|  |  |  |  |
| --- | --- | --- | --- |
| 核心程序设计—“冒险游戏模拟” | | 学时 | 支撑课程目标 |
| 主要内容 | 1. 流程控制语句的概念介绍：首先向学生解释什么是流程控制语句，包括循环，分支语句，它们在编程中的重要性以及主要作用。  2. if，else if和else的使用：讲解if，elif，else语句的使用方法和功能，如何使用elif来添加额外的条件判断，以及如何在前面的条件都不成立时执行else块中的代码，通过实际生活中的例子和文字冒险游戏中的不同选择来帮助学生理解。  3.switch-case的使用：讲解switch-case语句多分支选择的使用方法和功能，替代多个嵌套的if-else语句的作用，提高代码可读性。  4. while循环语句：介绍while循环的基本结构和用法，通过结合文字冒险游戏中的循环规则，结合生活实例，逐步讲解while循环运行的顺序。  5. for循环语句：讲解for循环语句的使用，包括如何遍历序列的每个元素，如何控制某个代码块运行的次数。  6.break,continue,goto 的使用：结合文字冒险游戏的情节讲解break,continue,goto的使用方法， | | |
| 学习目标 | 1.理解条件语句和循环语句的概念：学生能通过冒险游戏创设的情景了解条件语句是一种根据不同条件执行不同操作的语句结构，不同的选择有不同的游戏剧情得到不同的结果，而循环语句则是一种重复执行特定代码块直到满足特定条件的语句结构。  2.掌握if、elif、else的基本语法：学生能通过冒险游戏创设的情景学会使用if语句来判断一个条件是否满足，并执行相应的代码块；了解elif的作用，用于添加额外的条件判断；明白else的作用，处理不满足前面条件的选项。  3.能够灵活运用if...elif...else进行逻辑判断：学生能通过冒险游戏创设的情景应该能够根据具体问题，合理地设计条件语句的逻辑结构，实现不同的功能。  4.理解while，do……while和for循环的基本语法：学生能通过冒险游戏创设的情景应该学会使用while和for循环来重复执行特定的代码块，直到满足特定的条件。  5.理解break和continue：运用两者在for循环和while循环中实现循环的跳出或跳过某些条件继续运行。  7.实践应用：学生可以通过编写简单的程序，如判断数字大小、判断成绩等级等，将所学的条件语句和循环语句的知识应用到实际中，加深对知识的理解和记忆。  8.比较c++语言与其他语言的区别，比较c++的条件语句和循环语句与其他语言（如c语言）有什么相同之处和不同之处 | | |
| 学生课前阅读材料与其他准备 | 1.必读书目：《C++ Primer Plus》[美] 史蒂芬·普拉达；人民邮电出版社  2.课前思考问题：  ①假如你要构思一个文字冒险游戏，你会有什么奇思妙想？  ②由游戏情节所需，你觉得如何用编程实现？  ③当有条件判断时，你知道如何使用if else switch…case吗？  ④当遇到循环的游戏情节时，你是否知道如何使用while，for来实现循环，以及如何跳出循环?  ⑤你能想象一个情景，在这个情景中，使用条件语句和循环结构可以解决问题吗？ | | |
| 教学方式 | 1.理论介绍： 通过生动有趣的冒险游戏，深入解释各类流程控制语句的使用方法和功能，旨在以引人入胜的方式促进学生对新知识的迅速理解与吸收。  2.互动实践： 通过让学生运行文字冒险游戏的代码，实际操作加深对不同流程控制语句的理解。此实践旨在巩固理论知识，培养学生的编程实践能力。  3.角色扮演： 引入生动的角色扮演元素，学生可以充当魔法迷宫中的探险者，通过不同的选择触发不同的游戏剧情，可以旨在激发学生的兴趣，提高上课的专注度和学习效果。  4.提问巩固： 在课程末尾设计不同类型的问题，涉及不同流程控制结果的知识，以便学生能够在学习结束后通过回答问题来巩固并检验所获得的知识。这一环节旨在促使学生在实践中运用所学，进一步确保知识的深入理解。 | | |