
《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名：

年 月 日

课题名称	第 1 章 Python 概述	计划学时	3 学时
内容分析	Python 诞生于 20 世纪 90 年代初，是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python 具有语法简洁、易于学习、功能强大、可扩展性强、跨平台等诸多特点，逐渐成为最受欢迎的程序设计语言之一。		
教学目标及基本要求	1、了解 Python 的发展历程 2、了解 Python 的特点和应用领域 3、可以独立完成 Python 的安装 4、会简单使用 PyCharm 新建 Python 文件 5、掌握 Python 程序的执行原理		
教学重点	1、独立完成 Python 的安装 2、会使用 PyCharm 新建 Python 文件 3、掌握 Python 程序的执行原理		
教学难点	1、掌握 Python 程序的执行原理		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	第一课时 （Python 基础知识，Python 的安装） 一、创设情境，导入 Python 新概念 （1）教师通过 TIOBE 排行榜，引出本节课的主题 Python。 教师提问学生问题，例如问题是：大家知道 C 语言吗？现在哪种语言比较流行？然后带领大家查看 TIOBE 排行榜，引出 Python 语言。 （2）明确本节课的学习目标，让学生带着目标去听课。 ● 要求学生理解 Python 语言的特点 ● 要求学生知道 Python 2.x 和 3.x 的区别 ● 要求学生了解每个平台如何安装 Python 二、进行重点知识的讲解 （1）教师根据课件，讲述 Python 语言的发展历程。 （2）教师根据课件，讲解 Python 语言的显著特点。 Python 语言的特点包括简单易学、开源、高级语言、可移植性、解释性、面向对象、可扩展性、丰富的库和规范的代码。		

(3) 教师通过举例，引出 Python 语言的应用领域。

例如，安卓负责开发安卓手机的应用，iOS 用于开发 iPhone 和 iPad 的应用。

(4) 教师根据课件，查看 Python 发布的历史版本。

在本书成稿时，Python 2 的最新版本是 Python 2.7.13，Python 3 的最新版本是 Python 3.6。Python 3.x 不再兼容现有的 2.x 程序，在实际使用时，大家要根据实际情况选择使用的版本。

(5) 教师根据课件，讲述 Python 2.x 和 3.x 的区别。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(6) 教师根据课件，讲述不同平台下 Python 的安装。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

三、归纳总结，布置作业/随堂练习

(1) 回顾上课前的学习目标，并对本节课的重要知识点进行总结。

教师提问学生本节课需要掌握的知识点，包括 Python 语言的特点有哪些，Python 2 和 3 最新的版本，以及 2.x 和 3.x 版本的不同，并对学生的回答进行补充和总结。

(2) 使用博学谷系统下发课后作业。

第二课时

(IPython, PyCharm)

一、回顾上一课时讲解的知识，引出本课时要讲解的知识点

(1) 教师讲解上一课时的课后作业。

(2) 什么是交互式解释器？

Python 的优点是其交互式解释器，称为 shell。shell 提供了一种能够快速实现灵感、检验特性的方法，以及交互式的模块界面，能够将一些需要两三行脚本才能完成的任务一次性完成。

(3) 教师介绍 Python 交互式的特点，引出 IPython 解释器。

通常编写代码时，会采用同时运行文本编辑器和 Python 的方式，通过交互地使用编辑器和 shell，也就是在两者之间切换来完成程序的编写。IPython 具有卓越的 Python shell，其性能远远优于标准 Python 的 shell。

(4) 明确本节课的学习目标，让学生带着目标去听课。

- 要求学生知道如何安装 IPython 解释器

- 要求学生安装和使用 PyCharm

二、进行重点知识的讲解

- (1) 教师根据课件，讲解如何安装 IPython。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

- (2) 什么是 PyCharm?

PyCharm 是 JetBrains 开发的 Python IDE。PyCharm 用于一般 IDE 具备的功能，比如，调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。

- (3) 教师根据课件，讲解如何下载和安装 PyCharm。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

- (4) 教师根据课件，讲解如何使用 PyCharm。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

- (5) 教师通过举例，引出 Python 程序是如何执行的。

例如，使用 C/C++ 之类的编译性语言编写的程序，是需要从源文件转换成计算机使用的机器语言，经过链接器链接之后形成了二进制的可执行文件。运行该程序的时候，就可以把二进制程序从硬盘载入到内存中并运行。但是对于 Python 而言，Python 源码不需要编译成二进制代码，它可以直接从源代码运行程序。

- (6) 教师根据课件，介绍 Python 程序执行的原理。

当我们运行 Python 文件程序的时候，Python 解释器会执行两个步骤，一是把源代码编译成字节码，另一是把编译好的字节码转发到 Python 虚拟机 (PVM) 中进行执行。

- (7) 教师安排学生课上完成章节课后习题，并对一些难以理解的题目进行讲解。

三、归纳总结，布置作业/随堂练习

- (1) 回顾上课前的学习目标，并对本节课的重要知识点进行总结。

教师提问学生 Python 程序是如何执行的，并对学生的回答进行补充和总结。

- (2) 使用博学谷系统下发课后作业。

第三课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，

	对代码的熟练程度。 上机一：（考察知识点为 Windows 安装 Python 开发环境） 形式：单独完成 题目： 请按照以下需求安装 Python 开发环境。 1) 访问官网下载基于 Windows 平台的 Python 安装包； 2) 安装并配置 Python 环境。 上机二：（考察知识点为下载和安装 PyCharm，并使用 PyCharm 编写第一个 Python 程序） 形式：单独完成 题目： 请按照以下需求编写一个简单的 Python 程序。 1) 访问 PyCharm 官网下载 Windows 平台的安装包，并安装 PyCharm； 2) 使用 PyCharm 工具创建项目文件 HelloWorld.py； 3) 使用 PyCharm 工具编写程序，并运行成功。
思考题 和习题	见教材第 1 章配套的习题
教 学 后 记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 2 章 Python 基础语法	计划学时	4 学时
内容分析	Python 语言之所以简单易学，离不开它的基础语法。同其他计算机语言一样， Python 作为一门独立的语言，有自己独树一帜的特色语法。		
教学目标及基本要求	1. 掌握 Python 中的变量和变量类型 2. 掌握 Python 中的标识符，能准确判断标识符的合法性 3. 了解 Python 中的关键字，会借助工具查看关键字信息 4. 了解不同运算符的作用，会进行不同的数值运算		

教学重点	1. 掌握 Python 中的变量和变量类型 2. 掌握 Python 中的标识符，能准确判断标识符的合法性
教学难点	1. Python 的数据类型转换 2. Python 的位运算
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （基本语法、变量和数据类型，标识符和关键字） 一、 创设情境，导入基本语法 （1）教师通过列出 Python 代码，引出其具体的基本语法 教师通过一张 Python 代码截图，展现 Python 的代码结构，并且介绍每个结构的组成。例如，为了能编写规范的代码，接下来了解 Python 的一些基本语法，包括注释、行与缩进以及语句换行。 （2）明确学习目标 ● 要求学生理解 Python 的基本语法 ● 要求掌握变量和变量的类型 ● 要求学生熟悉 Python 中的标识符和关键字 二、 进行重点知识的讲解 （1）什么是注释？ Python 中的单行注释以#开头，多行注释可以使用三引号作为开头和结束符号，三引号可以是三个单引号或者三个双引号。 （2）对比 C 语言的条件语句，讲解 Python 缩进的代码风格。 c 语言使用{}来表示一个代码块，Python 使用缩进实现，这里可以使用图片区分这两种语言的缩进风格。缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。 （3）教师通过举例，讲解语句换行的知识。 例如，教师编写很长的一句文本内容，出现回行的情况。由这个问题的出现，引出 Python 使用“（）”处理语句过长的情况。注意：在 [], {}, 或 () 中的语句，不需要使用圆括号进行换行。

(4) 教师通过举例，讲解如何使用变量。

现实生活中，逛超市买东西时，往往通过一个购物车放置物品，等所有的物品购买完毕后，到收银台结账。在 Python 中，若要存储数据，需要用到变量。变量可以理解为去超市购物的菜篮子，它的类型和值在赋值的那一刻被初始化。

(5) 教师根据课件，讲解如何给变量赋值。

变量的赋值是通过等号来表示的。如果硬件条件允许的情况下，教师可以通过实践演示操作。

(6) 教师通过举例，引出变量的类型。

例如，逛超市的时候有手提筐和购物车，买的东西比较少，就用手提筐放东西，买的东西比较多，就用购物车放东西。同理，如果使用变量在存储数据时，为了更充分利用内存空间，我们可以为变量指定不同的数据类型。

(7) 教师根据课件，讲解变量的几种类型。

变量的数据类型包括：数字类型、布尔类型、字符串、列表和元组、字典。在 Python 中，只要定义了一个变量，并且该变量存储了数据，那么变量的数据类型就已经确定了。如果希望查看变量的类型，可以使用“type(变量的名字)”来实现。

(8) 教师根据课件，讲解标识符及其命名规范。

Python 中的标识符由字母、数字和下划线“_”组成，且不能以数字开头；

Python 中的标识符时区分大小写的；

Python 中的标识符不能使用关键字。

(8) 教师根据课件，讲解 Python 中的关键字。

关键字是 Python 语言自己已经使用的了，不允许开发者自己定义和关键字相同的名字的标示符。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

带领学生总结本课内容：Python 的基本语法、变量、变量的数据类型、标示符，以及关键字。

(2) 使用博学谷系统下发课后作业。

（简单数据类型，运算符）

一、 回顾上节课讲解的知识，继续讲解本节课的知识

（1） 教师讲解上一课时的课后作业。

（2） 回顾上节课所学习的知识，由上一课时知识点引出本课时内容。

教师提问学生问题，问题是变量有哪些数值类型？然后引出本节的知识点：本节课负责介绍这些简单的数值类型。

（3） 明确学习目标

- 要求学生掌握数值类型的使用
- 要求学生掌握运算符的使用

二、 进行重点知识的讲解

（1） 教师根据课件，讲述整数类型的概念。

整数类型（int）简称整型，它用于表示整数，例如，100、2016 等。整型字面值的表示方式有三种，分别是十进制、二进制（以“0B 或 0b”开头）、八进制（以数字“0”开头）和十六进制（以“0x”或“0X”开头）。

（2） 教师通过示例，讲述如何表示整数及整型面值的切换。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

（3） 教师根据课件，讲述浮点类型的概念。

浮点型（Float）用于表示实数，例如，3.14、9.19 等都属于浮点型。浮点型字面值可以用十进制或科学计数法表示。

（4） 教师通过示例，讲述浮点数的范围。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

（5） 教师根据课件，讲述布尔类型的概念。

布尔型其实是整型的子类型，布尔型数据只有两个取值：True 和 False，分别对应整型的 1 和 0。

（6） 教师根据课件，讲述复数类型的概念。

复数类型，用于表示数学中的复数，例如，5+3j、-3.4-6.8j 都是复数类型。Python 中的复数类型是一般计算机语言所没有的数据类型。注意：一个复数必须有表示虚部的实数和 j。

（7） 教师通过示例，讲述数字类型的转换。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(8) 什么是运算符？

对数据的变换称为运算，表示运算的符号称为运算符，参与运算的数据被称为操作数，举个简单的例子 $4+5$ ，这是一个加法运算，“+”称为运算符，4 和 5 称为操作数。

(9) 教师通过课件的案例，讲述算术运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(10) 教师通过课件的示例，讲述赋值运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(11) 教师通过课件的案例，讲述复合赋值运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(12) 教师通过课件的案例，讲述比较运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(13) 教师通过课件的案例，讲述逻辑运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(14) 教师通过课件的示例，讲述成员运算符的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

三、归纳总结，布置课后作业

(1) 回顾课前学习目标，对本节课的内容进行总结。

教师带领学生总结本节课要掌握的内容：简单的数值类型有哪些，以及常见运算符有哪些。

(2) 布置随堂练习，检查学生学习状况。

根据博学谷和随堂练习资源，给学生布置随堂练习，根据学生的完成情况，了解学习的学习理解状况，根据学生反映的问题，进行解答。

(3) 使用博学谷系统下发课后作业。

第三课时

(位运算，运算符优先级)

一、回顾上节课讲解的数值和运算符，继续讲解位运算的知识

(1) 教师讲解上一课时的课后作业。

	(2) 回顾上节课所学习的知识，继续讲解本课时的位运算的相关知识。 程序中所有的数在计算机内存中都是以二进制的形式存储的，位运算其实就是直接对整数在内存中的二进制位进行操作。 (3) 教师列举位运算。 按位左移、按位右移、按位与、按位或、按位异或、按位取反。 (4) 明确学习目标，掌握运算符的优先级。 二、 进行重点知识的讲解 (1) 教师使用画图工具，介绍什么是按位左移。 按位左移指的是二进位全部左移 n 位，高位丢弃，低位补 0，符号“<<”是按位左移的符号。 (2) 教师通过示例，帮助学生理解按位左移。 如果硬件条件允许的情况下，教师可以通过实践演示操作。 (3) 教师使用画图工具，介绍什么是按位右移。 按位右移指的是将二进位全部右移 n 位，移出的位丢弃，移进的位补符号位。按位右移的符号位保持不变。符号“>>”是按位右移的符号。 (4) 教师通过示例，帮助学生理解按位右移。 如果硬件条件允许的情况下，教师可以通过实践演示操作。 (5) 教师使用画图工具，介绍什么是按位与。 按位与指的是参与运算的两个数各对应的二进位进行“与”的操作。只有对应的两个二进位都是 1 时，结果位就为 1，否则结果位为 0。符号“&”是按位与的符号。 (6) 教师通过示例，帮助学生理解按位与。 如果硬件条件允许的情况下，教师可以通过实践演示操作。 (7) 教师使用画图工具，介绍什么是按位或。 按位或指的是参与运算的两个数各对应的二进位进行“或”的操作。只要对应的两个二进位有一个为 1 时，结果位就为 1。当参与运算的是负数时，参与运算的两个数均以补码出现。按位或使用符号“ ”表示。 (8) 教师通过示例，帮助学生理解按位或。 如果硬件条件允许的情况下，教师可以通过实践演示操作。 (9) 教师使用画图工具，介绍什么是按位异或。
--	--

按位异或就是将参与运算的两个数对应的二进制位进行比较，如果一个位为 1，另一个位为 0，则结果为 1，否则，结果位为 0。按位异或使用“^”符号来表示。

(10) 教师通过示例，帮助学生理解按位异或。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(11) 教师使用画图工具，介绍什么是按位取反。

按位取反就是将二进制的每一位进行取反，0 取反为 1，1 取反为 0。

(12) 教师通过示例，帮助学生理解按位取反。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(13) 教师根据课件，讲解运算符的优先级。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(14) 教师布置学生课上完成课后习题，并对一些难以理解的题目进行讲解。

三、归纳总结，布置课后作业

(1) 回顾位运算和运算符的优先级，对本节课的内容进行总结

(2) 使用博学谷系统下发课后作业

第四课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

形式：独立完成

上机一：（考察知识点为数字类型的转换，算术运算符）

题目：

请按照以下要求完成编程。

要求如下：

- 1) 接收用户输入的两个数字
- 2) 计算两个数字的和

	形式：单独完成 上机二：（考察知识点为数比较运算符） 题目： 请按照以下要求完成编程。 要求如下： 1) 接收用户输入的两个数字； 2) 使用 if-else 语句比较它们的大小
考题和习题	见教材第 2 章配套的习题
教学后记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 3 章 Python 常用语句	计划学时	6 课时
内容分析	俗话说“不积跬步无以至千里”。学习了 Python 的基本语法后,接下来进入 Python 常用语句的学习,本章将针对 Python 中的判断语句以及循环语句进行详细的讲解。		
教学目标及基本要求	6、掌握判断语句的使用 7、掌握循环语句的使用 8、掌握 break、continue、pass 和 else 语句的作用		
教学重点	4、掌握判断语句的使用 5、掌握循环语句的使用 6、掌握 break、continue、pass 和 else 语句的作用		
教学难点	2、掌握循环语句的使用 3、掌握 break、continue、pass 和 else 语句的作用		
教学方式	教学采用教师课堂讲授为主,使用教学 PPT 讲解		

教 学 过 程	第一课时 (判断语句) 一、创设情境，导入判断语句的概念 (1) 教师通过生活中的举例，引出判断语句。 在生活中，过马路要看红绿灯。如果是绿灯才能通过马路，否则需要停止等待。在程序中，用户登录要输入用户名和密码，只有两者全部正确才能允许登录。为此，Python 提供了判断语句。 (2) 明确学习目标 ● 要求学生掌握判断语句的使用 二、进行重点知识的讲解 (1) 教师根据课件，讲述 if 语句的格式和执行流程。 只有判断条件成立，才可以执行下面的语句。 (2) 教师通过 3.1.1 的案例，讲述 if 语句的使用。 (3) 教师根据课件，讲述 if-else 语句的格式和执行流程。 只有判断条件成立，才可以执行满足条件时要做的事情，否则，执行不满足条件要做的事情。 (4) 教师通过 3.1.2 的案例，讲述 if-else 语句的使用。 (5) 教师根据课件，讲述 if-elif 语句的格式和执行流程。 (6) 教师通过 3.1.3 的案例，讲述 if-elif 语句的使用。 (7) 教师通过举例，引出 if 嵌套的概念。 例如，乘火车时，必须先买票，只有拿到车票才能进入安检，安检通过以后方可正常乘车。基于这种情况，后面的过程是在前面成立的基础上进行的。在编程中，可以使用 if 嵌套实现。 (8) 教师根据课件，讲述 if 语句嵌套的格式和执行流程。 (9) 教师通过 3.1.4 的案例，讲述 if 嵌套的使用。 (10) 教师通过猜拳游戏的案例，带领学生巩固对判断语句的认识。 如果条件允许，教师可以通过实践演示操作。 三、归纳总结，布置作业/随堂练习 (1) 回顾上课前的学习目标，并对本节课的内容进行总结。 教师总结本节课需要掌握的知识点，包括 if 语句、if-else 语句、if-elif 语句，以及 if 语句嵌套。
------------------	---

(2) 使用博学谷系统下发课后作业。

第二课时

(while 循环, for 循环, while 循环案例)

一、回顾上节课的内容, 继续讲解本课时的循环语句

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容, 教师通过举例, 引出循环语句。

在生活中, 每天都是重复的事情: 起床、吃饭, 睡觉。程序中, 执行重复操作需要用到循环语句, 包括 while 循环和 for 循环。

(3) 明确学习目标

- 要求学生掌握循环语句的使用

二、进行重点知识的讲解

(1) 教师根据课件, 讲述 while 语句的格式和执行流程。

在 while 循环中, 同样需要注意冒号和缩进。另外, 在 Python 中没有 do..while 循环。

(2) 教师通过 3.2.1 的案例, 讲述 while 循环的使用。

(3) 教师根据课件, 讲述 for 循环语句的格式, 以及 range 函数的使用。

(4) 教师给学生布置 3.2.3 的案例, 让学生自己动手完成, 并对学生出现的问题进行一一解答。

三、归纳总结, 布置作业

(1) 回顾学习目标, 对本节课的内容进行总结。

教师带领学生总结本节课需要掌握的内容, 包括 while 循环和 for 循环。

(2) 布置随堂练习, 检查学生掌握情况。

根据博学谷和随堂练习资源, 给学生布置随堂练习, 检测学生的掌握程度, 并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第三课时

(while 嵌套, while 嵌套应用)

一、回顾上节课内容, 继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，继续介绍 `while` 循环嵌套的知识。

在上节课中，介绍了 `while` 循环的内容。我们知道，`if` 语句是能够嵌套的，那么 `while` 循环同样能嵌套使用，接下来针对循环嵌套的知识进行介绍。

(3) 明确学习目标

- 要求学生掌握循环语句的使用

二、进行重点知识的讲解

(1) 教师根据课件，讲述 `while` 语句嵌套的格式和执行流程。

(2) 教师通过打印三角形的应用，讲述 `while` 语句嵌套的使用。

(3) 教师给学生布置 3.2.6 的需求，让学生自己动手完成，并对学生出现的问题进行一一解答。

三、归纳总结，布置作业

(1) 回顾学习目标，对本节课的内容进行总结

教师带领学生总结 `while` 语句嵌套的使用。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时

(Python 其它语句)

一、回顾上节课内容，引出 `break` 语句的知识

(1) 教师讲解上节课有难度的作业，并对学生的疑问进行统一答疑。

(2) 回顾上节课的内容，教师由程序的需求，引出 `break` 语句的使用。

`break` 语句用于结束整个循环。

(3) 明确学习目标

- 要求学生掌握其它语句的使用

二、进行重点知识的讲解

(1) 教师通过 3.3.1 的案例，讲述 `break` 语句在 `for` 循环中的使用。

(2) 什么是 `continue` 语句？

`continue` 的作用是用来结束本次循环，紧接着执行下一次的循环。

(3) 教师通过 3.3.2 的案例，讲述 `continue` 语句在 `for` 循环中的使用。

(4) 什么是 `pass` 语句？

Python 中的 `pass` 是空语句，它的出现是为了保持程序结构的完整性。`pass` 不做任何事情，一般用做占位语句。

(5) 教师通过 3.3.4 的案例，讲述 `else` 语句在 `while` 循环中的使用。

(6) 教师布置学生课上完成本章习题，并对一些难以理解的题目进行讲解。

三、归纳总结，布置作业

(1) 回顾学习目标，对本节课的内容进行总结。

教师带领学生总结本节课的内容，内容包括 `break` 语句、`continue` 语句、`pass` 语句，以及 `else` 语句。

(2) 使用博学谷系统下发随堂练习，检测学生对课堂内容的掌握情况。

第五课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为 `if-elif` 语句）

形式：单独完成

题目：

请按照以下要求，完成人机猜拳的游戏。

要求如下：

- 1) 0 表示剪刀，1 表示石头，2 表示布；
- 2) 接收用户输入的数，只能是 0、1、2；
- 3) 电脑随机产生一个 0~2 的数字；
- 4) 比较电脑和玩家的出拳，分别输出赢、输和平局的情况。

上机二：（考察知识点为 `if` 语句嵌套）

形式：单独完成

题目：

请按照以下要求操作。

要求如下：

- 1) 通过变量 `num1` 和 `num2` 接收用户输入的两个数值；
- 2) 定义一个表示计算符号的变量 `calculate`；
- 3) 如果 `calculate` 为 “+”，计算 `num1` 和 `num2` 的和；如果为 “-”，计算

	它们的差；如果为“*”，计算它们的乘积。 4) 如果 calculate 为“/”，必须保证 num2 的值不能为 0 第六课时 (上机练习) 上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。 上机一：（考察知识点为 while 循环） 形式：单独完成 题目： 请使用 while 循环，打印九九乘法表。
思考题 和习题	见教材第 3 章配套的习题
教 学 后 记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

20 年 月 日

课题名称	第 4 章 字符串	计划学时	6 课时
内容分析	当打开浏览器登录某些网站的时候，需要输入密码，浏览器把密码传送到服务器后，服务器会对密码进行验证，其验证过程是把之前保存的密码与本次传递过去的密码进行对比，如果相等，那么就认为密码正确，否则就认为不对。服务器既然想要存储这些密码可以用数据库（比如 MySQL），当然为了简单起见，咱们可以先找个变量把密码存储起来即可，而可以存储带有字母密码的就是字符串。		
教学目标及基本要求	9、掌握字符串的输入和输出 10、会使用切片的方式访问字符串中的值 11、掌握常见的字符串的内建函数		
教学重点	7、掌握字符串的输入和输出 8、会使用切片的方式访问字符串中的值 9、掌握常见的字符串的内建函数		

教学难点	4、会使用切片的方式访问字符串中的值 5、掌握常见的字符串的内建函数
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （判断语句） 一、创设情境，导入字符串的概念 （3）教师通过生活中的举例，引出字符串的知识。 在生活中，登录某个网站时需要输入用户名和密码进行验证，过程就是把保存的用户信息跟本次传递的进行比对。在程序中，可以找个变量存储密码，能存储带有字母密码的就是字符串。 （4）什么是字符串？ 字符串是一种表示文本数据的类型，字符串中的字符可以是 ASCII 字符、各种符号以及各种 Unicode 字符。 （5）明确学习目标 ● 要求学生创建字符串 ● 要求学生掌握字符串的输入与输出 二、进行重点知识的讲解 （11）教师根据课件，讲述表示字符串的 3 种方式，并且通过示例进行验证。 使用单引号、双引号或三引号表示字符串。 （12）通过单引号表示字符串的错误，引出转义字符。 例如，使用单引号表示“let's go!”时，程序会出现报错信息。这时，我们需要使用斜线对单引号进行转义。 （13）教师根据课件，介绍常见的一些转义字符。 如果条件允许的情况下，教师可以抽取任意个比较典型的转义字符，通过示例代码演示操作。 （14）通过多个重复输出语句的弊端，抛出格式化输出的知识。 例如，输出“我今年**岁”时，每个人的结果因人而异。为此，可以用字符串格式输出，就是将一个值插入到一个有字符串格式符%s 的字符串中。 （15）教师根据课件，介绍常用的一些格式化符号。

(16) 教师通过示例，讲解比较常见的格式化符号。

(17) 教师根据课件，介绍字符串输入的内容。

Python 3 提供了 `input()` 函数从标准输入读取一行文本，默认的标准输入是键盘。

(18) 教师通过 4.2.2 的案例，讲述字符串的输入。

三、归纳总结，布置作业/随堂练习

(3) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括字符串的格式、字符串的输出，以及字符串的输入。

(4) 使用博学谷系统下发课后作业。

第二课时

(访问字符串的值，字符串内建函数（部分）)

一、回顾上节课的内容，继续讲解本课时的知识

(4) 教师对学生们的疑问进行统一答疑。

(5) 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了字符串的内容，接下来，本节课继续介绍字符串的内容。

(6) 明确学习目标

- 要求学生理解字符串的存储方式
- 要求学生使用切片截取字符串
- 要求学生掌握内建函数的使用

二、进行重点知识的讲解

(5) 字符串是怎样存储的？

如果希望访问字符串中的值，需要使用下标来实现。每个字符都对应着一个编号，并且编号是从 0 开始的，这些编号就代表着下标。如果要从字符串中取出字符，可以通过下标获取。

(6) 教师通过 4.3.2 的案例，讲述如何使用切片截取字符串。

(7) 教师根据课件，介绍字符串比较常见的内置函数。

(8) 教师通过 4.4.1 的案例，讲述 `find` 函数的使用。

(9) 教师通过 4.4.2 的案例，讲述 `index` 函数的使用。

(10) 教师从题库中随机抽取几道与上述两个函数相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。

(11) 教师通过 4.4.3 的案例，讲述 count 函数的使用。

三、归纳总结，布置作业

(2) 回顾学习目标，对本节课的内容进行总结。

教师带领学生总结本节课需要掌握的内容，包括字符串的存储方式、使用切片截取字符串，以及部分内建函数的使用。

(3) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(4) 使用博学谷系统下发课后作业。

第三课时

(字符串内建函数（部分）)

一、回顾上节课内容，继续讲解本节课的内容

(4) 教师对学生们的疑问进行统一答疑。

(5) 回顾总结上节课内容，继续介绍内建函数的知识。

在上节课中，介绍了字符串的内建函数，接下来继续介绍其它函数。

(6) 明确学习目标

- 要求学生掌握内建函数的使用

二、进行重点知识的讲解

(4) 教师通过 4.4.4 的案例，讲述 replace 函数的使用。

(5) 教师通过 4.4.5 的案例，讲述 split 函数的使用。

(6) 教师通过 4.4.6 的案例，讲述 capitalize 函数的使用。

(7) 教师从题库中随机抽取几道与上述这些函数相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。

(8) 教师通过 4.4.7 的案例，讲述 title 函数的使用。

(9) 教师通过 4.4.8 的案例，讲述 startswith 函数的使用。

(10) 教师通过 4.4.9 的案例，讲述 endswith 函数的使用。

(11) 教师通过 4.4.10 的案例，讲述 upper 函数的使用。

(12) 教师从题库中随机抽取几道与上述这些函数相关的题目，让学生自

已动手编程，并根据学生出现的问题进行解答。

三、归纳总结，布置作业

(2) 回顾学习目标，对本节课的内容进行总结

教师带领学生总结字符串内建函数的使用。

(3) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时

(字符串内建函数(部分))

一、回顾上节课内容，继续讲解本节课的内容

(4) 教师讲解上节课有难度的作业，并对学生的疑问进行统一答疑。

(5) 回顾上节课的内容，继续介绍内建函数的知识。

在上节课中，介绍了字符串的内建函数。接下来继续介绍其它函数。

(6) 明确学习目标

- 要求学生掌握内建函数的使用

二、进行重点知识的讲解

(1) 教师通过 44.11 的案例，讲述 ljust 函数的使用。

(2) 教师通过 4.4.12 的案例，讲述 rjust 函数的使用。

(3) 教师通过 4.4.13 的案例，讲述 center 函数的使用。

(4) 教师从题库中随机抽取几道与刚刚介绍的上述三个函数相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。

(5) 教师通过 4.4.14 的案例，讲述 lstrip 函数的使用。

(6) 教师通过 4.4.15 的案例，讲述 rstrip 函数的使用。

(7) 教师通过 4.4.16 的案例，讲述 strip 函数的使用。

(8) 教师从题库中随机抽取几道与刚刚介绍的上述三个函数相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。

(9) 教师根据课件，讲解字符串的运算符，并且通过实践演示操作。

三、归纳总结，布置作业

(1) 回顾学习目标，对本节课的内容进行总结。

教师带领学生总结字符串内建函数的使用。

(2) 使用博学谷系统下发随堂练习，检测学生对课堂内容的掌握情况。

第五课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为字符串的格式输出）

形式：独立完成

题目：

请按照以下要求完成。

要求如下：

- 5) 定义表示姓名的变量 `name`，赋值为“小明”；
- 6) 定义表示年龄的变量 `age`，赋值为“28”
- 7) 定义表示体重的变量 `height`，赋值为“1.85”
- 8) 使用 `print` 函数输出上述信息。

上机二：（考察知识点为使用切片截取字符串）

形式：独立完成

题目：

请按照以下要求操作。

要求如下：

- 5) 定义一个字符串，内容为“传智黑马欢迎您”；
- 6) 使用切片截取下标为 0~1 的字符，并打印输出；
- 7) 使用切片截取下标为 2~末尾的字符，并打印输出；
- 8) 使用切片截取下标为 2~倒数第 2 个字符，并打印输出；
- 9) 使用切片截取倒序且步长为 3 的字符，并打印输出。

第六课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练

	程度。 上机一：（考察知识点为字符串的内建函数） 形式：单独完成 题目： 已知有字符串“000itcast and itheima000”，请按照以下要求操作。 要求如下： 1) 查找 it 出现的索引位置，并打印输出； 2) 检测 it 出现的次数； 3) 把“000”替换为空格； 4) 把所有字符转换为大写； 5) 删除字符两头的空格 上机二：（考察知识点为字符串运算符） 形式：单独完成 题目： 已知字符串“itcast”，请按照以下要求操作。 要求如下： 1) 使用 not in 运算符，检测字符串中是否包含“heima” 2) 如果不包含，就在字符串后面加上 heima，并使用空格分隔 3) 让每个单词的首字母变成大写 4) 分别截取两个单词为两个子串
思考题 和习题	见教材第 4 章配套的习题
教 学 后 记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 5 章 列表、元组和字典	计划学时	5 课时
内容分析	序列是 Python 中最基本的数据结构。序列中的每个元素都分配一个数字—它的位置或索引，第一个索引是 0，第二个索引是 1，依此类推。Python 有 6 个序列的内置类型，但最常见的是列表和元组。除此之外，Python 还提供了一种存储数		

	据的容器—字典。
教学目标及基本要求	12、 掌握什么是列表以及列表的常见操作 13、 掌握列表的嵌套使用 14、 掌握元组的使用 15、 掌握什么是字典以及字典的常见操作
教学重点	10、 掌握什么是列表以及列表的常见操作 11、 掌握列表的嵌套使用 12、 掌握元组的使用 13、 掌握什么是字典以及字典的常见操作
教学难点	6、 掌握什么是列表以及列表的常见操作 7、 掌握列表的嵌套使用 8、 掌握什么是字典以及字典的常见操作
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （列表概述，列表遍历，列表常见操作（部分）） 一、创设情境，导入列表的概念 （6） 教师通过提出需求，引出列表的知识。 假设一个班有 100 名学生，如果要存储所有学生的名字，可以使用 100 个变量存储，这样是极其繁琐的，列表可以很好地解决这个问题。 （7） 什么是列表？ 列表是 Python 中的一种数据结构，它可以存储不同类型的数据。 （8） 明确学习目标 ● 要求学生会创建列表 ● 要求学生掌握列表的遍历 ● 要求学生会列表中添加、查找和修改元素 二、进行重点知识的讲解 （19） 教师根据课件，讲述创建列表的方式，并通过示例进行验证。 创建列表的方式很简单，只需要把逗号分隔的不同的数据项使用方括号括起来即可。 （20） 教师通过 5.1 的案例，讲述如何使用索引访问列表元素。

与字符串的索引一样，列表索引是从 0 开始的。

(21) 教师通过 5.2 的案例，讲解使用 for 循环和 while 循环遍历列表。

(22) 教师从题库中随机抽取几道比较典型的题目，随机提问学生问题，并根据学生回答进行补充和总结。

(23) 教师通过 5.3.1 的案例，讲解如何在列表中增加元素。

(24) 教师通过 5.3.2 的案例，讲解如何在列表中查找元素。

(25) 教师通过 5.3.3 的案例，讲解如何在列表中修改元素。

三、归纳总结，布置作业/随堂练习

(5) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括列表概述、遍历列表，以及列表的常见操作（增加、查找、修改）。

(6) 使用博学谷系统下发课后作业。

第二课时

（列表的常见操作（部分），列表的嵌套，元组）

一、回顾上节课的内容，继续讲解本课时的知识

(7) 教师对学生们的疑问进行统一答疑。

(8) 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了列表的一些基础知识，以及一些常见操作。接下来，本节课继续介绍列表的其它常见操作，包括删除列表元素、排序列表等。

(9) 明确学习目标

- 要求学生掌握列表的常见操作（删除、排序）
- 要求学生熟悉列表的嵌套
- 要求学生掌握元组的使用

二、进行重点知识的讲解

(12) 教师通过 5.3.4 的案例，讲解在列表中删除元素的三种方式。

(13) 教师通过 5.3.5 的案例，讲解列表的排序操作。

(14) 什么是列表的嵌套？

列表的嵌套指的是一个列表的元素又是一个列表。

(15) 教师通过 5.4.2 的案例，讲解列表的嵌套。

(16) 和列表相比, 元组有什么特点?

Python 的元组与列表类似, 不同之处在于元组的元素不能修改, 元组使用圆括号, 列表使用方括号。

(17) 教师根据课件, 讲述创建元组的方式。

元组的创建很简单, 只需要在圆括号中添加元素, 并使用逗号分隔即可。

(18) 教师通过案例, 讲解如何使用下标访问元组的元素。

(19) 教师通过案例, 讲解怎样对元组进行连接组合。

(20) 教师通过案例, 讲解元组的遍历操作。

(21) 教师根据课件, 介绍元组的内置函数, 并抽取几个函数使用程序演示讲解。

三、归纳总结, 布置作业

(3) 回顾学习目标, 对本节课的内容进行总结。

教师带领学生总结本节课需要掌握的内容, 包括列表的常见操作(删除、排序)、列表的嵌套, 以及元组的常见操作。

(4) 布置随堂练习, 检查学生掌握情况。

根据博学谷和随堂练习资源, 给学生布置随堂练习, 检测学生的掌握程度, 并对学生出现的问题进行解决。

(5) 使用博学谷系统下发课后作业。

第三课时

(字典)

一、回顾上节课内容, 导入字典的概念

(7) 教师对学生们的疑问进行统一答疑。

(8) 回顾总结上节课内容, 根据程序的需求, 引出字典的概念。

假设列表中的元素位置发生了变化, 如果想准确地找到某个人的信息, 显然不是很便捷。因此, Python 提供了既可以存储多个数据, 又能快速准确定位到元素的数据类型, 那就是字典。

(9) 与列表进行对比, 讲述字典的概念和特点。

在软件开发中, 字典是一种存储数据的容器, 它和列表一样, 都可以存储多个数据。不同的是, 列表在查找某个元素时, 是根据下标来进行的, 而字典在查找某个元素时, 是根据‘名字’来查找的。

(10) 明确学习目标

- 要求学生掌握字典的使用

二、进行重点知识的讲解

- (13) 教师根据课件，讲解如何创建字典。
- (14) 教师通过 5-18 的案例，讲述根据键访问字典的值。
- (15) 教师通过 5-19 的案例，讲述如何修改字典的元素。
- (16) 教师从章节习题或题库中随机抽取几道与上述知识点相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。
- (17) 教师通过 5-20 的案例，讲述如何添加字典的元素。
- (18) 教师通过 5-21 和 5-22 的案例，讲述如何删除字典的元素。
- (19) 教师通过 5-23 的案例，讲述计算字典中键值对的个数。
- (20) 教师通过 5-24 的案例，讲述如何获取字典中的键视图。
- (21) 教师通过 5-25 的案例，讲述如何获取字典中值视图。
- (22) 教师通过 5-26 的案例，讲述如何获取字典中元组视图。
- (23) 教师从章节习题或题库中随机抽取几道与上述知识点相关的题目，让学生自己动手编程，并根据学生出现的问题进行解答。
- (24) 教师通过 5.6.3 的案例，讲解字典的遍历方式。

三、归纳总结，布置作业

- (3) 回顾学习目标，对本节课的内容进行总结

教师带领学生总结字典的使用。

- (4) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

- (3) 使用博学谷系统下发课后作业。

第四课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为使用 while 和 for 循环遍历列表）

形式：单独完成

题目：

请按照以下要求完成。

要求如下：

- 9) 定义列表，其内部包括：小明，小红，小兰，小白
- 10) 获得列表的长度
- 11) 使用 while 遍历列表，并输出每个元素的信息。
- 12) 使用 for 遍历列表，并输出每个元素的信息。

上机二：（考察知识点为列表的常见操作）

形式：单独完成

题目：

请按照以下要求操作。

要求如下：

- 10) 定义一个空列表，往列表中添加 5 个任意整数；
- 11) 在列表索引为 1 的位置，插入一个元素 99；
- 12) 找出下标为偶数的元素，并从列表中删除它们；
- 13) 让列表由大到小排序，然后输出。

第五课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为元组的使用）

形式：单独完成

题目：

请按照以下要求操作。

要求如下：

- 6) 定义元组，其包括 6、7、8、9、0 共 5 个元素；
- 7) 定义另一元组，其包括 1、2、3、4、5 共 5 个元素；
- 8) 把上述两个元素合并为一个新的元组；
- 9) 获取元组的最大值和最小值，并打印输出。

上机二：（考察知识点为字符串运算符）

	形式：单独完成 题目： 请按照以下要求操作。 要求如下： 5) 把 1~6 作为字典的键，把 itcast 的每个字符作为字典的值 6) 获取字典的值视图，只要值为 t，就从字典删除这些元素。
思考题 和习题	见教材第 5 章配套的习题
教 学 后 记	

《Python 语言程序设计》

教学设计

课程名称: Python 语言程序设计

授课年级: _____

授课学期: _____

教师姓名: 某某老师

年 月 日

课题名称	第 6 章 Python 函数	计划学时	7 课时
内容分析	在实际开发中,如果有若干段代码的执行逻辑完全相同,那么可以考虑将这些代码抽象成一个函数,这样不仅可以提高代码的重用性,而且条理会更加清晰,可靠性更高。		
教学目标及基本要求	16、 掌握函数的定义和调用方式 17、 掌握函数的参数和返回值 18、 掌握函数的嵌套调用 19、 理解变量作用域 20、 掌握局部变量和全局变量的区别 21、 掌握递归函数与匿名函数的使用 22、 掌握日期函数和随机数函数的使用		
教学重点	14、 掌握函数的定义和调用方式 15、 掌握函数的参数和返回值 16、 掌握函数的嵌套调用 17、 理解变量作用域 18、 掌握局部变量和全局变量的区别		
教学难点	9、 掌握函数的嵌套调用 10、 掌握递归函数与匿名函数的使用		

	11、 掌握日期函数和随机数函数的使用
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （什么是函数，函数的定义和调用，函数的参数，函数的返回值） 一、创设情境，导入函数的概念 （9） 教师根据课件提出的程序需求，引出函数的概念。 例如，要在程序中多个地方输出某个图形，显然重复输出的做法是不可取的。为了提高编写程序的效率以及代码的重用性，我们可以把具有独立功能的代码块组织成一个小模块，这就是函数。 （10） 明确学习目标 ● 要求理解函数的用途 ● 要求学生掌握函数的定义和调用 ● 要求学生会使用函数的参数和返回值 二、进行重点知识的讲解 （26） 教师根据课件，讲述函数定义的基本格式，以及定义规则。 函数代码块以 <code>def</code> 关键词开头，后接函数标识符名称和圆括号 <code>()</code>；任何传入参数和自变量必须放在圆括号中间；函数内容以冒号起始，并且缩进；<code>return</code> [表达式] 结束函数，选择性地返回一个值给调用方。 （27） 教师通过 6.2 的案例，讲解简单函数的定义。 （28） 教师在上述案例的基础上，增加调用函数的代码。 （29） 教师根据程序需求，引出函数传递参数的内容。 例如，定义没有任何参数的函数，用于计算两个数的和，这样每次调用函数执行的结果一样。为了能让定义的函数更加通用，在定义函数时让它用参数接收数据。 （30） 教师通过 6.3.1 的案例，讲述函数传递参数。 （31） 教师通过画图工具，介绍函数传递参数的执行过程。 （32） 教师通过 6.3.2 的案例，讲述函数中默认参数的使用。 （33） 教师通过 6.3.3 的案例，讲述函数中不定长参数的使用。

(34) 教师通过 6.4 的示例，讲述在函数中使用返回值。

三、归纳总结，布置作业/随堂练习

(7) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括函数的定义、函数的调用，以及函数的参数。

(8) 使用博学谷系统下发课后作业。

第二课时

(函数的 4 种类型，函数应用-名片管理系统)

一、回顾上节课的内容，继续讲解本课时的知识

(10) 教师对学生们的疑问进行统一答疑。

(11) 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了函数的参数和返回值，我们知道，有的函数可以有参数或者没有参数，有的函数可以有返回值或者没有返回值。为此，函数有很多类型。

(12) 函数有哪些类型？

函数分为 4 种类型：

- 函数无参数，无返回值
- 函数无参数，有返回值
- 函数有参数，无返回值
- 函数有参数，有返回值

(13) 明确学习目标

- 要求学生知道函数的 4 种类型
- 要求学生会在实际应用中选择合适的函数

二、进行重点知识的讲解

(22) 教师通过 6.5.1 的案例，讲解无参数、无返回值的函数。

(23) 教师通过 6.5.2 的案例，讲解无参数、有返回值的函数。

(24) 教师通过 6.5.3 的案例，讲解有参数、无返回值的函数。

(25) 教师通过 6.5.4 的案例，讲解有参数、有返回值的函数。

(26) 教师以名片管理器应用为例，讲解如何在实际开发中选择合理的函数类型。

(27) 教师让学生随堂练习上述应用，并对使用过程中产生的问题进行一一解答。

三、归纳总结，布置作业

(4) 回顾学习目标，对本节课的内容进行总结。

教师带领学生总结函数的 4 种类型：无参数、无返回值的函数，无参数、有返回值的函数，有参数、无返回值的函数，以及有参数、有返回值的函数。

(5) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(6) 使用博学谷系统下发课后作业。

第三课时

(函数的嵌套调用，变量作用域)

一、回顾上节课内容，继续介绍本课时的内容

(11) 教师对学生们的疑问进行统一答疑。

(12) 回顾总结上节课内容，教师提问学生问题，引出函数的嵌套调用。

例如问题是：什么是函数的嵌套调用？教师根据学生的回答进行总结：在一个函数里面调用了另外一个函数，这就是所谓的函数嵌套调用。

(13) 明确学习目标

- 要求学生掌握函数的嵌套使用
- 要求学生理解变量作用域
- 要求学生掌握局部变量和全局变量的使用

二、进行重点知识的讲解

(25) 教师通过 6.7 的案例，讲述函数的嵌套调用。

(26) 教师通过画图工具，分析嵌套函数的执行过程。

(27) 教师通过对不能位置的变量进行访问，引出作用域的概念。

例如，在程序中定义一个函数，在这个函数内部和外部分别定义两个同名的变量，在函数内部访问这个同名变量，函数输出的是函数内部变量的值。从而抛出一个疑问？为什么输出的是函数内部变量的值呢？究其原因，主要是因为变量作用域的不同所导致的。

(28) 教师根据课件，讲述著名的 LEGB 规则。

(29) 什么是局部变量？

所谓局部变量，就是在函数内部定义的变量。局部变量的作用域是函数内部，意味着它只在定义它的函数中有效，一旦函数结束就会消失。

(30) 什么是全局变量？

定义在函数内部的变量拥有一个局部作用域，定义在函数外的拥有全局作用域。局部变量只能在其被声明的函数内部访问，而全局变量可以在整个程序范围内访问。

(31) 教师通过 6.8.2 的案例，讲述局部变量和全局变量的使用。

(32) 教师通过 6.8.3 的案例，讲述如何在内部作用域中修改外部作用域的变量。

三、归纳总结，布置作业

(4) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括函数的嵌套调用、作用域、LEGB 规则、局部变量、全局变量，以及如何在内部作用域修改外部作用域的变量。

(5) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时

(递归函数，匿名函数，时间函数，日历函数)

一、回顾上节课内容，继续介绍本课时的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，由上节课的内容引出本课时的知识。

在上节课中，我们介绍了嵌套函数的内容，它是在函数的内部调用另一个函数。那么，在函数内部能调用自己本身吗？答案是可以的，这个函数是递归函数。

(3) 明确学习目标

- 要求学生掌握递归函数和匿名函数的使用
- 要求学生掌握时间和日历函数的使用

二、进行重点知识的讲解

(1) 教师通过 6.9.1 的案例，讲述递归函数的使用。

(2) 教师通过画图工具，帮助学生分析递归函数的执行过程。

(3) 什么是匿名函数？

简单来说，匿名函数就是没有名称的函数，也就是不再使用 `def` 语句定义的函数。如果要声明匿名函数，则需要使用 `lambda` 关键字。

(4) 教师通过 6.9.2 的案例，讲述匿名函数的使用。

(5) 教师通过 6.10.1 的案例，讲述在 Python 中表示时间的方式。

(6) 教师通过 6.10.2 的案例，讲述在 Python 中表示日历的方式。

(7) 教师根据课件，讲述 `calendar` 模块提供的内置函数，并且从中随机抽取几个函数，使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括匿名函数、递归函数、时间函数和日历函数。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第五课时

(随机数函数，案例-学生管理系统)

一、回顾上节课内容，继续介绍本课时的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，继续讲解本课时的知识。

在上节课中，我们介绍了一些关于时间和日历的内置函数，本课时再介绍一个比较常用的内置函数，并且带领大家完成一个案例，真正地在实际开发中应用函数的知识。

(3) 明确学习目标

- 要求学生掌握随机数函数的使用
- 要求学生在实际开发中熟练地运用函数

二、进行重点知识的讲解

(1) 教师通过 6.11 的案例，讲述 `random` 模块中一些常见函数的使用。

(2) 教师以学生管理系统为例，讲解如何在实际开发中应用函数。

(3) 教师让学生在课堂上练习上述案例，并对编写过程中产生的问题进行一一解答。

三、归纳总结，布置作业

(1) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课随机数函数的使用。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第六课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为使用函数的参数）

形式：单独完成

题目：

定义一个用于计算的函数，请按照以下要求完成。

要求如下：

- 13) 函数有 3 个参数，operation 默认值为“+”，只能接收“-”、“*”和“/”符号；num1 和 num2 分别用于接收整数。
- 14) 使用 if-elif 语句判断，如果为“+”，返回 num1 和 num2 的和；如果为“-”，返回 num1 和 num2 的差，以此类推。
- 15) 注意，处理 num2 不为 0 的情况。

上机二：（考察知识点为局部变量和全局变量）

形式：单独完成

题目：

请按照以下要求操作。

要求如下：

- 14) 定义全局变量 count，并赋值为 10；
- 15) 定义函数 test，在函数内部有变量 count，赋值为 100，并输出了 count 的值；
- 16) 调用函数，查看 count 的值；

	17) 在函数内部的 <code>count</code> 前面加上 <code>global</code>; 18) 再次调用函数, 查看 <code>count</code> 的值。 第七课时 (上机练习) 上机练习主要针对本章中需要重点掌握的知识点, 以及在程序中容易出错的内容进行练习, 通过上机练习可以考察同学对知识点的掌握情况, 对代码的熟练程度。 上机一: (考察知识点为匿名函数) 形式: 单独完成 题目: 使用匿名函数, 验证三条线是否能构成直角三角形。 要求如下: 10) 定义一个函数, 有两个接收线长度的参数 <code>a</code> 和 <code>b</code>, 返回一个匿名函数。 11) 判断选择最短的两条线; 12) 匿名函数用于计算这两个短边的平方和; 13) 如果函数返回的结果跟第三条边的平方相等, 表示是直角三角形。 上机二: (考察知识点为随机数函数) 形式: 单独完成 题目: 请按照以下要求操作。 要求如下: 7) 导入 <code>random</code> 模块; 8) 生成 0~1 的随机浮点数, 打印输出; 9) 生成 1~100 的随机浮点数, 打印输出; 10) 生成 1~100 的随机整数, 打印输出; 11) 生成 1~10 的偶数, 并打印输出;
思考题 和习题	见教材第 6 章配套的习题

教 学 后 记	
------------------	--

《Python 语言程序设计》 教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 7 章 高级函数	计划学时	3 课时
内容分析	上一个章节介绍了函数的基本内容，本章将带领大家继续深入地学习函数，包括闭包、装饰器、内置函数。		
教学目标及基本要求	23、掌握闭包的使用 24、理解装饰器的概念，会装饰函数 25、掌握常见内置函数的使用		
教学重点	19、掌握闭包的使用 20、理解装饰器的概念，会装饰函数 21、掌握常见内置函数的使用		
教学难点	12、掌握闭包的使用 13、理解装饰器的概念，会装饰函数		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	第一课时 （闭包，装饰器（部分）） 一、创设情境，导入闭包的概念 （11） 教师通过嵌套函数，引出闭包的概念。 Python 支持使用嵌套函数。如果在一个内部函数中对外部函数作用域（非全局作用域）的变量进行引用，那么内部函数就会被称为闭包。 （12） 明确学习目标 ● 要求学生掌握闭包的使用 ● 要求学生掌握装饰器的使用		

二、进行重点知识的讲解

(35) 闭包有哪些特点？

存在于两个嵌套关系的函数中，并且闭包是内部函数；

函数的内部引用了外部函数的变量；

嵌套的外部函数会将内部函数的名称作为返回值进行返回。

(36) 教师通过 7.1 的案例，讲解闭包是如何使用的。

(37) 教师使用画图工具，分析案例执行的流程，从而更加深入地理解闭包。

(38) 什么是装饰器？

装饰器本质上是一个 Python 函数，它可以在不改动其他函数的前提下，对函数的功能进行扩充。

(39) 装饰器可以应用在哪些场景？

引入日志；

函数执行时间统计；

执行函数前预备处理；

执行函数后清理功能；

权限校验；

缓存。

(40) 教师通过 7.2.1 的案例，讲解如何使用装饰器扩充函数的功能。

(41) 教师通过使用画图工具，分析 7.2.1 案例执行的具体流程。

(42) 教师通过 7.2.2 的案例，讲述多个装饰器的使用。

(43) 教师通过使用画图工具，分析 7.2.2 案例具体执行的过程。

三、归纳总结，布置作业/随堂练习

(9) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括闭包、装饰器的概念，装饰器的简单使用以及多个装饰器的使用。

(10) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(11) 使用博学谷系统下发课后作业。

第二课时

（装饰器（部分），Python 内置函数）

一、回顾上节课的内容，继续讲解本课时的知识

（14） 教师对学生们的疑问进行统一答疑。

（15） 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了装饰器。现在，如果装饰器装饰的函数有参数，那么要怎样使用呢？接下来，将针对装饰有参数的函数进行介绍。

（16） 明确学习目标

- 要求学生知道函数的 4 种类型
- 要求学生会在应用中选择适合的函数

二、进行重点知识的讲解

（28） 教师通过 7.2.3 的案例，讲述装饰器如何有参数的函数进行装饰。

（29） 教师使用画图工具，分析 7.2.3 案例执行的具体流程，帮助学生更好地理解。

（30） 教师通过 7.2.4 的案例，讲述装饰器如何对带有返回值的函数进行装饰。

（31） 教师使用画图工具，帮助学生分析案例执行的具体流程，以得到更好地理解。

（32） 教师通过 7.2.5 的案例，讲述如何使用带有参数的装饰器。

（33） 教师通过 7.3.1 的示例，讲述 map 函数的使用。

（34） 教师通过 7.3.2 的案例，讲述 filter 函数的使用。

（35） 教师通过 7.3.3 的案例，讲述 reduce 函数的使用。

三、归纳总结，布置作业

（5） 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括装饰器装饰有参数和返回值的函数、带参数的装饰器，以及常见的一些与序列相关的内置函数。

（6） 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

（7） 使用博学谷系统下发课后作业。

	第三课时 (上机练习) 上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。 上机一：（考察知识点为装饰有参数的函数） 形式：单独完成 题目： 请按照以下要求完成。 要求如下： 16) 定义一个带有两个参数 a 和 b 的函数 foo，返回 a 和 b 的和； 17) 定义一个装饰器函数 timefun，用于获取函数调用的时间； 18) 使用@符号修饰 foo 函数，让装饰器生效； 19) 调用 foo 函数，传入任意两个数值。 上机二：（考察知识点为带参数的装饰器） 形式：单独完成 题目： 请按照以下要求操作。 要求如下： 19) 定义一个没有参数的函数 foo，用于输出语句； 20) 定义一个装饰器函数 timefun_arg，它带有一个 pre 参数，用于输出函数的名称、调用时间和参数的值； 21) 使用@符号跟装饰器的名称修饰 foo 函数，让两个函数产生联系； 22) 调用 foo 函数。 上机三：（考察知识点为 filter 函数） 形式：单独完成 题目： 有一个列表，其内部包含任意 1~100 之间的整数若干个，筛选出不能够被 3 整除的数。
思考题 和习题	见教材第 7 章配套的习题

教 学 后 记	
------------------	--

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 8 章 Python 文件操作	计划学时	5 课时
内容分析	俗话说“好记性不如烂笔头”，不仅人的大脑会遗忘事情，计算机也会如此。比如一个程序在运行过程中用了九牛二虎之力终于计算出了结果，试想一下如果不把这些数据存放起来，下次重启电脑之后，“哭都没地方哭了”。可见，在把数据存储起来有这么大的价值。文件就是用于存储数据的，它可以让程序下次执行的时候，直接使用存储的数据，而不必重新制作一份，省时省力。		
教学目标及基本要求	26、掌握文件的打开和关闭 27、掌握文件的不同操作，例如，读写、重命名、删除		
教学重点	1、掌握文件的打开和关闭 2、掌握文件的不同操作，例如，读写、重命名、删除		
教学难点	1、掌握文件的不同操作，例如，读写、重命名、删除		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	第一课时 （文件的打开和关闭，写文件，读文件） 一、创设情境，导入函数的概念 （13） 教师通过提问学生问题，引出文件的打开和关闭操作。 例如问题是：使用 Word 软件编写一份简历，包括哪些流程呢？并根据学生的回答进行总结，从而引出文件操作的流程，包括打开、读写数据、关闭。本课时接下来针对文件的打开和关闭进行介绍。 （14） 明确学习目标 ● 要求学生会打开和关闭文件 ● 要求学生会读写文件		

二、进行重点知识的讲解

(44) 教师根据课件，讲述打开文件的方式，并通过示例代码演示。

在 python 中，open 方法用来打开文件，“文件名”必须要填写，“访问模式”是可选的，在不写模式时必须保证文件时存在的，否则会出现异常。

(45) 教师根据课件，讲述文件的访问模式。

如果条件允许的情况下，教师可以随机抽取几个比较重要的模式，通过示例代码进行演示。

(46) 教师根据课件，讲述打开文件的方式，并通过示例代码演示。

凡是打开的文件，切记要使用 close 方法关闭文件。

(47) 教师通过 8.2.1 的案例，讲述如何向文件中写数据。

向文件写数据，需要使用 write 方法来完成，在操作某个文件时，每调用一次 write 方法，写入的数据就会追加到文件末尾。

(48) 教师通过案例，讲述如何使用 read 方法读取文件的数据。

read(num) 可以从文件中读取数据，num 表示要从文件中读取的数据的长度（单位是字节），如果没有传入 num，那么就表示读取文件中所有的数据。

(49) 教师通过案例，讲述如何使用 readlines 方法读取文件的数据。

如果文件不是很大，那么可以使用 readlines 方法把整个文件中的内容进行一次性读取，并且返回的是一个列表，列表中的每一个元素为文件中的每一行数据。

(50) 教师通过案例，讲述如何使用 readline 方法一行一行读取文件的数据。

三、归纳总结，布置作业/随堂练习

(12) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括文件的打开和关闭，写文件和读文件。

(13) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(14) 使用博学谷系统下发课后作业。

第二课时

(应用-制作备份，文件定位读写，文件的重命名和删除，文件夹相关操作)

一、回顾上节课的内容，继续讲解本课时的知识

(17) 教师对学生们的疑问进行统一答疑。

(18) 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了文件的读写操作，接下来，本节课主要围绕上节课的内容，完成一个用于制作文件的备份的应用。

(19) 明确学习目标

- 要求学生掌握文件的定位读写
- 要求学生会重命名和删除文件
- 要求学生掌握文件夹的相关操作

二、进行重点知识的讲解

(36) 教师以制作文件备份应用为例，讲解如何在文件中读取和写入数据。

(37) 通过问题所产生的需求，引出文件的定位读写。

例如，打开某个文件，要求从某个位置开始写入数据。为此，我们需要对文件的读写位置进行定位，先获取当前读写的位置，再对文件进行读写操作。

(38) 教师通过案例，讲述使用 `tell` 方法获取文件当前的读写位置。

(39) 教师通过案例，讲述使用 `seek` 方法定位到文件的指定读写位置。

(40) 文件重命名要用到哪个方法？

`os` 模块中的 `rename()` 方法可以完成对文件的重命名操作。

(41) 教师通过示例代码，讲述文件的重命名操作。

(42) 删除文件要用到哪个方法？

`os` 模块中的 `remove()` 方法可以完成对文件的删除操作。

(43) 教师通过示例代码，讲述文件的删除操作。

(44) 教师根据课件，讲述与文件夹相关的操作，并使用示例程序演示操作。

三、归纳总结，布置作业

(6) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括文件的定位读写、重命名、删除，以及文件夹的相关操作。

(7) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(8) 使用博学谷系统下发课后作业。

第三课时

(应用-批量修改文件名，案例-学生管理系统)

一、回顾上节课内容，继续介绍本课时的内容

(14) 教师对学生们的疑问进行统一答疑。

(15) 回顾上节课的知识，继续讲解本节课的内容。

关于文件和文件夹的操作，我们都已经有个大概的了解。接下来，本课时通过两个文件操作的应用，带领大家巩固前面所讲的内容。

(16) 明确学习目标

- 要求学生会应用文件的操作

二、进行重点知识的讲解

(33) 教师以批量修改文件名应用为例，讲解如何在实例演练中使用文件的相关操作。

(34) 教师安排学生在课堂上练习上述应用，让学生亲自手动编写案例代码，并对操作过程中产生的问题进行一一解答。

(35) 教师以学生管理系统文件版为例，讲解如何在实际开发中把数据保存到文件中。

(36) 教师安排学生在课堂上练习上述应用，让学生亲自手动编写案例代码，并对操作过程中产生的问题进行一一解答。

三、归纳总结，布置作业

(5) 回顾学习目标，对本节课的内容进行总结。

(6) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为文件的定位读写，文件的重命名）

形式：单独完成

题目：

已知有名为 data.txt 的文件，其内容为“你好，欢迎来到 itheima”。请按照以下要求完成。

要求如下：

- 20) 打开 data.txt 文件，访问模式为 r+（可读写）；
- 21) 找到 itheima 所在的位置，在前面插入“黑马”；
- 22) 删除 itheima 内容；
- 23) 保存文件，把文件重命名为“data-最终.txt”

上机二：（考察知识点为文件的重命名）

形式：单独完成

题目：

按照教材 8.5 应用的要求，完成批量修改文件名的操作。

第五课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为在开发中把数据保存到文件中）

形式：单独完成

题目：

按照教材 8.6 小节关于文件的案例-学生管理系统，完成文件保存数据的功能。

思考题 和习题	见教材第 8 章配套的习题
教 学 后 记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级： _____

授课学期： _____

教师姓名： 某某老师

年 月 日

课题名称	第 9 章 异常	计划学时	4 课时
内容分析	编写程序的时候，程序员通常需要辨别事件的正常和异常（非正常）情况，这类异常事件可能是错误，或者是不希望发生的情况。为了能够处理这些异常事件，可以在所有可能发生这类事件的地方都使用条件语句。但是，这样做可能既没有效率，又降低了程序的可读性，Python 的异常对象提供了强大的替代解决方案。		
教学目标及基本要求	28、理解异常的概念 29、掌握处理异常的几种方式 30、掌握 raise 和 assert 语句，会抛出自定义的异常 31、掌握 with 和 as 环境安装器的使用		
教学重点	3、理解异常的概念 4、掌握处理异常的几种方式 5、掌握 raise 和 assert 语句，会抛出自定义的异常 6、掌握 with 和 as 环境安装器的使用		
教学难点	2、掌握处理异常的几种方式 3、掌握 raise 和 assert 语句，会抛出自定义的异常 4、掌握 with 和 as 环境安装器的使用		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	第一课时 （异常简介，异常类，异常处理（部分）） 一、创设情境，导入异常的概念 （15） 教师根据课件的示例程序，引出异常的概念。 例如，打开一个不存在的文件 123.txt，会产生 FileNotFoundError 异常。在 Python 中，程序在执行的过程中产生的错误称为异常，比如打开不存在的文件等。 （16） 分析系统产生异常的默认格式，并进行简要介绍。		

异常主要由名称和描述信息组成。在程序中遇到如打开不存在文件的问题，如果没有进行任何处理，程序就会终止。

(17) 明确学习目标

- 要求学生理解异常和异常类的概念
- 要求学生能处理异常

二、进行重点知识的讲解

(51) 什么是异常类？

Python 中的所有异常都是基类 `Exception` 的成员，并且在 `exceptions` 模块中定义，例如 `NameError`。Python 已经自动把所有异常的名称放在内建命名空间中，所以不必再导入 `exceptions` 模块就能直接使用异常。

(52) 教师根据课件介绍常见的异常，并且使用示例演示操作。

(53) 怎么处理异常？

通常，Python 会向用户反馈错误信息，但是遇到错误就终止程序。统为了继续程序的运行，并且告知用户遇到的问题，应该使用异常处理语句 `try-except` 捕获和处理。

(54) 教师根据课件，讲述如何捕获简单的异常，并且通过案例演示操作。

注意，只要监控到错误，程序就会执行 `except` 里面的语句，并且不再执行 `try` 里面未执行的语句。

(55) 教师根据课件，讲述如何捕获多个异常，并且通过案例演示操作。

(56) 教师根据课件，讲述如何捕获异常的描述信息，并且通过案例演示操作。

三、归纳总结，布置作业/随堂练习

(15) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括异常、异常类，以及异常的处理。

(16) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(17) 使用博学谷系统下发课后作业。

第二课时

(异常处理（部分），Python 处理异常的完整格式)

一、回顾上节课的内容，继续讲解本课时的知识

(20) 教师对学生们的疑问进行统一答疑。

(21) 回顾总结上节课内容，继续介绍本课时的内容。

上节课介绍了处理异常的几种方式，接下来，本节课继续介绍处理异常的知识。

(22) 明确学习目标

- 要求学生会处理异常

二、进行重点知识的讲解

(45) 教师根据课件，讲述如何捕获所有的异常。

(46) 教师根据课件，讲述程序没有捕获到异常的情况。

(47) 教师根据课件，讲述异常的终止行为。

(48) 教师根据多学一招，讲述 Python 中处理异常的完整语句。

(49) 教师随机从题库中抽取与本节课相关的题目，让学生随堂练习，并且对学生出现的问题进行解答。

三、归纳总结，布置作业

(7) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括捕获所有的异常、没有捕获到异常、异常的终止行为，以及处理异常的完整语句。

(8) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(9) 使用博学谷系统下发课后作业。

第三课时

(抛出异常，自定义异常，with 和 as 环境安装器)

一、回顾上节课内容，继续介绍本课时的内容

(17) 教师对学生们的疑问进行统一答疑。

(18) 回顾上节课的知识，继续讲解本节课的内容。

上节课介绍了捕获并且处理系统产生的异常的内容，接下来，本节课将针对抛出异常的知识进行讲解。

(19) 明确学习目标

- 要求学生抛出异常
- 要求学生自定义异常
- 要求学生掌握预定义清理的使用

二、进行重点知识的讲解

(37) 如何抛出异常？

要想在程序中主动抛出异常，可以使用 `raise` 和 `assert` 语句主动引发异常。

(38) 教师根据课件，讲述 `raise` 语句的格式和使用技巧。

(39) 教师根据课件，讲述 `assert` 语句的格式和使用技巧。

(40) 教师根据课件，讲述自定义异常。

(41) 什么是 `with` 语句？

`with` 语句适用于对资源进行访问的场合，确保不管使用过程中是否发生异常都会执行必要的“清理”操作，释放资源。

(42) 教师根据课件，介绍 `with` 语句的格式。

(43) 教师根据课件，介绍上下文管理器的相关概念。

(44) 教师根据课件，讲述 `with` 语句的整个执行过程。

三、归纳总结，布置作业

(6) 回顾学习目标，对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括抛出异常、自定义异常和 `with` 语句。

(7) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时 (上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为异常的处理）

	形式：单独完成 题目： 请按照以下要求完成。 要求如下： 24) 打开文件 123.txt，使用 while True 语句读取文件的数据，直到读完为止； 25) 使用 try 语句检测上述行为； 26) 使用 except 语句捕获所有的异常，并获取异常描述的具体信息； 27) 使用 else 语句处理没有异常的情况。 28) 使用 finally 语句处理释放资源的操作，如关闭文件。 上机二：（考察知识点为抛出自定义异常） 形式：单独完成 题目： 按照教材 9.5 案例的要求，完成抛出自定义异常的功能。
思考题和习题	见教材第 9 章配套的习题
教学后记	

《Python 语言程序设计》

教学设计

课程名称: Python 语言程序设计

授课年级: _____

授课学期: _____

教师姓名: _____

年 月 日

课题名称	第 10 章 Python 模块	计划学时	3 课时
内容分析	有过 C 语言编程经验的朋友都知道, 在 C 语言中如果要引用 <code>sqrt</code> 函数, 必须用语句 <code>#include <math.h></code> 引入 <code>math.h</code> 这个头文件, 否则是无法正常进行调用的。 那么在 Python 中, 如果要引用一些其他的函数, 该怎么处理呢? 在 Python 中有一个概念叫做模块 (module), 这个和 C 语言中的头文件以及 Java 中的包很类似, 比如在 Python 中要调用 <code>sqrt</code> 函数, 必须用 <code>import</code> 关键字引入 <code>math</code> 这个模块。		

教学目标及基本要求	32、 掌握模块的使用 33、 掌握模块的制作 34、 掌握包的使用 35、 了解模块的发布和安装
教学重点	7、 掌握模块的使用 8、 掌握模块的制作 9、 掌握包的使用
教学难点	5、 掌握模块的制作 6、 掌握包的使用
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （模块的基本使用，模块的制作，Python 中的包） 一、创设情境，导入模块的概念 （18） 教师通过提问学生问题，引出模块的概念。 例如问题是：前面使用随机数函数时，要如何才能使用这些函数？教师根据学生的回答进行总结：在调用函数时，必须引入模块，它和 C 语言中的头文件很类似。接下来，本课时就围绕着模块的内容进行详细讲解。 （19） 明确学习目标 ● 要求学生掌握模块的使用和制作 ● 要求学生会生成包 二、进行重点知识的讲解 （57） 教师根据课件，讲述导入模块的几种方式，并通过示例进行演示。 在 Python 中用关键字 <code>import</code> 来引入某个模块。有时候我们只需要用到模块中的某个函数，只需要引入该函数即可。 （58） 教师根据课件，讲述 Python 引入模块后搜索文件的路径。 Python 的搜索路径是由一系列目录名组成的，Python 解析器就依次从这些目录中寻找所引入的模块。搜索路径是在 Python 编译或安装的时候确定的，安装新的库应该也会修改。 （59） 教师根据课件，讲述模块的制作，以及 <code>__name__</code> 属性的使用。 每个模块都有一个 <code>__name__</code> 属性，当其值为 <code>'__main__'</code> 时，表明该模块自身在运行，否则是被引用。

(60) 教师根据课件, 讲述 Python 中的包。

包是 python 模块文件所在的目录, 且该目录下必须存在__init__.py 文件(文件内容可以为空)。

三、归纳总结, 布置作业/随堂练习

(18) 回顾上课前的学习目标, 并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点, 包括模块的基本使用、模块的制作, 以 Python 中的包。

(19) 布置随堂练习, 检查学生掌握情况。

根据博学谷和随堂练习资源, 给学生布置随堂练习, 检测学生的掌握程度, 并对学生出现的问题进行解决。

(20) 使用博学谷系统下发课后作业。

第二课时

(模块的发布, 模块的安装)

一、回顾上节课的内容, 继续讲解本课时的知识

(23) 教师对学生们的疑问进行统一答疑。

(24) 回顾总结上节课内容, 继续介绍本课时的内容。

上节课介绍了模块的使用和制作, 接下来, 本节课继续介绍模块的相关知识, 包括模块的发布和安装。

(25) 明确学习目标

- 要求学生掌握模块的发布
- 要求学生掌握模块的安装

二、进行重点知识的讲解

(50) 教师根据课件, 介绍模块发布的流程。

如果硬件条件允许的情况下, 教师可以通过实践演示相关操作。

(51) 教师根据课件, 介绍模块的安装步骤。

如果硬件条件允许的情况下, 教师可以通过实践演示相关操作。

三、归纳总结, 布置作业

(8) 回顾学习目标, 对本节课的内容进行总结。

教师总结本节课需要掌握的知识点, 包括模块的发布和安装。

(9) 布置随堂练习, 检查学生掌握情况。

	根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。 (10) 使用博学谷系统下发课后作业。 第三课时 (上机练习) 上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。 上机一：（考察知识点为模块的使用，模块的制作） 形式：单独完成 题目： 请按照以下要求完成。 要求如下： 29) 创建一个 Python 文件 test.py，在文件中有个计算两个数值和的函数； 30) 创建一个 Python 文件，取名为 main.py； 31) 在 main.py 文件中，导入 test 模块，调用 add 函数并打印结果； 32) 在 test.py 中使用__name__，测试调用 add 函数并打印结果； 33) 执行程序，检测程序输出的结果。 上机二：（考察知识点为模块的发布和安装） 形式：单独完成 题目： 按照教材 10.4 和 10.5 的要求，完成模块的发布和安装。
思考题和习题	见教材第 10 章配套的习题
教 学 后 记	

《Python 语言程序设计》 教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 11 章 Python 面向对象编程（上）	计划学时	5 学时
内容分析	面向对象编程是 Python 语言采用的基本编程思想，它把变量和函数组合在一起形成类，使得程序设计变得更加简单且有条理。在接下来的两个章节中，将为大家讲解如何使用面向对象的思想开发 Python 应用。		
教学目标及基本要求	5. 理解面向对象编程思想 6. 明确类和对象的关系，会独立设计类 7. 会使用类创建对象，并添加属性 8. 掌握构造方法和析构方法的使用 9. 熟悉 self 的使用技巧 10. 掌握运算符重载，会定制对象字符串的形式		
教学重点	3. 明确类和对象的关系，会独立设计类 4. 会使用类创建对象，并添加属性 5. 掌握构造方法和析构方法的使用 6. 熟悉 self 的使用技巧 7. 掌握运算符重载，会定制对象字符串的形式		
教学难点	3. 掌握构造方法和析构方法的使用 4. 熟悉 self 的使用技巧 5. 掌握运算符重载，会定制对象字符串的形式		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	第一课时 （面向对象编程概述，类和对象） 三、 创设情境，导入面向对象思想 （1）由面向过程，引出面向对象思想。 C 是一门面向过程语言，面向过程是分析解决问题所需要的步骤，然后用函数把这些步骤一一实现，使用的时候一个一个依次调用。例如：灯坏了，自己去买灯管、拆卸灯管、换灯管。而 Python 就是一门面向对象语言，它是通过对象解决问题的。如：		

灯坏了，找维修工修理。

(2) 进入主题，讲解面向对象和面向过程的区别？

面向过程关注的是解决问题需要哪些步骤；面向对象关注的是解决问题需要哪些对象。两种思想都能达到解决问题的目的，只是解决问题的思路不同。

(3) 明确学习目标

- 要求学生理解面向对象思想
- 要求学生掌握面向对象和面向过程的区别
- 要求学生掌握类和对象的使用

四、 进行重点知识的讲解

(1) 什么是面向对象编程？

面向对象编程(Object Oriented Programming-OOP) 是一种解决软件复用的设计和编程方法。这种方法会把软件系统中相似的操作逻辑和操作应用数据、状态以类的型式描述出来，以对象实例的形式在软件系统中复用，以达到提高软件开发效率的作用。

(2) 类和对象有什么关系？

类用于描述多个对象的共同特征，它是对象的模板。对象用于描述现实中的个体，它是类的实例。对象是根据类创建的，并且一个类可以对应多个对象。

(3) 教师根据课件，讲述如何定义类。

面向对象程序的设计把事物的特征和行为包含在类中。其中，事物的特征当做类的属性，事物的行为当做类的方法，而对象是类的一个实例。

(4) 教师根据课件，讲述类的组成和基本语法格式，并且通过示例进行演示。

(5) 教师通过案例，讲述如何根据定义类创建对象。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

带领学生总结本课内容：什么是面向对象、面相对象和面向过程的区别、类和对象，类的定义，以及根据类创建对象。

(2) 布置随堂练习，检查学生学习状况

根据博学谷和随堂练习资源，给学生布置随堂练习，根据学生的完成情况，了解学习的学习理解状况，根据学生反映的问题，进行解答。

(3) 使用博学谷系统下发课后作业

第二课时

（构造方法和析构方法，self 的使用）

三、 回顾上节课讲解的类和对象，导入构造方法的概念

（4） 教师讲解上一课时的课后作业。

（5） 教师通过程序潜在的问题，引出构造方法的概念。

例如，在 11.2.3 案例的基础上，再创建一个 Car 类对象，并且为其添加属性，这样每创建一个对象就要动态地添加属性是非常繁琐的。为了解决这个问题，让对象创建以前就有默认的属性，为此 Python 提供了一个构造方法，该方法的固定名称为__init__。

（3）明确学习目标

- 要求学生掌握构造方法和析构方法的使用
- 要求学生掌握 self 的使用

四、 进行重点知识的讲解

（1）怎么使用构造方法？

当创建类的实例的时候，系统会自动调用构造方法，从而实现对类进行初始化的操作。

（2）教师通过 Car 类的案例，讲述构造方法和带参数的构造方法。

（3）什么是析构方法？

当删除一个对象来释放类所占资源的时候，Python 解释器默认会调用另外一个方法，这个方法就是__del__()方法。

（4）教师通过 Person 类案例，讲述析构方法的使用。

（5）教师通过 Dog 类案例，讲述 self 关键字的使用。

三、归纳总结，布置课后作业

（1）回顾课前学习目标，对本节课的内容进行总结

带领学生总结本课内容：构造方法、析构方法，以及 self 的使用。

（2）布置随堂练习，检查学生学习状况

根据博学谷和随堂练习资源，给学生布置随堂练习，根据学生的完成情况，了解学习的学习理解状况，根据学生反映的问题，进行解答。

（3）使用博学谷系统下发课后作业

第三课时

(运算符的重载)

三、 回顾上节课讲解的构造方法，继续介绍本课时的运算符重载

(5) 教师讲解上一课时的课后作业。

(6) 回顾上节课所学习的知识，继续介绍本节课学习的内容。

上一节课中，我们学习了构造方法和析构方法，接下来，在本节课将学习运算符的重载。

(7) 明确学习目标，掌握运算符的重载。

四、 进行重点知识的讲解

(1) 什么是运算符重载？

运算符重载是通过实现特定的方法，使类的实例对象支持 Python 的各种内置操作。

(2) 教师通过 11.5.1 的案例，讲述加法运算重载。

(3) 教师通过 11.5.2 的案例，讲述索引和分片重载。

(4) 教师通过 11.5.3 的案例，讲述定制对象的字符串形式。

三、归纳总结，布置课后作业

(1) 回顾运算符的重载，对本节课的内容进行总结。

(2) 使用博学谷系统下发课后作业。

第四课时

(反恐精英 CS)

一、 回顾本章讲解的全部内容，介绍本课时讲解的案例

(1) 教师讲解上一课时的课后作业。

(2) 回顾前面所学习的知识，继续讲解本课时要完成的案例。

本课时主要围绕着本章所学的面向对象的知识，描述一个射击游戏的场景。

(3) 明确学习目标

- 要求学生掌握类和对象的定义和使用
- 要求学生掌握方法的使用

二、 进行重点知识的讲解

- (1) 教师根据课件，先讲述案例的需求和具体流程。
- (2) 教师根据案例的需求，分析需要设计的类。
- (3) 教师根据案例的分析，讲解实现案例的具体步骤。
- (4) 教师让学生在课堂上练习案例，并针对学生出现的问题进行解决。

三、归纳总结，布置课后作业

- (1) 回顾本节课介绍的案例。对本节课的内容进行总结
- (2) 使用博学谷系统下发课后作业

第五课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

形式：独立完成

上机一：（考察知识点为类和对象的创建、添加属性、定义方法）

题目：

请按照以下要求设计一个 Car 类。

要求如下：

- 3) Car 类中声明两个属性 price 和 speed，分别表示汽车的价格和最高时速；
- 4) Car 类中声明一个方法 run，表示汽车行驶的行为，在方法中要求访问 speed 和 price。
- 5) 创建一个 Car 类对象，调用 run 方法输出。

上机二：（考察知识点为重载__str__方法）

题目：

请按照以下要求设计一个 Demo 类。

要求如下：

- 1) 定义一个 Demo 类，声明一个 data1 属性赋值为 100；

	2) 定义 <code>set</code> 方法，该方法接收一个 <code>num</code> 值，它会赋值给 <code>data2</code> 属性； 3) 重载 <code>__str__</code> 方法返回自定义的字符串，即打印 <code>data1</code> 和 <code>data2</code> 的值； 4) 创建一个 <code>Demo</code> 类实例 <code>demo</code>，调用 <code>set</code> 方法给 <code>data2</code> 赋值为 200； 5) 分别使用 <code>print()</code>、<code>str()</code>、<code>repr()</code> 函数输出 <code>demo</code> 的信息。 上机三：（考察知识点为反恐精英案例） 题目： 请按照教材的要求完成反恐精英案例。
考题和习题	见教材第 11 章配套的习题
教学后记	

《Python 语言程序设计》

教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：

教师姓名： 某某老师

年 月 日

课题名称	第 12 章 Python 面向对象编程（下）	计划学时	3 学时
内容分析	在上一章节中，我们了解了面向对象的基本知识。在接下来的本章中，将继续围绕着面向对象进行深入地介绍，包括面向对象的三大特性（封装、继承和多态）、类属性和实例属性、静态方法和类方法。		

教学目标及基本要求	11. 理解如何利用封装保护属性 12. 掌握单继承和多继承，会重写和调用父类方法 13. 理解多态的使用 14. 掌握类属性和实例属性 15. 熟悉类方法和静态方法的使用技巧
教学重点	8. 理解如何利用封装保护属性 9. 掌握单继承和多继承，会重写和调用父类方法 10. 理解多态的使用 11. 掌握类属性和实例属性 12. 熟悉类方法和静态方法的使用技巧
教学难点	6. 掌握单继承和多继承，会重写和调用父类方法 7. 掌握类属性和实例属性 8. 熟悉类方法和静态方法的使用技巧
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解
教学过程	第一课时 （封装、继承） 五、 创设情境，导入封装的思想 （1）教师通过 Person 示例，讲述公有属性存在的潜在风险，从而引出封装属性的内容。 例如，给 Person 的 age 属性赋值为 200。从实际生活的角度来看，人的年龄显然不符合需求。为了避免外界随意赋值，需要隐藏数据和保护属性。 （2）通过什么方式保护类的属性？ 把属性定义为私有属性，即在属性名的前面加上两个下划线；添加用于设置或者获取属性值的两个方法供外界调用。 （3）明确学习目标 ● 要求学生掌握封装和继承 六、 进行重点知识的讲解 （1）教师通过 Person 案例，讲述使用私有属性隐藏数据。 （2）教师对上述案例进行总结，讲述私有属性和公有属性的区别。

Python 中没有任何关键字来区分公有属性和私有属性，它是以属性命名的方法进行区分的，如果属性名的前面加了两个下划线，就表明该属性是私有属性，否则就是公有属性。

(3) 什么是单继承？

列举生活中的继承，再区分编程中的继承。例如：生活中，继承一般指的是子女继承父辈的财产。在程序中，继承是一种关系，一个类通过继承可以拥有被继承类的所有功能。例如：汽车类有轮胎个数的属性和行驶功能。当轿车继承汽车类后，轿车也拥有了轮胎个数的属性和行驶功能。

(4) 教师通过 Cat 类案例，围绕着继承的基本语法，讲述单继承的使用。

(5) 教师通过 Animal 类案例，讲述继承的注意事项。

父类的私有属性和方法是不会被子类继承的，更不能被子类访问。

(6) 什么是多继承？

例如，水鸟既有鸟的特点，能在天空飞翔，又能像鱼一样在水里遨游。Python 是支持多继承的，多继承指的是子类拥有多个父类，并且具有它们共同的特性，即子类继承了父类的方法和属性。

(7) 教师通过 Bird 类案例，围绕着多继承的基本语法，讲述多继承的使用。

(8) 在上述案例的基础上，在两个父类中添加相同的方法，讲述多个父类出现同名方法的情况。

(9) 教师通过 12.2.3 的案例，讲述如何重写父类方法，以及调用父类方法。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

带领学生总结本课内容：封装（保护属性）、单继承、多继承，以及重写父类的方法与调用父类方法。

(2) 使用博学谷系统下发课后作业

第二课时

（多态，类属性和实例属性，类方法和静态方法）

五、回顾上节课讲解的面向对象，导入多态的概念

(6) 教师讲解上一课时的课后作业。

(7) 教师根据课件提供的示例代码，引出多态。

调用同一个方法，出现了两种表现形式，这个过程体现的就是多态。

(8) 与强类型语言相比，介绍什么是多态？

在 Python 中，多态指的是在不考虑对象类型的情况下使用对象。相比于强类型的语言，Python 更推崇“鸭子类型”。“鸭子类型”是这样推断的：如果一只生物走起路来像鸭子，叫起来也像鸭子，那么它就可以被当做鸭子。

(9) 明确学习目标

- 要使学生了解多态
- 要使学生掌握类属性和实例属性
- 要求学生掌握类方法和实例方法

六、 进行重点知识的讲解

(1) 教师通过 Animal 类的案例，讲述多态的使用。

(2) 什么是类属性？什么是实例属性？

前面我们所接触到的属性都是实例属性，通过“实例.属性”添加属性。类属性是类所拥有的属性，它需要在类中进行显式地定义（位于类内部，方法的外面），它被所有类的实例对象所共有，在内存中只存在一个副本。

(3) 教师通过 Cat 类的案例，讲述类属性和实例属性的使用。

如果硬件条件允许的情况下，教师可以通过实践演示操作。

(4) 什么是类方法？

类方法是类所拥有的方法，它可以使用修饰器@classmethod 来标识为类方法。

(5) 教师通过 Test 类的案例，讲述类方法的使用。

(6) 什么是静态方法？

静态方法同样只属于定义它的类，可以使用修饰器@staticmethod 来标识为静态方法。

(7) 教师通过 Test 类的案例，讲述静态方法的使用。

(8) 教师总结类方法、实例方法及静态方法的区别，帮助学生理解并很好地选择运用它们。

如果要修改实例属性的值，就直接使用实例方法；如果要修改类属性的值，就直接使用类方法；如果是辅助功能，比如打印菜单，这时可以考虑使用静态方法，可以在

不创建对象的前提下使用。

三、归纳总结，布置课后作业

(1) 回顾课前学习目标，对本节课的内容进行总结

带领学生总结本课内容：多态、类属性和实例属性，以及类方法和静态方法。

(2) 布置随堂练习，检查学生学习状况

根据博学谷和随堂练习资源，给学生布置随堂练习，根据学生的完成情况，了解学习的学习理解状况，根据学生反映的问题，进行解答。

(3) 使用博学谷系统下发课后作业

第三课时

(上机练习)

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

形式：单独完成

上机一：（考察知识点为封装，保护属性）

题目：

请按照以下要求设计一个 **Person** 类。

要求如下：

- 6) **Person** 类中有带两个参数的构造方法，用于给 **name** 和 **age** 属性赋值。其中，**age** 属性是私有的；
- 7) **Person** 类中声明一个给 **age** 属性赋值的方法 **setAge**，只有传入的参数大于 0 且小于 120 就能赋值；
- 8) **Person** 类中声明一个获取 **age** 属性值的方法 **getAge**，直接返回 **age** 的值；
- 9) 重载 **__str__** 方法，用于打印输出 **Person** 类对象的属性信息。
- 10) 创建一个 **Person** 类的对象，使用 **print** 函数输出对象的信息。

上机二：（考察知识点为多继承）

题目：

请按照以下要求设计类。

要求如下：

- 6) 定义 Bird 类，在该类中声明 fly 方法，用于打印“鸟儿飞翔”的信息；
- 7) 定义 Fish 类，在该类中声明 swim 方法，用于打印“鱼儿游泳”的信息；
- 8) 定义 Volador 类继承自 Bird 和 Fish 类，在 Volador 类中没有任何属性和方法；
- 9) 创建一个 Volador 类的对象，依次调用 fly 和 swim 方法。
- 10) 在 Bird 和 Fish 类中增加 breathe 方法，分别用于打印“鸟儿呼吸”和“鱼儿呼吸”的信息；
- 11) 让刚刚创建的对象，调用 breathe 方法。

上机三：（考察知识点为重写父类方法）

题目：

请按照以下要求设计类。

要求如下：

- 1) 定义 Animal 类，在该类中有一个带有参数的构造方法，用于给 legNum 属性赋值；
- 2) 定义继承自 Animal 类的子类 Bird 类，重写父类的构造方法，在构造方法中添加 plume 属性，并赋值为“白色”；
- 3) 创建 Bird 类的对象，输出“有一只白色的鸟儿在树上唱歌。”

上机四：（考察知识点为调用父类方法）

题目：

请按照以下要求设计类。

要求如下：

- 1) 在上机三的基础上进行修改；
- 2) 在 Bird 类的构造方法中调用父类的构造方法；
- 3) 创建 Bird 类的对象后，输出“有一只 2 条腿白色的鸟儿站在树上唱歌。”

考题和习题	见教材第 12 章配套的习题
教学 后 记	

《Python 语言程序设计》 教学设计

课程名称： Python 语言程序设计

授课年级：

授课学期：_____

教师姓名： 某某老师

年 月 日

课题名称	第 13 章 用 Python 玩微信	计划学时	5 学时
内容分析	微信 是一款非常火爆的手机交流工具。本章将带领大家开发一个 微信软件 ，让大家更加深入地理解如何在实际应用中使用面向对象编程。		
教学目标及基本要求	16. 会在 PyCharm 工程中添加第三方框架 17. 掌握 pygame 框架的基本使用 18. 理解面向对象编程思想 19. 会基于面向对象的思想编写程序 20. 明确类和对象的关系，会独立设计类		
教学重点	13. 会在 PyCharm 工程中添加第三方框架 14. 掌握 pygame 框架的基本使用 15. 理解面向对象编程思想 16. 会基于面向对象的思想编写程序 17. 明确类和对象的关系，会独立设计类		
教学难点	9. 掌握 pygame 框架的基本使用 10. 会基于面向对象的思想编写程序		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		

教 学 过 程	第一课时 （游戏介绍，分析游戏显示的过程，添加 pygame 模块，搭建游戏界面） 七、 创设情境，引出飞机大战游戏。 （1）教师运行飞机大战游戏项目，向学生展示本章要介绍的项目。 运行项目的过程中提问学生问题。例如，这个游戏大家认识吗？游戏的规则有哪些？教师根据学生的回答进行总结，从而引出飞机大战项目。 （2）教师介绍本章要完成的功能。 主要包括：玩家飞机可以左右方向移动以躲避子弹；玩家飞机可以发射子弹；敌方飞机位于屏幕的顶部保持左右移动，并且随机地向玩家飞机发射子弹。 （3）明确学习目标 ● 要求学生了解要完成的游戏功能 ● 要求学生会在 PyCharm 中添加 pygame 模块 ● 要求学生会搭建游戏界面 八、 进行重点知识的讲解 （1）教师通过展示的游戏界面，分析实现整个项目的步骤。 （2）教师根据课件的内容，讲述如何导入第三方模块 pygame。 （3）教师在创建的项目中导入图片素材。 （4）教师导入 pygame 模块，创建窗口并显示背景图片。 三、归纳总结，布置课后作业 （1）回顾上课前的学习目标，对本节课知识点进行总结。 带领学生总结本课内容：在 PyCharm 中添加 pygame 模块，搭建游戏界面。 （2）布置课后作业，让学生课下练习本节课所编写的程序。 第二课时 （检查键盘输入，显示玩家飞机并控制移动，显示玩家飞机的子弹） 七、 回顾上节课讲解的内容，继续讲述本课时的内容 （10） 回顾上节课介绍的知识，继续讲解本课时要完成的游戏功能。 在上节课中，我们在工程中添加了第三方模块，搭建了游戏的背景。本节课继续完成游戏的功能，包括检测键盘的输入、玩家飞机的显示、移动和发射子弹。
---	--

(11) 明确学习目标

- 要求学生会检查键盘的输入
- 要求学生会显示飞机，并控制飞机的移动

八、 进行重点知识的讲解

- (1) 对比游戏运行的效果，教师根据课件讲述检查键盘的输入。
- (2) 对比游戏运行的效果，教师根据课件在界面上显示玩家飞机，并且控制飞机移动的方向。
- (3) 对比游戏运行的效果，教师带领学生一起分析显示子弹的过程。
- (4) 教师根据课件，讲述如何在玩家飞机的头部显示子弹图片。

三、归纳总结，布置课后作业

- (1) 回顾课前学习目标，对本节课的内容进行总结

带领学生总结本课内容：检查键盘键入，显示玩家飞机并控制移动，以及显示玩家飞机的子弹。

- (2) 布置课后作业，让学生课下练习本节课所编写的程序。

第三课时

(子弹移动，显示敌人飞机，控制敌人飞机移动)

五、 回顾上节课讲解的内容，继续讲述本课时内容

- (8) 回顾上节课介绍的知识，继续讲解本课时要完成的游戏功能。

在上节课中，我们在工程中添加了检查了键盘的输入，显示了玩家飞机和子弹，并且能控制玩家飞机左右移动。本节课继续完成游戏的功能，包括子弹的移动，敌人飞机的显示和移动。

- (9) 明确学习目标

- 要求学生会控制子弹的移动
- 要求学生会显示敌人飞机，并控制敌人飞机移动

六、 进行重点知识的讲解

- (1) 如何让子弹向上移动？

每执行一次 `while` 循环，就会调用一次 `display()` 方法，让子弹再次显示到屏幕上。由于屏幕刷新的速度特别快，使用肉眼是无法捕捉到的。利用程序的这个特点，每刷新

一次屏幕，就让子弹显示的位置上移几个单位，从而形成向上发射子弹的动画效果。

- (2) 对比游戏运行的效果，教师根据课件让子弹向上移动。
- (3) 按照显示玩家飞机的步骤，在界面上显示敌人飞机。
- (4) 对比游戏运行的效果，教师根据课件讲述如何控制敌人的飞机移动。

三、归纳总结，布置课后作业

- (1) 回顾本课时实现的游戏功能，对本节课的内容进行总结
- (2) 使用博学谷系统下发课后作业

第四课时

(敌人飞机发射子弹，抽取子弹基类，抽取飞机基类)

三、 回顾上节课讲解的内容，继续讲述本课时的内容

- (4) 回顾上节课介绍的知识，继续讲解本课时要完成的游戏功能。

在上节课中，我们让子弹向上发射，在界面上显示了敌人飞机，并且控制敌人飞机的循环运动。本节课将继续完成敌人飞机发射子弹的功能，从整体上看，项目代码的功能很相似，代码结构不够分明，另外对程序进行一些优化。

- (5) 明确学习目标
 - 要求学生掌握继承的使用

四、 进行重点知识的讲解

- (1) 教师根据课件，讲述敌人飞机发射子弹的功能。
- (2) 教师总结现有程序存在的缺陷，从而引出优化程序的需求。

在前面的程序中，我们发现玩家飞机类和敌人飞机类的功能很相似，代表子弹的两个类的功能也极其相似,这种代码显得每个类的结构不够分明,同时又增加了代码量。本节主要利用继承对上述完成的程序进行优化，减少没有必要的代码。

- (3) 教师根据课件，讲述如何抽取子弹基类。
- (4) 教师根据课件，讲述如何抽取飞机基类。

三、归纳总结，布置课后作业

- (1) 回顾本课时实现的游戏功能，对本节课的内容进行总结
- (2) 布置随堂练习，检查学生学习状况

根据博学谷和随堂练习资源，给学生布置随堂练习，根据学生的完成情况，了解学

	习的学习理解状况，根据学生反映的问题，进行解答。 (3) 使用博学谷系统下发课后作业 第五课时 (上机练习) 上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。 形式：独立完成 上机一：（考察知识点为基于面向对象的思想编写程序） 题目： 请按照第 13 章的内容，基于面向对象的思想，完成飞机大战游戏的相应功能。
考题和习题	见教材第 13 章配套的习题
教学后记	