题（3道题，总分51分，最后一题21分）。

* **程序填空题**

每道题给出一个程序框架，需补充三个空格中的代码。最后一次课已讲解相关题型，继承相关内容为必考内容。

* **编程题**

三道编程题中，第一道较简单，第二道中等难度，第三道最难（21分），且重点考察“继承”与“存在”的编程应用。

* **复习建议与提醒**

建议重点复习课本内容、课后习题（如第3章打印星号图形、第9章构造与析构函数）以及最后一次课程所讲重点。实验报告是关键，未完成将影响考试表现。

* **各章节复习重点**

1. **初步知识**

程序设计基本概念、面向对象特性（封装、继承、多态）。

1. **基本概念**
2. 面向对象技术：面向对象技术是一种以对象为基础，以事件或消息来驱动对象执行处理的程序设计技术。
3. 对象：对象（Object）由属性（Attribute）和行为（Action）两部分组成。对象是包含客观世界物体特征的抽象实体，是属性和行为的封装体。
4. **面向对象特性**
5. 抽象性：抽象就是忽略事物中与当前目标无关的非本质特征，更充分地注意与当前目标有关的本质特征。
6. 封装性：封装就是把对象的属性和行为结合成一个独立的单位，并尽可能隐蔽对象的内部细节。
7. 继承性：继承是一种联结类与类的层次模型。继承性是指特殊类的对象拥有其一般类的属性和行为的特性。
8. 多态性：多态是指类中同一函数名对应多个具有相似功能的不同函数。多态性是指可以使用相同的调用方式来调用这些具有不同功能的同名函数的特性。
9. **数据的存储、表示形式和基本运算**

数据类型（整型、实型等）、变量定义规则、命名规范。

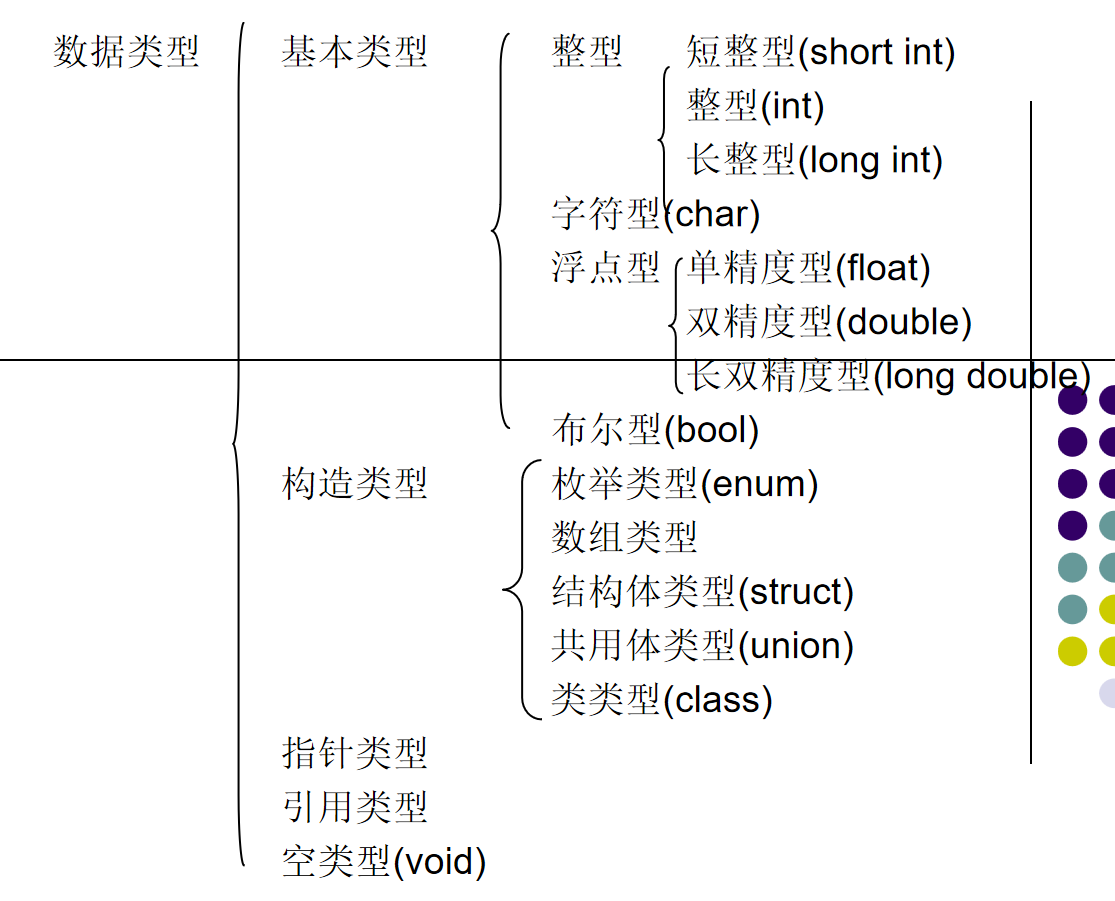
1. **变量定义规则**

变量的定义不仅声明变量的存在，还会为变量分配存储空间，并可以初始化变量的值。例如：

int b = 10; // 定义变量b，并初始化为10

\*变量不可重复定义\*

1. **命名规范**
2. 变量名只能由字母、数字和下划线组成：例如，a123、\_myVar、my\_var是合法的，而my-var、my var是非法的。
3. 变量名不能包含字母、数字、下划线以外的特殊字符：例如，my%var、my#var是非法的，因为%和#是特殊字符。
4. 变量名必须以字母或下划线开头，不能以数字开头：例如，1var是非法的，而var1、\_var是合法的。
5. 变量名不能包含空白字符：例如，my var是非法的，因为包含空格。
6. 变量名不能与C语言的预定义关键字冲突：例如，int、float、char是C语言的关键字，不能作为变量名。
7. 变量名区分大小写：例如，Var和var是两个不同的变量。
8. **数据类型**



1. **面向过程的程序设计**

程序结构（顺序、选择、循环）、break 与 continue 的区别、伪代码与流程图表示算法。

1. **程序结构**
2. C++支持最基本的三种程序运行结构:**顺序结构、选择结构、循环结构**
   1. 顺序结构:程序按顺序执行，不发生跳转
   2. 选择结构:依据条件是否满足，有选择的执行相应功能
      1. If语句：执行满足条件的语句
      2. 三目运算符：实现简单的判断
      3. switch语句：执行多条件分支语句
   3. 循环结构:依据条件是否满足，循环多次执行某段代码
      1. while 循环语句：while(循环条件) { 循环语句 }
      2. do…while循环：do { 循环语句 } while ( 循环条件 );与while的区别在于do…while会先执行一次循环语句，再判断循环条件
      3. for循环：for(起始表达式;条件表达式;末尾循环体) { 循环语句; }
      4. 嵌套循环：在循环体中再嵌套一层循环。
   4. 跳转语句：break语句、continue 语句、goto语句（标记）
3. **break 与 continue 的区别**

break：中止执行switch或循环语句。一般格式为：

break;

其作用为使流程从循环体内跳出循环体，即提前结束循环，接着执行循环体下面的语句。break语句只能用于循环语句和switch语句内，不能单独使用或用于其他语句中。

continue：结束本次循环语句。一般格式为：

continue;

其作用为结束本次循环，即跳过循环体中下面尚未执行的语句，接着进行下一次是否执行循环的判定。

说明：continue语句和break语句的区别是：continue语句只结束本次循环，而不是终止整个循环的执行。而break语句则是结束整个循环过程，不再判断执行循环的条件是否成立。

1. **伪代码与流程图表示算法**
2. **函数**

函数概念、递归函数、变量作用域（局部与全局）、存储类型（静态、动态、寄存器变量）。

1. **函数概念：**为了便于规划、组织、编程和调试，一般的做法是把一个大的程序划分为若干个程序模块(即程序文件)，每一个模块实现一部分功能。
2. **递归函数：**在调用一个函数的过程中又出现直接或间接地调用该函数本身，称为函数的递归调用。

int f(int x)

{int y,z;

z=f(y); //在调用函数f的过程中，又要调用f函数

return (2\*ｚ);

}

1. **变量作用域：**指程序中可以引用该变量的区域。

全局作用域、局部作用域、命名空间作用域、类作用域、块作用域、函数原型作用域

1. **存储类型**
2. 从**作用域**角度分，如下：
   1. 局部变量
      1. 自动变量,即动态局部变量(离开函数,值就消失)
      2. 静态局部变量(离开函数,值仍保留)
      3. 寄存器变量(离开函数,值就消失)
      4. 形式参数(可以定义为自动变量或寄存器变量)
   2. 全局变量
      1. 静态外部变量(只限本文件引用)
      2. 外部变量(即非静态的外部变量,允许其他文件引用)
3. 从**变量存储期(存在的时间)**来区分,静态存储是程序整个运行时间都存在,而动态存储则是在调用函数时临时分配单元：
   1. 动态存储
      1. 自动变量(本函数内有效)
      2. 寄存器变量(本函数内有效)
      3. 形式参数
   2. 静态存储
      1. 静态局部变量(函数内有效)
      2. 静态外部变量(本文件内有效)
      3. 外部变量(其他文件可引用)
4. 从**变量值存放的位置**来区分,可分为：
   1. 内存中静态存储区
      1. 静态局部变量
      2. 静态外部变量(函数外部静态变量)
      3. 外部变量(可为其他文件引用)
   2. 内存中动态存储区：自动变量和形式参数
   3. CPU 中的寄存器：寄存器变量
5. **数组**

数组定义与使用、数组名作为函数参数的地址传递方式。

1. **数组定义与使用**
   1. 一维数组：类型标识符 数组名［常量表达式］：int a［10］;它表示数组名为a，此数组为整型，有10个元素。
      1. 数组名定名规则和变量名相同，遵循标识符定名规则。
      2. 用方括号括起来的常量表达式表示下标值：int a［2\*5］；
      3. 常量表达式的值表示元素的个数，即数组长度。
      4. 常量表达式中可以包括常量、常变量和符号常量，但不能包含变量。也就是说，C++不允许对数组的大小作动态定义，即数组的大小不依赖于程序运行过程中变量的值。二维数组
   2. 二维数组：类型标识符 数组名［常量表达式］［常量表达式］：float a［3］［4］；
   3. 一维数组元素的表示形式为：数组名［下标］；二维数组的元素的表示形式为：数组名 ［下标］［下标］，如a［2］［3］。
2. **数组名作为函数参数的地址传递方式**
   1. **说明**
      1. 如果函数实参是数组名，形参也应为数组名，形参不能声明为普通变量(如int array;)。
      2. 数组名代表数组首元素的地址，并不代表数组中的全部元素。因此用数组名作函数实参时,不是把实参数组的值传递给形参，而只是将实参数组首元素的地址传递给形参。
   2. **方法：**利用数组传递定义新函数：a[i]=a[k]
3. **指针**

指针概念、指针与数组、函数、结构体的关系，理解基本指针操作。

* 1. **指针概念**：如果在程序中定义了一个变量，在编译时就给这个变量分配内存单元。
     1. ＆取地址运算符。；\* 指针运算符（或称间接访问运算符）
     2. 如： &a为变量a的地址，\*p为指针变量p所指向的存储单元。
  2. **关系**
     1. 数组：数组元素的指针就是数组元素的地址。
     2. 函数：函数指针变量常见的用途之一是作为函数的参数，将函数名传给其他函数的形参。这样就可以在调用一个函数的过程中根据给定的不同实参调用不同的函数。

<https://blog.csdn.net/A3152005345/article/details/148909852?sharetype=blogdetail&sharerId=148909852&sharerefer=PC&sharesource=A3152005345&spm=1011.2480.3001.8118>

* + 1. 结构体：结构体是新定义的数据类型，和int,char,float这些关键字一样只是表示类型,不会占用内存空间,而结构体变量或者结构体数据才是实实在在创建出来的数据,有内存空间,我们可以创建指针指向结构体数据,但是不能指向结构体本身。结构体变量名和数组名不同,数组名本身就代表了数组的地址,而结构体变量名不是,所以指针指向结构体地址需要使用取地址符。

https://blog.csdn.net/A3152005345/article/details/148909993?spm=1001.2014.3001.5501

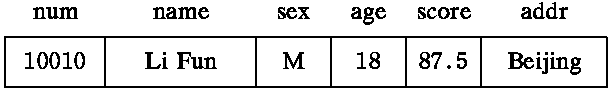
* 1. **基本操作：**

https://blog.csdn.net/A3152005345/article/details/148920554?spm=1001.2014.3001.5502

1. **自定义数据类型**

结构体、共用体、枚举类型的定义与使用。

* 1. **结构体**：有时需要将不同类型的数据组合成一个有机的整体，以供用户方便地使用。这些组合在一个整体中的数据是互相联系的。[struct语句]



* 1. **共用体**：有时需要使几种不同类型的变量存放到同一段内存单元中。例如，可把一个整型变量、一个字符型变量、一个双精度型变量放在同一个地址开始的内存单元中。不能引用共用体变量，而只能引用共用体变量中的成员。

union 共用体类型名

｛成员表列

｝；

* 1. **枚举类型**：如果一个变量只有几种可能的值，可以定义为枚举类型。所谓“枚举”是指将变量的值一一列举出来，变量的值只能在列举出来的值的范围内。
     1. 声明：enum 枚举类型名 {枚举常量表列};
     2. https://blog.csdn.net/A3152005345/article/details/148919835?sharetype=blogdetail&sharerId=148919835&sharerefer=PC&sharesource=A3152005345&spm=1011.2480.3001.8118

1. **类和对象**

类的基本属性（公有、私有、保护）与继承方式。

* 1. **类的三大特性**：封装、继承、多态
  2. **继承机制：**

https://blog.csdn.net/A3152005345/article/details/148922121?sharetype=blogdetail&sharerId=148922121&sharerefer=PC&sharesource=A3152005345&spm=1011.2480.3001.8118

1. **使用类和对象**

构造函数与析构函数的功能与调用顺序。

https://blog.csdn.net/A3152005345/article/details/148922640?spm=1001.2014.3001.5501

1. **运算符重载（大题）**

运算符重载为重点，特别是大题考点。

1. **输入输出流\*此为第13章内容**

输入输出流的基本概念。

* 1. **流的概念**：在C++中，将数据从一个对象到另一个对象的流动抽象为“流”。流动的方向不同，构成输入/输出流，即I/O流。
  2. **标准输出流**：是流向标准输出设备(显示器)的数据。ostream类定义了3个输出流对象（cout,cerr,clog）
     1. cout流对象：是console output的缩写，意为在控制台的输出。
     2. cerr流对象：是标准错误流。其作用是向标准错误设备输出有关出错信息。
     3. clog流对象：也是标准错误流，它是console log的缩写。作用和cerr相同，都是在终端显示器上显示出错信息。区别：cerr是不经过缓冲区，直接向显示器上输出有关信息，而clog中的信息存放在缓冲区中，缓冲区满后或遇endl时才向显示器输出。
  3. **标准输入流：**是从标准输入设备(键盘)流向程序的数据。istream类定义了4个输入流对象（cin,cout,cerr,clog）
     1. eof函数：eof是end of file的缩写，表示“文件结束”。从输入流读取数据，如果到达文件末尾(遇文件结束符)，eof函数值为非零值(表示真)，否则为0(假)。
     2. peek函数：peek是“观察”的意思，peek函数的作用是观测下一个字符。其调用形式为c=cin.peek( );cin.peek

函数的返回值是指针指向的当前字符，但它只是观测，指针仍停留在当前位置，并不后移。如果要访问的字符是文件结束符，则函数值是EOF(-1)。

* + 1. putback函数：其调用形式为cin.putback(ch);

其作用是将前面用get或getline函数从输入流中读取的字符ch返回到输入流，插入到当前指针位置，以供后面读取。

* + 1. ignore函数：其调用形式为cin,ignore(n, 终止字符)

函数作用是跳过输入流中n个字符，或在遇到指定的终止字符时提前结束(此时跳过包括终止字符在内的若干字符)。

1. **标准模板库\*教材暂无**

模板与标准库的简单了解。

https://blog.csdn.net/A3152005345/article/details/148922944?spm=1001.2014.3001.5501