

《08075 计算机高级程序设计》

实践考核大纲

一、课程性质与目标

(一) 课程性质和特点

“计算机高级程序设计”实践课程是一门专业实践课程，聚焦 Python 语言程序设计的实际应用能力培养。课程紧密结合企业开发技术要求，衔接理论课程中 Python 基础语法、常见库的使用、组合数据类型的使用、函数的设计和使用的、文件的读取和写入、面向对象编程等核心知识，通过实操训练，助力考生将理论转化为解决实际问题的能力，适配计算机程序开发相关岗位的实践需求。

(二) 课程目标

通过本课程的学习，使得学生能够理解 Python 的编程模式，验证、理解直至熟练运用课堂所学知识，熟练使用 IDLE、anaconda 或其他 Python 开发环境，熟练运用 Python 列表、元组、字典、集合等基本数据类型以及相关列表推导式、切片等特性来解决实际问题，熟练掌握 Python 分支结构、循环结构、函数设计以及类的设计与使用，熟练使用 Python 读写文本文件与二进制文件，了解 Python 程序的调试方法，熟练运用 Python 编写面向对象程序，同时使学生了解不同领域的 Python 扩展模块基本用法，同时还应培养学生的代码优化与安全编程意识，培养学生的创新意识、人文素养、科学思维、严谨求实和工匠精神。

(三) 课程的重点

本课程的重点内容包括：重点章节有第 3 章“程序控制结构”、第 4 章“组合数据类型”、第 5 章“函数”、第 8 章“Python 面向对象编程”。

二、考核内容和考核目标

第 1 章 Python 概述

一、学习目的与要求

- (1) 熟练安装配置 Python 开发环境，掌握 IDLE 的基本操作，能独立编写、运行简单 Python 程序。
- (2) 学会标准库导入与使用，为数据结构操作奠基。
- (3) 学会规范化的 pip 安装流程和扩展库合法性验证，培养安全编程意识。
- (4) 编程规范（缩进/注释）和测试驱动开发理念，强化工匠精神。

二、课程内容

- (1) 程序；
- (2) Python 简介、环境的安装与使用；（增加如何安装 anaconda）
- (3) Python 语言的基本语法规则；
- (4) Python 模块。

三、考核知识点及要求

- (1) 程序
识记：程序设计语言分类，程序的执行方式。

(2) Python 简介、环境的安装与使用

领会：Python 开发环境的安装配置方法。

(3) Python 语言的基本语法规则

应用：简单的 python 程序分析，IDLE 编写、运行及调试程序，正确运用注释、标识符、强制缩进等语法规则。

(4) Python 模块

领会：扩展库的安装方法。

识记：模块的导入和使用有哪些方式。

第 2 章 Python 基本数据类型、表达式和内置函数

一、学习目的与要求

(1) 通过数字、字符串、列表等内置对象的实践操作，培养基础数据处理能力。

(2) 运算符与内置函数的综合应用，奠定业务逻辑实现基础。

(3) 变量命名规范强化代码可读性意识，培养严谨求实的科学思维。

二、课程内容

(1) Python 数据类型；

(2) Python 运算符与表达式；

(3) Python 常用内置函数；

三、考核知识点及要求

(1) Python 数据类型

识记：整型、浮点型、布尔型等数据类型，变量和常量的区别，变量命名规范。

应用：数字、字符串、列表、元组、字典、集合的简单使用。

(2) Python 运算符与表达式

领会：Python 常见的运算符的使用；运算法的优先级。

综合应用：结合算术运算符、逻辑运算符编写正确的表达式完成计算任务。

(3) Python 常用内置函数

综合应用：常用 Python 内置函数，基本输入输出。迭代器函数的运用。

第 3 章 程序控制结构

一、学习目的与要求

(1) 循环优化技术提升大数据处理效率。

(2) 选择结构与循环结构实现业务逻辑。从分支选择算法流程引导学生做一个凡事有条理的人。

(3) 边界条件处理培养严谨思维，提高调试纠错能力，教导学生如何正确看待自己的错误？

二、课程内容

(1) 控制结构概述；

(2) 选择结构；

- (3) 循环控制结构；
 - (4) 异常处理结构。
- 三、考核知识点及要求

(1) 控制结构概述

识记：流程图的分类。

领会：算法如何语言表示。

(2) 选择结构

综合应用：单分支选择结构、双分支选择结构、多分支选择结构、选择结构的嵌套。

(3) 循环控制结构

综合应用：for 循环、while 循环，range 对象在循环中的使用，成员测试符 in 在循环语句中的使用，带有 else 子句的循环结构。利用选择结构和循环结构，编写解决实际问题的程序（如数值计算、简单逻辑判断等）。

(4) 异常处理结构

应用：利用 try..except...else..finally 语句，处理程序中的常见异常（如类型错误、索引错误等）。

第 4 章 组合数据类型

一、学习目的与要求

(1) 熟练掌握列表、元组、集合、字典的创建、访问、切片及常用操作方法，能运用组合数据类型存储和处理批量数据，实现数据高效处理（如用户 ID 去重）。

(2) 列表尾部操作，规范培养性能意识。

二、课程内容

(1) 组合数据类型概述；

(2) 列表类型；

(3) 元组类型；

(4) 集合类型；

(5) 字典类型。

三、考核知识点及要求

(1) 组合数据类型概述；

识记：组合数据类型的分类。

(2) 列表类型；

领会：列表的双向索引，成员资格判断运算符 in。

应用：列表对象的创建，列表元素访问与计数，切片的应用，列表元素的增加与删除，列表的常用操作。

(3) 元组类型；

领会：其他类型到元组的转换。

应用：元组的创建，元组元素的访问，元组的常用操作。

(4) 集合类型

领会：集合的并、交、差以及对称差等运算。

应用：集合对象的创建与删除，集合的访问，集合的常用操作，使用集合来

提取序列中的唯一元素。

(5) 字典类型

应用：字典对象的创建与删除，字典元素的访问读取，字典的常用操作，元素的添加与修改，使用字典来统计。

第 5 章 函数

一、学习目的与要求

(1) 掌握函数的定义与调用方法，理解参数传递规则，能根据需求设计带参数的函数，学习函数设计与参数传递实现业务逻辑。

(2) 能编写简单的递归函数，解决阶乘计算等经典问题。

(3) 区分全局变量与局部变量的作用域，学习变量作用域，管理避免数据污染风险。

(4) 了解 lambda 函数的使用场景并能简单应用。

(5) 递归边界控制培养严谨思维，lambda 表达式体现代码简洁性。

通过本章学习，我们掌握函数定义和调用的用法，理解递归函数的执行过程，对编程如何通过函数设计更加高效清晰的逻辑关系的思维进一步进行锻炼。

二、课程内容

(1) 函数的定义与调用；

(2) 参数传递；（5.1.4 不考试）

(3) 变量作用域；（5.2.3、5.2.4 不考试）；

(4) lambda 表达式函数（5.3.3 不考试）。

三、考核知识点及要求

(1) 函数的定义与调用；

识记：定义函数的语法规则，函数的调用格式。

综合应用：使用函数进行模块化程序设计；应用递归函数解决实际问题；函数嵌套调用，递归调用。

(2) 参数传递；

识记：函数参数的 3 种分类。

领会：位置参数、默认值参数是怎么使用的，形参与实参的概念。

(3) 变量作用域；

领会：全局变量与局部变量的使用规则；，局部作用域，全局作用域。

(4) lambda 表达式函数

应用：lambda 函数完成简单的运算逻辑。

第 6 章 字符串与正则表达式

一、学习目的与要求

(1) 掌握字符串格式化表示；

(2) 掌握字符串的拼接、查找、替换、格式化等操作，能处理文本类数据；

(3) 通过讲解黑客实例及其法律后果，强化学生遵纪守法意识，在字符串处理（如用户密码加密存储）中注重数据安全，避免技术滥用导致的隐私泄露风险；

(4) 字符串加密解密、重复单词检查等实践项目，培养学生严谨细致的编程态度，以及面对复杂文本处理任务时的耐心与专注力，体现工匠精神。遵纪守法。

二、课程内容 (6.3 不考试)

- (1) 字符串概述；
- (2) 字符串处理的函数和方法。

三、考核知识点及要求

- (1) 字符串概述；
- 识记：字符串的表示方法。

领会：字符串的运算。

- (2) 字符串处理的函数和方法

应用：掌握字符串的内置函数和方法，完成文本清洗、格式转换、内容提取等任务。

第 7 章 文件

一、学习目的与要求

(1) 熟练运用文件打开模式，完成文本文件的读写操作，掌握 CSV 文件的基础处理方法。

(2) 强调文件操作中的编码安全与异常处理，培养安全编程意识，保障数据隐私与操作稳定性。

二、课程内容

- (1) 文件基本处理；
- (2) 数据维度；
- (3) 文件管理。

三、考核知识点及要求

- (1) 文件基本处理；

识记：文件的分类。

应用：掌握文件读写方法，实现数据的读取、写入与保存。

- (2) 数据维度

领会：数据维度的概念。

- (3) 文件管理

应用：掌握简单的文件操作，能处理文件读写过程中的常见问题。

第 8 章 Python 面向对象编程

一、学习目的与要求

(1) 理解类与实例的关系，能定义类、创建实例，实现实例属性和方法的访问与调用。

(2) 了解继承的概念，能实现简单的类继承关系，体会多态的编程思想。

(3) 类设计规范培养系统化工程思维，成员访问控制强化严谨编码习惯。

通过学习，了解继承的基本概念，对类和对象有了进一步的了解，为以后设计大型的编程程序奠定了面向对象的基础。

二、课程内容

- (1) 面向对象概述;
- (2) 类对象与实例对象;
- (3) 对象成员;
- (4) 继承和多态。

三、考核知识点及要求

- (1) 面向对象概述;

识记：对象、类的定义。封装、继承、多态的定义。

- (2) 类对象与实例对象

领会：类和实例的构造方法。

- (3) 对象成员

领会：私有成员和共有成员的区别；属性成员和方法成员的区别。

- (4) 继承和多态

领会：继承和多态的实现方式。

综合应用：创建类，创建实例并调用方法，并能编写简单的子类。

三、参考教材与考核实施要求

(一) 本课程使用的参考书

《Python语言程序设计基础教程》，傅清平、李雪斌、徐文胜 著，清华大学出版社，2022年第1版。

教材中第5章的5.1.4、5.2.3、5.2.4、5.3.3，第6章的6.3，第9章不纳入实践考核范围。

(二) 本课程的考试要求

1. 考核场所为计算机机房，机房提供 Python 语言集成开发环境 (IDLE) 或者 anaconda。采用**现场实操考核**方式，考生可携带本书指定教材，不得携带其他任何书籍、参考资料和电子设备。考生需在规定时间内完成题目要求的程序编写、运行、调试及结果提交等任务。考试期间，机房断掉外网，防止考生上网查询答案和作弊。

2. 本门课程时间为 150 分钟。

3. 考试重在考查学生的动手编程能力，多出编程逻辑方面的题目，少出语法需要记忆方面的题目。

4. 考生的答案以每道题目最后一次提交为准。

(三) 关于本课程考试命题的若干规定

1. 本大纲各章所规定的基本要求，知识点及知识点下的知识细目，都属于考核的内容。考试命题既要覆盖到章，又要避免面面俱到。要注意突出课程的重点、章节重点，侧重综合应用能力的考查。

2. 命题不应有超出大纲中考核知识点范围的题，考核目标不得高于大纲中所规定的相应的最高能力层次要求。命题应着重考核自学者对基本概念、基本知识和基本理论是否了解或掌握，对基本创作实践方法是否会用或熟练。不应出与基本要求不符的偏题或怪题。

3. 教材中第 5 章的 5.1.4、5.2.3、5.2.4、5.3.3，第 6 章的 6.3，第 9 章不纳入实践考核范围。

4. 试题难易程度的比例基本为：易 20%；较易 35%；较难 35%；难 10%。

5. 课程考试命题的主要题型有简单实操题、综合实操题等题型。