

2024-2025 学年初中信息技术（信息科技） 八年级上粤高教 B 版（2019）教学设计合 集

目录

一、第一章 感受程序设计的魅力

1.1 第一节 初识计算机程序

1.2 第二节 体验程序设计

1.3 本单元复习与测试

二、第二章 程序设计入门

2.1 第一节 神奇故事制造机

2.2 第二节 用程序解九章算术

2.3 第三节 智能运动助手

2.4 第四节 自动售货机

2.5 第五节 飞花令

2.6 第六节 妙笔生花

2.7 第七节 项目活动：我是信息安全小助手

2.8 本单元复习与测试

三、第三章 程序设计进阶

3.1 第一节 词云图

3.2 第二节 图像检测

3.3 第三节 飞翔的小鸟

3.4 第四节 综合活动：聪明的小鸟

3.5 本单元复习与测试

第一章 感受程序设计的魅力第一节 初识计 算机程序

学 校		授课教师		课 时	
授课班级		授课地点		教 具	
教材分析	“初中信息技术（信息科技）八年级上粤高教 B 版（2019）第一章 感受程序设计的魅力第一节 初识计算机程序”主要介绍计算机程序的基本概念、程序设计的作用及意义，以及程序设计的基本步骤。本节课旨在帮助学生建立对计算机程序的基本认识，激发学生对程序设计的兴趣，为后续学习打下基础。教材内容深入浅出，通过实例引导学生理解程序设计的核心概念，符合八年级学生的认知水平。				
核心素养目标	培养学生信息意识，使其能够认识到程序设计在解决实际问题中的作用，提高利用信息技术解决问题的能力。通过本节课的学习，学生将能够理解程序设计的基本概念，掌握程序设计的基本步骤，培养逻辑思维和创新能力，为成为适应未来信息社会发展的数字公民奠定基础。				
学习者分析	1. 学生已经掌握了计算机的基本操作和简单的信息技术知识，对计算机有初步的认识，能够使用常见的软件进行基本操作。 2. 学生对信息技术充满好奇心，喜欢探索新事物，具备一定的自学能力和动手能力。他们的学习风格多样，有的喜欢通过实践操作学习，有的则偏好理论学习。 3. 学生可能遇到的困难和挑战包括：对程序设计概念的理解可能存在难度，初次接触编程语言可能会感到困惑；逻辑思维能力和问题解决能力有待提高；此外，由于编程需要耐心和细致，部分学生可能会因为遇到挫折而丧失学习兴趣。				
教学方法与手段	教学方法： 1. 讲授法：通过讲解计算机程序的基本概念和程序设计的步骤，使学生建立初步的理论框架。 2. 案例分析法：通过分析具体的程序设计案例，引导学生理解程序设计在实际问题中的应用。 3. 实践操作法：通过动手编写简单的程序，让学生在实践中掌握编程的基本技巧。 教学手段： 1. 多媒体教学：使用 PPT 展示程序设计的流程和关键概念，增强视觉效果，帮助学生理解。 2. 编程软件：利用编程软件进行实时演示，让学生直观感受程序运行的过程和效果。 3. 网络资源：引导学生利用网络资源进行自主学习，扩展知识面，提高学习效率。				
教学实施过程	1.				

	课前自主探索 教师活动： 发布预习任务：通过在线平台发布预习资料，包括教材第一章第一节的相关内容和辅助教学视频，要求学生预习并理解计算机程序的基本概念。 设计预习问题：设计问题如“程序设计在生活中的应用有哪些？”引导学生思考程序设计的重要性。 监控预习进度：通过在线平台查看学生预习情况，对未完成预习的学生进行提醒。 学生活动： 自主阅读预习资料：学生根据预习任务自主阅读相关内容，观看视频。 思考预习问题：学生思考预习问题，尝试用自己的语言回答。 提交预习成果：学生将预习笔记和问题答案提交至在线平台。 教学方法/手段/资源： 自主学习法：培养学生自主学习能力。 信息技术手段：利用在线平台进行资源分享和进度监控。 作用与目的： 帮助学生提前了解课程内容，为课堂学习打下基础。 2. 课中强化技能 教师活动： 导入新课：通过展示一个简单的程序实例，引导学生思考程序是如何解决问题的。 讲解知识点：详细讲解程序设计的基本步骤，如分析问题、设计算法、编写代码等。 组织课堂活动：分组讨论，让学生尝试设计一个简单的程序流程。 解答疑问：对学生提出的问题进行解答，确保学生对程序设计步骤的理解。 学生活动： 听讲并思考：学生认真听讲，对程序设计的基本步骤进行思考。 参与课堂活动：学生分组讨论，设计简单的程序流程，体验程序设计过程。 提问与讨论：学生针对不理解的部分提问，与同学和老师讨论。 教学方法/手段/资源： 讲授法：讲解程序设计的基本步骤。 实践活动法：通过小组讨论设计程序流程。 合作学习法：分组讨论，培养学生的团队合作能力。 作用与目的： 帮助学生掌握程序设计的基本步骤，理解程序设计的流程。 3.
--	---

	课后拓展应用 教师活动： 布置作业：布置编写一个简单程序的作业，如设计一个计算器程序。 提供拓展资源：提供在线编程平台和相关的编程教程，供学生进一步学习。 反馈作业情况：批改作业，对学生的代码进行评价和反馈。 学生活动： 完成作业：学生根据所学，尝试编写简单的程序代码。 拓展学习：利用提供的资源，进一步学习编程知识。 反思总结：学生反思编程过程中的困难，总结学习经验。 教学方法/手段/资源： 自主学习法：学生自主完成编程作业。 反思总结法：学生通过反思总结提升编程能力。 作用与目的： 巩固学生对程序设计的理解，通过实践提高编程能力。
学生学 习效果	1. 理解了计算机程序的基本概念：学生能够明确计算机程序是由一系列指令组成的，这些指令用于告诉计算机如何执行特定任务。他们了解了程序设计在解决实际问题中的重要作用。 2. 掌握了程序设计的基本步骤：学生学会了如何分析问题、设计算法、编写代码和测试程序。他们能够按照这些步骤，设计并实现简单的程序。 3. 提升了逻辑思维能力：在编写程序的过程中，学生需要逻辑清晰地思考问题，这有助于培养他们的逻辑思维能力。他们能够通过编程解决实际问题，提高了问题解决能力。 4. 增强了自主学习和探索能力：通过课前预习和课后拓展学习，学生养成了自主学习的习惯。他们能够利用网络资源和编程平台，自主探索更深入的编程知识。 5. 提高了团队合作和沟通能力：在课堂活动和课后作业中，学生需要与同学合作，共同完成编程任务。这有助于培养他们的团队合作精神和沟通能力。 6. 以下是一些具体的知识点和学习效果： - 学生能够描述程序设计的基本概念，如程序、算法、代码等。 - 学生能够解释程序设计的基本步骤，包括问题分析、算法设计、代码编写和程序测试。 - 学生能够使用简单的编程语言（如 Python 或 Scratch）编写简单的程序，如计算器、九九乘法表等。 - 学生能够理解变量、循环和条件语句等基本编程概念，并在程序中使用这些概念。 - 学生能够通过调试程序，找出并修复代码中的错误。 - 学生能够利用在线编程平台进行编程实践，如 Code.org、LeetCode 等。 - 学生能够通过编程解决实际问题，如设计一个简单的游戏、自动化数据处理等。 - 学生能够阅读和理解他人的代码，提高自己的编程能力。 - 学生能够结合数学知识，编写程序解决数学问题，如计算斐波那契数列、求解二次方程等。
教学反 思与改	这节课结束后，我进行了深入的反思，觉得在教学内容、教学方法以及学生学习效果等方面都有值得改进的地方。

进	在设计课程时，我尽力将理论与实践相结合，但回顾起来，我觉得案例的选择还可以更加贴近学生的生活，这样能更好地激发他们的兴趣。例如，我可以选择一个与学生生活紧密相关的程序设计案例，比如制作一个班级成绩管理系统，这样学生在学习过程中能够更加直观地感受到程序设计的实际应用。 在教学方法的运用上，我尝试了小组讨论和实践活动，但我也注意到，部分学生在小组活动中参与度不高，可能是因为他们对编程的兴趣不够浓厚，或者是害怕在同伴面前犯错。为了解决这个问题，我计划在未来的教学中，增加一些个性化的辅导，帮助这些学生建立信心。 1. 设计反思活动： 在课程结束时，我会设计一个简单的问卷，让学生填写他们对本节课内容的理解和掌握程度，以及他们对教学方法的看法。这样我可以及时了解学生的学习情况，发现教学中的不足。 2.
---	--

	制定改进措施： - 加强案例教学的针对性：选择更贴近学生生活的案例，让学生能够更好地理解和应用所学知识。 - 提供个性化辅导：对于参与度不高或学习有困难的学生，提供一对一的辅导，帮助他们克服学习障碍。 - 创设更多的实践机会：增加编程练习的次数，让学生有更多的机会动手实践，提高他们的编程技能。 - 强化课堂互动：通过提问、讨论等方式，增加课堂互动，提高学生的参与度。 - 利用信息技术辅助教学：更多地利用在线平台和编程软件，让学生在课后也能进行自主学习和练习。 在未来的教学中，我会根据这些反思和改进措施，调整教学策略，努力提高教学质量，帮助学生更好地理解 and 掌握程序设计知识。我相信，通过不断的反思和改进，我们能够为学生提供更加优质的教学。
内容逻辑关系	① 程序设计基本概念 - 重点知识点：程序、算法、代码、执行 - 重点词：指令、序列、逻辑 - 重点句：程序是一系列按顺序执行的指令，用于完成特定任务。 ② 程序设计基本步骤 - 重点知识点：问题分析、算法设计、代码编写、程序测试 - 重点词：分析、设计、编写、测试 - 重点句：程序设计遵循分析问题、设计算法、编写代码、测试程序的基本步骤。 ③ 编程实践与逻辑思维 - 重点知识点：变量、循环、条件语句、调试 - 重点词：变量、循环、条件、调试 - 重点句：编程实践中，使用变量存储数据，通过循环和条件语句控制程序流程，调试是找出并修复代码错误的重要步骤。
典型例题讲解	
例题 1：编写一个程序，输入三个整数，输出这三个整数中的最大值。 解题步骤： 1. 定义三个变量用于存储输入的整数。 2. 通过条件语句比较这三个整数的大小。 3. 输出最大的整数。 答案： <pre> python a = int(input("请输入第一个整数：")) b = int(input("请输入第二个整数：")) c = int(input("请输入第三个整数：")) if a > b: if a > c: print("最大的整数是：", a) else: </pre>	

```
print("最大的整数是：",
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。
。如要下载或阅读全文，请访问：

<https://d.book118.com/405311142314011333>