

湖南省高等教育自学考试
课程考试大纲

可视化程序设计
(课程代码: 08762)

湖南省教育考试院组编
2016年12月

高等教育自学考试课程考试大纲

课程名称：可视化程序设计

课程代码：08762

第一部分 课程性质与目标

一、课程性质与特点

可视化程序设计是高等教育自学考试游戏软件开发技术（本科）专业的专业核心课程。通过本课程的学习，考生能够掌握可视化程序设计的相关知识技能。具体包括：C++语言程序设计基础，Windows 程序设计基础，MFC 程序设计基础，以及若干可视化程序的深入内容。本课程是一门技术性非常强的课程，非常强调考生的实践能力培养，要求考生必须在课后进行大量的实践练习。

二、课程目标与基本要求

通过本课程的学习，考生首先需要掌握 C++语言程序设计基础，具体包括：C++语言的词法，如常量、变量、标识符、关键字、运算符等内容；C++的基本数据类型，如 int、float、double、char 等；C++语言的语法，如赋值语句、选择语句、循环语句、函数等；C++语言的编译预处理内容。其次考生需要掌握有关面向对象部分的内容，具体包括类、封装、继承、多态、重载、模板等。再次，考生需要掌握 Windows 程序设计的有关内容。这其中首先要求掌握 Windows 程序设计中的文档/视图结构，然后需要掌握对话框和标准控件的使用，再次需要重点需要掌握 MFC 的原理与方法，能够使用 MFC 进行 Windows 程序设计，最后需要了解比较高级的 Windows 程序设计技术如图形绘制、非模态对话框编程、数据库编程、异常处理、动态链接库、多线程等。

三、与本专业其他课程的关系

可视化程序设计是一门极为重要的专业必修课程，在游戏软件开发技术专业中占有重要地位。本课程的后续课程为“DirectX”、“游戏开发流程与引擎原理”。

第二部分 考核内容与考核目标

第一章 Visual C++集成开发环境

一、学习目的与要求

通过本章的学习，考生需要了解 Visual C++集成开发环境的各组成部分及其能够为用户提供的功能。考生需要掌握 Visual C++集成开发环境的初步用法，能够熟练地使用常见的菜单、工具等指令，能够在 Visual C++中建立软件开发项目。

二、考核知识点与考核目标

（一）Visual C++概述（次重点）

识记：Visual C++是 Microsoft 公司的一个可视化集成开发环境，其他开发环境：Borland 的 C++ Builder，IBM 的 VisualAge C++。MSDN：

Microsoft developer network library

应用：安装 Visual C++

(二) 项目和项目工作区（次重点）

识记：项目文件 DSP，项目工作区文件 DSW，视图种类：类视图，资源视图，文件视图

理解：各类视图的用途

应用：使用 Visual C++ 建立一个项目

(三) 编辑器（一般）

识记：编辑器的种类；资源编辑器包含类型

(四) 菜单栏和工具栏（一般）

识记：菜单栏：File 菜单，Edit 菜单，View 菜单，Insert 菜单，Project 菜单，Build 菜单，Tools 菜单，Window 菜单，Help 菜单。工具栏：Standard 工具栏，Build MiniBar 工具栏，WizardBar 工具栏

第二章 C++语言基础

一、学习目的与要求

通过本章的学习，考生需要了解 C++ 语言基本的语法基础。考生首先需要了解 C++ 语言中的常用关键字，其次需要掌握标识符的命名规则，再次需要掌握 C++ 中各种运算符以及表达式的使用。在此基础上，考生需要掌握 C++ 中的流程控制语句，包括选择语句、循环语句和转移语句。然后需要掌握 C++ 函数的使用和编写方法。最后，还需要了解一些 C++ 比较深入的内容，例如变量的作用域和生存周期，变量和函数存储类型，编译预处理指令等。

二、考核知识点与考核目标

(一) C++ 概述（一般）

识记：C++ 对 C 的主要扩充：C++ 的函数原型不能省略；C++ 的变量声明语句可以出现在任何地方；C++ 使用 new 和 delete；标准 C++ 提供字符串类；C++ 一般使用输入输出流来进行输入输出

理解：C++ 的特点：丰富的运算符和数据类型，结构化的程序设计方法，高效的机器代码，良好的可移植性

(二) C++ 程序基本要素（重点）

识记：C++ 主要关键字：auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete, do, double, dynamic_cast, else, enum, explicit, extern, false, float, for, friend, goto, if, inline, int, long, namespace, new, operator, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template, this, throw, true, try, typedef, typename, union, unsigned, using, virtual, void, volatile, while。
C++ 基本运算符及其优先级和结合性

理解：标识符的命名规则：由字母、数字和下划线组成，以字母或下划线开头。常量和变量的区别

应用：给定数学表达式，转换为合法的 C++ 表达式

（三）C++数据类型（重点）

识记：数据类型：基本数据类型，指针类型，构造类型。基本数据类型：
`char, int, float, double, void, short, long, bool`

理解：`const` 的使用方法。数组的定义方法。指针的使用方法。字符串的结束标记。结构体的使用方法。枚举的使用方法。`typedef` 类型定义的使用方法

（四）控制语句（重点）

理解：选择语句。循环语句。转移语句

（五）函数（重点）

识记：函数的定义格式：函数类型，函数名，存储类型

理解：函数调用。函数的默认参数值。内联函数的使用。引用的概念和特色

应用：给定简单任务，能够编写 C++ 函数

（六）作用域与存储类型（次重点）

识记：作用域的概念。生存期的概念。局部变量的概念。全局变量的概念

理解：局部变量和全局变量的区别。C++ 内存分配方式：自动分配，静态分配，动态分配。C++ 变量的存储类型：`auto, register, extern, static`，作用域限定符的使用。函数的存储类型：内部函数，外部函数。`new` 的使用方法

（七）编译预处理（次重点）

识记：`#include` 的 2 种形式：尖括号，双引号

理解：常见编译预处理指令的作用：`#define, #include, #if`

第三章 类和对象

一、学习目的与要求

通过本章的学习，考生需要掌握面向对象程序设计思想的基本概念，具备基本的面向对象程序设计的能力。具体而言，考生需要理解面向对象程序设计的四个基本特征的概念及其涵义。考生需要掌握 C++ 中类定义的基本语法格式，需要详细了解关于访问权限的规则和使用方法。考生还需要理解和掌握 C++ 类中有关静态的概念，包括静态数据成员和静态成员函数。进一步的，考生需要掌握组合类的设计方法与友元的使用方法。最后，考生必须掌握 `const` 的使用时机，必须能够正确的使用常数据成员和常成员函数。

二、考核知识点与考核目标

（一）面向对象程序设计方法及特征（重点）

识记：4 个特征：抽象，封装，继承，多态

理解：结构化程序设计。面向对象程序设计

(二) C++类 (重点)

识记：类定义的基本形式。数据成员。成员函数。访问权限：`private`，`protected`，`public`

理解：构造函数。拷贝构造函数。析构函数。`this` 指针

应用：编写程序，定义 `Time` 类

(三) 静态成员 (重点)

识记：`static` 关键字。静态成员通过作用域限定符 (`::`) 进行访问

理解：类属性和实例属性的区别。静态数据成员。静态成员函数。静态成员函数只能访问静态数据成员，而不能访问非静态数据成员

应用：编写程序，定义一个含有静态数据成员计数器的类

(四) 组合类 (重点)

识记：组合类的定义

理解：组合类的拷贝构造函数

应用：编写程序，实现一个组合类

(五) 常对象和常对象成员 (重点)

识记：使用 `const` 修饰对象使之成为常对象。使用 `const` 修饰成员函数使之成为常成员函数。使用 `const` 修饰数据成员使之成为常数据成员

理解：常对象的属性值不能被修改。常成员函数只能访问数据成员，不能设置或者修改数据成员。常成员函数只能调用常成员函数，不能调用其他普通成员函数。而普通成员函数可以调用常成员函数。常数据成员的初始值只能在构造函数获取，且只能在初始化列表中完成。静态常数据成员的初始化只能在类外进行

应用：编写程序，定义具有常成员函数的类

(六) 友元 (一般)

识记：友元函数。友元类

理解：访问权限控制符不影响友元声明。友元是单方向的，不具备交换性和传递性

应用：编写程序，为指定类实现友元函数

第四章 继承与多态

一、学习目的与要求

通过本章的学习，考生需要深入理解面向对象程序设计思想中比较重要的有关继承和多态的概念。首先，考生必须掌握继承的基本语法格式，包括三种继承方式的实现。其次，考生需要掌握虚函数和纯虚函数的实现方式。考生需要重点理解函数重载机制以及运算符重载机制，具备运算符重载的实现能力。再次，考生需要了解有关模板函数和模板类的初步知识。最后，考生需要掌握 `Microsoft Visual C++` 相关的语法扩充以及相应的编程规范，这是使用 `Visual C++` 开发

Windows 程序的必备。

二、考核知识点与考核目标

(一) Microsoft Visual C++的语法扩充 (重点)

识记：扩充定义的数据类型：FAR, NEAR, CONST, BOOL, UINT, BYTE, WORD, DWORD, SHORT, LONG, LONGLONG, FLOAT, CHAR, VOID, BSTR, LPCSTR, LPSTR, LPCTSTR, LPTSTR, LPVOID, LPARAM, LPRESULT, LPCRECT, PROC, WNDPROC, WPARAM, HANDLE, COLORREF。Visual C++运行库：libc.lib, libcmnt.lib, msvcr.lib, libcp.lib, libcpmt.lib, msvcprt.lib, libci.lib, libcimt.lib, msvcirt.lib。匈牙利命名法：前缀 a 表示数组变量，前缀 b 表示布尔变量，前缀 c 表示字符变量，前缀 n、i 表示整型变量，前缀 x、y 表示坐标，前缀 sz 表示以 0 结束的字符串变量，前缀 p 表示指针变量，前缀 lp 表示长指针变量，前缀 h 表示句柄，前缀 fn 表示函数，前缀 m_ 表示类的成员变量，前缀 C 表示类和结构，前缀 Afx、afx、AFX 表示应用程序框架，前缀 ID*_ 表示资源标识

理解：动态链接库和静态链接库。运行时类型识别：run-time type information, RTTI。RTTI 的 2 种方法：dynamic_cast, typeid

(二) 继承与派生 (重点)

识记：派生类定义的语法格式。多继承。虚基类

理解：3 种派生方式：public, private, protected。派生类的构造函数和析构函数。派生类显式方式构造函数。派生类隐式调用基类构造函数。虚基类的作用

应用：编写程序，继承指定基类

(三) 多态和虚函数 (次重点)

识记：虚函数的关键：基类的指针指向派生类的对象。派生类的指针不能指向基类的对象。纯虚函数的定义形式。抽象类。CObject 是 MFC 的抽象基类。

理解：多态的具体表现：一种接口，多种方法。指向派生类对象的基类指针，只能访问派生类从基类中继承而来的公有成员。虚函数。虚析构函数。抽象类不能用来声明对象，只能作为基类使用

应用：编写函数，定义具有虚函数的基类和派生类

(四) 重载 (次重点)

识记：函数重载的机制：函数参数类型不同，函数名相同。运算符重载的格式

理解：函数返回类型不能作为函数重载的依据。运算符重载不能创造新的运算符，只能重载 C++ 已经存在而且允许重载的运算符。运算符重载不能改变原运算符的优先级和结合性。运算符重载不能改变原运算符的参数个数。对于“++”和“--”运算符的前缀、后缀形式，对其

进行运算符重载的格式

应用：为 **Complex** 类编写运算符重载函数

(五) 模板（一般）

识记：函数模板。类模板

第五章 创建应用程序框架

一、学习目的与要求

通过本章的学习，考生需要从全局上了解 **Visual C++** 应用程序框架的构成和创建。考生需要掌握利用应用程序创建向导创建不同类型的应用程序的方法，需要详细了解应用程序向导创建的各类文件。此外，考生需要能够在程序中利用类向导为项目创建新的类。最后，考生需要掌握程序调试的方法。

二、考核知识点与考核目标

(一) 应用程序向导（次重点）

识记：向导类型

应用：利用应用程序向导，创建 **Win32** 程序项目

(二) 应用程序向导生成文件（次重点）

识记：文件类型

理解：头文件。其他文件

(三) **ClassWizard** 类向导（一般）

识记：类向导的功能

理解：添加消息处理函数

(四) 程序调试（一般）

识记：语法错误：错误和警告

第六章 文档与视图

一、学习目的与要求

通过本章的学习，考生需要牢固树立 **Windows** 程序设计中的文档/视图观念，必须要熟练掌握文档/视图的结构构成及其相互作用。考生还需要对菜单设计、工具栏设计和状态栏设计有一定的了解。进一步，考生需要熟练掌握鼠标消息的处理流程和处理方法，考生还需要掌握序列化机制及其实现方法。最后，考生需要对滚动视图和多视图有一定的了解。

二、考核知识点与考核目标

(一) 文档视图结构（重点）

识记：文档和视图是 **MFC** 应用程序的核心。文档是 **MFC** 应用程序处理数据的集合，包括文本、图形、图像和表格。视图是文档在应用程序窗口的一个映像。单文档和多文档

理解：文档与视图的相互联系。一个视图只能拥有一个文档，一个文档可

以拥有多个视图。文档与视图的相互作用的主要函数：
GetDocument, UpdateAllViews, OnUpdate

应用：编写程序，创建一个单文档项目

(二) 鼠标消息处理 (重点)

识记：鼠标消息：WM_MOUSEMOVE, WM_LBUTTONDOWN,
WM_LBUTTONUP, WM_RBUTTONDOWN, WM_RBUTTONUP,
WM_LBUTTONDBLCLK

理解：客户区和非客户区

(三) 文档的读写 (次重点)

识记：读写的 3 种方法。序列化的概念

理解：CFile 类的打开方式。OnFileSave 函数完成的工作：文档对象获取 CFile 对象指针创建 CArchive 对象；文档对象调用 Serialize 函数；Serialize 函数利用 CArchive 对象把序列化数据存储到关联文件。序列化只能访问顺序文件，不能访问随机存取文件；只能处理二进制文件，不能处理文本文件。自定义类实现序列化的 4 个条件

应用：编写程序，为指定类实现序列化

(四) 菜单设计 (次重点)

识记：菜单资源

理解：固定菜单。弹出式菜单

(五) 滚动视图和多视图 (一般)

识记：滚动视图类：CScrollView

理解：多视图

(六) 工具栏和状态栏设计 (一般)

识记：工具栏。状态栏

第七章 对话框和标准控件

一、学习目的与要求

通过本章的学习，考生需要掌握 Windows 程序设计中有有关对话框和标准控件的概念及其使用方法。考生需要掌握在程序中创建对话框的方法，还需要掌握在对话框中进行数据交换和数据验证的方法。考生应该要熟练掌握对话框中各种常用的成员函数的使用，能够正确的设置函数参数、把握调用时机。最后，考生需要掌握 Windows 程序中提供的各种标准控件，至少要熟练掌握所有控件都具有的公共属性。

二、考核知识点与考核目标

(一) 对话框概述 (重点)

识记：对话框类：CDialog。信息对话框的 3 个函数：AfxMessageBox, MessageBox, CWnd::MessageBox。信息对话框按钮类型：MB_ABORTRETRYIGNORE, MB_OK, MB_OKCANCEL,

MB_RETRYCANCEL, MB_YESNO, MB_YESNOCANCEL

理解: 对话框常用成员函数。对话框数据交换 DDX。对话框数据验证 DDV

应用: 编写程序, 创建指定的对话框

(二) 使用对话框 (重点)

理解: 对话框流程: 创建对话框, 添加控件及关联的成员变量

(三) 标准控件 (重点)

识记: 常用的 MFC 控件类。列表框的 4 种类型。组合框的 3 种形式: 简单组合框, 下拉组合框, 下拉列表框。滚动条的 2 种类型: 垂直滚动条, 水平滚动条

理解: 控件共有属性: ID, Caption, Visible, Group, HelpID, Disabled, Tab Stop。按钮的 4 种类型。单选按钮和复选框的区别

第八章 MFC 原理与方法

一、学习目的与要求

通过本章的学习, 考生需要掌握 MFC 的基本原理和相应的使用方法。考生首先需要理解 Windows 程序设计的基本特点, 然后需要了解 Windows 的消息类型、Windows 的常用库以及 Windows 程序设计中的常用句柄。考生需要对 MFC 体系结构有一个全局性的认识, 对于常用的类、宏和全局函数均有良好的掌握。

二、考核知识点与考核目标

(一) Windows 编程基础 (重点)

识记: Windows 是一个基于事件驱动、消息响应的操作系统。Windows API: application programming interface。3 个主要的链接库: User32.dll, Gdi32.dll, Kernel32.dll。Windows SDK: software development kit。Windows 程序的 3 个部分。WinMain。消息: 用于描述某个事件发生的信息。事件: 用户操作应用程序产生的动作或 Windows 系统自身产生的动作。消息类型: 窗口消息, 命令消息, 控件通知消息。窗口消息: WM_PAINT, WM_CLOSE, WM_MOVE, WM_KEYDOWN, WM_MOUSEMOVE。命令消息: WM_COMMAND

理解: Kernel32.dll: 内核库, 定义了实现操作系统核心底层功能的函数。User32.dll: 用户界面库, 定义了窗口管理函数。Gdi32.dll: 图形设备接口库, 定义了图形设备 GDI 函数。MSG 结构: hWnd, message, wParam, lParam, time, pt。句柄: 窗口句柄 HWND, 运行实例句柄 HINSTANCE, 光标句柄 HCURSOR, 图标句柄 HICON, 菜单句柄 HMENU, 文件句柄 HFILE, 设备环境句柄 HDC, 字体句柄 HFONT, 画笔句柄 HPEN, 位图句柄 HBITMAP, 画刷句柄 HBRUSH, 调色板句柄 HPALETTE

(二) 微软基础类 (重点)

识记: MFC 体系结构: 类, 宏, 全局函数。全局函数以 Afx 为前缀, 全局变量以 afx 为前缀

理解: MFC 类: 根类, 应用结构类, 窗口支持类, 菜单类, 设备描述表和绘图类, 数据类型类, 文件和数据库类, 调试和异常处理类, Internet 和网络支持类, OLE 类。宏功能: 消息映射, 运行时对象类型识别, 诊断服务, 异常处理

(三) MFC 应用程序框架 (重点)

识记: 应用程序对象 theApp 是唯一的全局变量, 主要功能是通过调用主函数启动程序的运行

理解: theApp 是 CWinApp 派生类的对象, theApp 的销毁就表示运行程序实例的消亡

(四) MFC 消息管理 (重点)

识记: MFC 消息管理: 发送, 处理。消息值分为 4 段: Windows 系统消息, 自定义窗消息, 保留值, 字符串消息

理解: 消息映射机制。消息名和消息处理函数名之间的对应关系

(五) MFC 宏 (重点)

识记: 常用 MFC 宏: RUNTIME_CLASS, DECLARE_DYNAMIC, IMPLEMENT_DYNAMIC, DECLARE_DYNCREATE, IMPLEMENT_DYNCREATE, DECLARE_SERIAL, IMPLEMENT_SERIAL, TRACE, ASSERT, VERIFY, DELARE_MESSAGE_MAP, BEGIN_MESSAGE_MAP, END_MESSAGE_MAP, ON_COMMAND, ON_MESSAGE

理解: 运行时基础宏。动态支持宏。动态创建宏。序列化宏。跟踪声明宏。断言宏。断言有效宏

(六) 常用的 MFC 类 (重点)

识记: CObject 类。CCmdTarget 类。CWinApp 类。CWnd 类。CFrameWnd 类。CView 类。CDocument 类

理解: MFC 诊断。运行时访问类的信息

第九章 图形绘制

一、学习目的与要求

通过本章的学习, 考生需要了解图形绘制有关的内容, 掌握基本的图形绘制方法。首先, 考生需要掌握 GDI 的概念, 需要了解 DC 的定义和使用方法, 还需要掌握 Windows 程序设计中颜色的表示方法。其次, 考生需要具体掌握画笔和画刷的用法, 能够在程序设计中应用画笔和画刷。再次, 考生需要掌握文本与字体的相关内容。最后, 考生需要对位图、图标和光标有一定的了解。

二、考核知识点与考核目标

(一) 图形设备接口 (重点)

识记: 图形设备接口: graphics device interface, GDI。设备环境: device context, DC。屏幕坐标系。窗口坐标系。客户区坐标系。COLORREF

理解: MFC 的 DC 类: CDC, CPaintDC, CClientDC, CWindowDC, CMetaFileDC。黑色: RGB(0, 0, 0)。红色: RGB(255, 0, 0)。绿色: RGB(0, 255, 0)。蓝色: RGB(0, 0, 255)

(二) 画笔和画刷 (次重点)

识记: GDI 对象: 画笔, 画刷, 字体, 位图, 调色板。基本绘图成员函数: SetPixel, MoveTo, LineTo, Polyline, PolyBezier, Rectangel, RoundRect, Ellipse, ArcTo, Pie, Chord, Polygon, DrawIcon

理解: 画笔基本样式: PS_SOLID, PS_DOT, PS_DASH, PS_DASHDOT, PS_DASHDOTDOT, PS_NULL, PS_INSIDEFRAME。画刷的阴影样式: HS_CROSS, HS_DIAGCROSS, HS_HORIZONTAL, HS_VERTICAL, HS_BDIAGONAL, HS_FDIAGONAL。GDI 堆对象: BLACK_PEN, NULL_PEN, WHITE_BRUSH, GRAY_BRUSH, LTGRAY_BRUSH, WHITE_PEN, BLACK_BRUSH, NULL_BRUSH, DKGRAY_BRUSH, HOLLOW_BRUSH

(三) 文本与字体 (一般)

识记: Windows 提供的 6 种堆字体对象: ANSI_FIXED_FONT, ANSI_VAR_FONT, SYSTEM_FONT, SYSTEM_FIXED_FONT, DEVICE_DEFAULT_FONT, OEM_FIXED_FONT。

理解: Windows 支持的 3 类字体: 光栅字体, 矢量字体, TrueType 字体。

(四) 位图、图标与光标 (一般)

识记: 位图的概念。位图的种类: 2 色, 4 色, 16 色, 256 色, 增强色, 真彩色。图标的尺寸: 16×16, 32×32, 48×48。光标的尺寸: 32×32。

理解: 位图的类型: 设备相关位图, 设备无关位图。图标与位图的关系。光标与位图的关系。系统光标: IDC_ARROW, IDC_IBEAM, IDC_WAIT, IDC_CROSS, IDC_UPARROW, IDC_SIZE, IDC_SIEALL, IDC_ICON, IDC_SIZENWSE, IDC_SIZENESW, IDC_SIZENS, IDC_SIZEWE。 ,

第十章 对话框高级编程

一、学习目的与要求

通过本章的学习, 考生需要掌握非模态对话框和模态对话框的区别, 需要深入理解非模态对话框的工作流程。考生还需要掌握通用对话框的使用方法, 能够在程序中根据需要使用合适的通用对话框。最后, 考生需要了解有关公共控件的初步知识。

二、考核知识点与考核目标

(一) 通用对话框 (重点)

识记: 通用对话框: CColorDialog, CFileDialog, CFindReplaceDialog, CFontDia

理解: 通用对话框的使用步骤: 声明一个通用对话框类的对象, 修改相应的数据成员, CPagesSetupDialog, CPrintDialog, COleDialog

(二) 非模态对话框 (次重点)

理解: 模态对话框和非模态对话框的区别。

应用: 编写程序, 利用通用对话框访问文件。

(三) 公共控件 (一般)

识记: 旋转按钮。滑块。进度条。标签。属性表。图像列表。列表视控件。树视控件。

第十一章 Visual C++编程深入

一、学习目的与要求

通过本章的学习, 考生需要掌握 ODBC 数据库应用程序的开发方法, 需要掌握 Windows 程序设计中异常处理的基本手段, 需要了解有关动态链接库的有关知识, 最后需要对多线程程序开发有一定的了解。

二、考核知识点与考核目标

(一) ODBC 数据库应用 (次重点)

识记: ODBC: 开放式数据库连接, open database connectivity。MFC ODBC: CDatabase, CRecordset, CRecordView, CFieldExchange, CDbException

理解: ODBC 体系结构: 应用程序, 驱动程序管理器, ODBC 驱动程序, 数据源。

应用: 编写程序, 访问指定数据库

(二) 异常处理 (一般)

识记: 常见异常: EXCEPTION_ACCESS_VIOLATION, EXCEPTION_FLT_DIVIDE_BY_ZERO, EXCEPTION_FLT_INVALID_OPERATION, EXCEPTION_FLT_OVERFLOW, EXCEPTION_INT_DIVIDE_BY_ZERO, EXCEPTION_INT_OVERFLOW

理解: MFC 异常类: CException, CMemoryException, CFileException, CArchiveException, CResourceException, CUserException, CNotSupportedException, COleException, CDbException

(三) 动态链接库 (一般)

识记: 动态链接库: dynamics link library, DLL

理解: 隐式链接和显式连接

(四) 多线程（一般）

识记：工作线程。用户界面线程

第三部分 有关说明与实施要求

一、考核的能力层次表述

本大纲在考核目标中，按照“识记”、“理解”、“应用”三个能力层次规定其应达到的能力层次要求。各能力层次为递进等级关系，后者必须建立在前者的基础上，其含义是：

识记：能知道有关的名词、概念、知识的含义，并能正确认识和表述，是低层次的要求。

理解：在识记的基础上，能全面把握基本概念、基本原理、基本方法，能掌握有关的概念、原理、方法的区别与联系，是较高层次的要求。

应用：在理解的基础上，能运用基本概念、基本原理、基本方法联系学过的多个知识点分析和解决有关的理论问题和实际问题，是最高层次的要求。

二、教材

指定教材：Visual C++面向对象编程教程，王育坚，清华大学出版社，2013年第3版

三、自学方法指导

1. 在开始阅读指定教材某一章之前，先翻阅大纲中有关这一章的考核知识点及对知识点的能力层次要求和考核目标，以便在阅读教材时做到心中有数，有的放矢。
2. 阅读教材时，要逐段细读，逐句推敲，集中精力，吃透每一个知识点，对基本概念必须深刻理解，对基本理论必须彻底弄清，对基本方法必须牢固掌握。
3. 在自学过程中，既要思考问题，也要做好阅读笔记，把教材中的基本概念、原理、方法等加以整理，这可从中加深对问题的认知、理解和记忆，以利于突出重点，并涵盖整个内容，可以不断提高自学能力。
4. 完成书后作业和适当的辅导练习是理解、消化和巩固所学知识，培养分析问题、解决问题及提高能力的重要环节。在做练习之前，应认真阅读教材，按考核目标所要求的不同层次，掌握教材内容。在练习过程中对所学知识进行合理的回顾与发挥，注重理论联系实际和具体问题具体分析。解题时应注意培养逻辑性，针对问题围绕相关知识点进行层次（步骤）分明的论述或推导，明确各层次（步骤）间的逻辑关系。

四、对社会助学的要求

1. 应熟知考试大纲对课程提出的总要求和各章的知识点。
2. 应掌握各知识点要求达到的能力层次,并深刻理解对各知识点的考核目标。
3. 辅导时,应以考试大纲为依据,指定的教材为基础,不要随意增删内容,以免与大纲脱节。
4. 辅导时,应对学习方法进行指导,宜提倡“认真阅读教材,刻苦钻研教材,主动争取帮助,依靠自己学通”的方法。
5. 辅导时,应注意突出重点,对考生提出的问题,不要有问即答,要积极启发引导。
6. 注意对考生能力的培养,特别是自学能力的培养,要引导考生逐步学会独立学习,在自学过程中善于提出问题、分析问题,做出判断,解决问题。
7. 要使考生了解试题的难易与能力层次高低两者不完全是一回事,在各个能力层次中会存在着不同难度的试题。
8. 助学学时:本课程共 4 学分,建议总课时 72 学时,其中助学课时分配如下:(每学分 18 学时)

章次	内容	学时
第 1 章	Visual C++集成开发环境	4
第 2 章	C++语言基础	8
第 3 章	类和对象	8
第 4 章	继承与多态	8
第 5 章	创建应用程序框架	8
第 6 章	文档与视图	8
第 7 章	对话框和标准控件	8
第 8 章	MFC 原理与方法	8
第 9 章	图形绘制	4
第 10 章	对话框高级编程	4
第 11 章	Visual C++编程深入	4
合计		72

五、关于命题考试的若干规定

1. 本大纲各章所提到的内容和考核目标都是考试内容。试题覆盖到章,适当突出重点。
2. 试卷中对不同能力层次的试题比例大致是:“识记”为 30%、“理解”为 40%、“应用”为 30%。
3. 试题难易程度应合理:易、较易、较难、难比例为 2:3:3:2。
4. 每份试卷中,各类考核点所占比例约为:重点占 60%,次重点占 30%,一般占 10%。

5. 试题类型一般为：单项选择题、填空题、名词解释题、简答题、设计题。
6. 考试采用闭卷考试，考试时间 150 分钟，采用百分制评分，60 分合格。

六、题型示例（样题）

一、单项选择题（本大题共 i_0 小题，每小题 j_0 分，共 $i_0 j_0$ 分）

在每小题列出的四个备选项中只有一个是符合题目要求的，请将其选出并将“答题卡”上的相应字母涂黑。错涂、多涂或未涂均无分。

1. 下列选项中，属于 C++ 基本数据类型的是

A. int B. integer C. Int D. Integer

二、填空题（本大题共 i_0 小题，每小题 j_0 分，共 $i_0 j_0$ 分）

1. 文档和_____完成了 Windows 程序的大部分功能，是 MFC 应用程序的核心。

三、名词解释题（本大题共 i_0 小题，每小题 j_0 分，共 $i_0 j_0$ 分）

1. 动态内存分配

四、简答题（本大题共 i_0 小题，每小题 j_0 分，共 $i_0 j_0$ 分）

1. Visual Studio 应用程序向导能够生成 6 种类型的头文件，简述其中 3 种。

五、设计题（本大题共 i_0 小题，每小题 j_0 分，共 $i_0 j_0$ 分）

1. 如下是一个 Complex 类的定义，请为其设计加号运算符重载。加号运算符重载的声明已经给出，请不要改变 Complex 类本身的定义。

```
class Complex
public:
    double real;
    double image;
    Complex operator + (Complex lhs,Complex rhs);
```