

The Enlightenment of Computer Education in Current U.S. High Schools to Corresponding Education in China

ACM TURC 2021

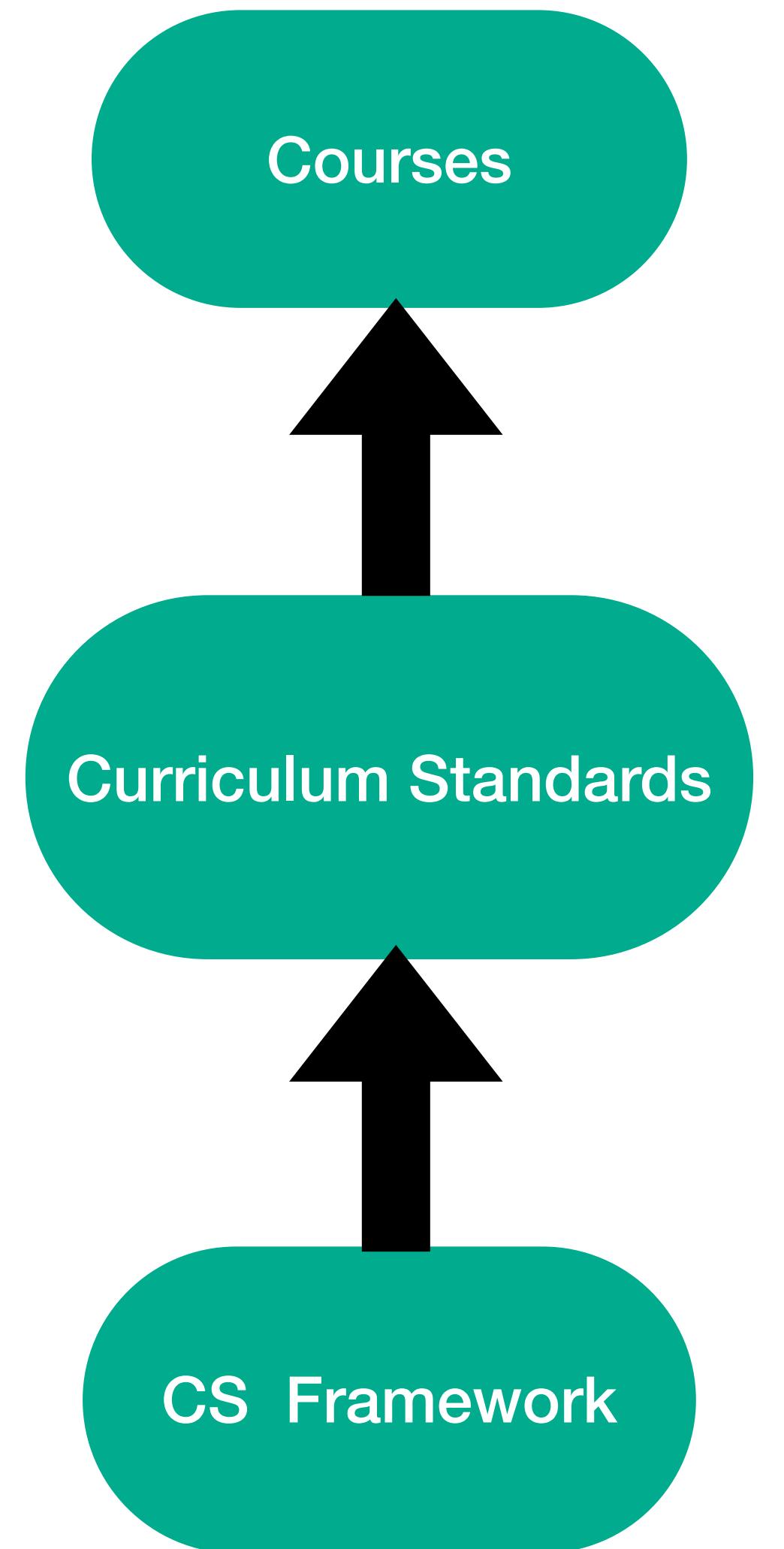
ACM SIGCSE China

Hefei, China

Xiaochun Yang, Jing Zhang, Meiling Cai, Zida Hu, Wenbing Lu Aug 1,2021

Discussion Topics

- 1 Introduction to K-12 Computer Science Education Framework in two countries
- 2 Comparison between curriculum standards in two countries
- 3 Advanced Placement Courses
- 4 Suggestions for Computer Science Education



美国高中计算机教育对我国相应教育的启示

- Education collaboration practice between industry professionals and university teachers
- Software Guide Vol.20 No.4 Apr 2021

杨晓春¹, 张 锦², 蔡美玲², 胡子达², 陆玟冰²

(1. 上海成趣信息科技有限公司, 上海 200000; 2. 湖南师范大学 信息科学与工程学院, 湖南 长沙 410081)

摘要:通过研究美国高中计算机教育对我国相应教育的启示,促进我国高中计算机教育的发展。介绍中美两国中小学计算机教育发展的背景和过程,总结相关研究进展,描述中美两国高中计算机教育框架的组成和目标,分析比较两国体系中核心概念和核心实践的内容。然后,介绍两国高中课程标准,并以算法和程序实现为例,分析两国课程标准的异同。通过比较两国大学先修课程的内容、教学体系、结构和评价方式,探讨两国高中计算机教育对大学计算机教育的影响以及未来两国大学教育可能存在的差异。最后基于全文内容,提出我国高中计算机教育发展的建议和期望。

关键词:K-12 计算机科学框架;计算思维;普通高中信息技术课程标准;高中信息技术教育;大学先修课程

DOI:10.11907/rjdc.211045

中图分类号:G434

文献标识码:A

开放科学(资源服务)标识码(OSID):

文章编号:1672-7800(2021)004-0013-07



Enlightenment of Computer Education in Current U.S. High Schools to Corresponding Education in China

YANG Xiao-chun¹, ZHANG Jin², CAI Mei-ling², HU Zi-da², LU Wen-bing²

(1. *Shanghai AchieveFun Info Tech Co., Ltd., Shanghai 200000, China;*

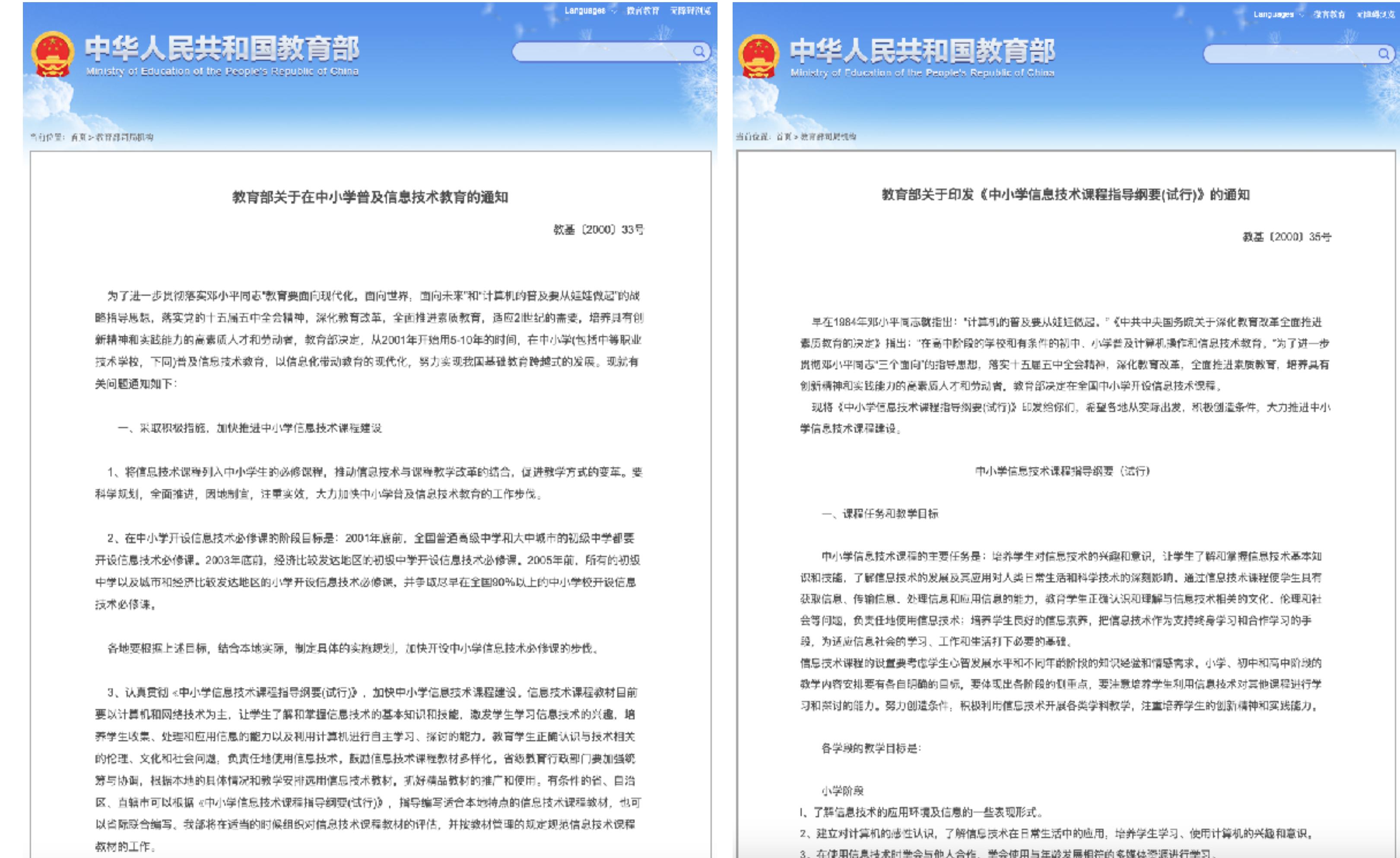
2. *School of Information Science and Engineering, Hunan Normal University, Changsha 410081, China*)

1. Introduction to K-12 Computer Science Education Framework in two countries

History of K-12 IT education in China

- 2000年11月14日，教育部关于在中小学普及信息技术教育的通知教基〔2000〕33号；

- 教育部关于印发《中小学信息技术课程 指导纲要(试行)》的通知教基〔2000〕35号，介绍中小学信息技术课程的主要任务。



The image shows two side-by-side screenshots of the official website of the Ministry of Education of the People's Republic of China. Both screenshots feature the ministry's logo and name at the top. The left screenshot displays the document titled 'Notice on Promoting Information Technology Education in Primary and Secondary Schools' (教基〔2000〕33号), dated November 14, 2000. It includes a brief introduction about the goal of achieving widespread IT education in schools by 2010, followed by three main points detailing the implementation plan. The right screenshot displays the document titled 'Notice of the Ministry of Education on Issuing the Trial Guidance Document for Primary and Secondary School Information Technology Curriculum' (教基〔2000〕35号), also dated November 14, 2000. This document begins with a quote from Deng Xiaoping and outlines the curriculum's main tasks, emphasizing its role in cultivating students' interest in IT and its practical application.

History of K-12 IT education in China

- 2003年，教育部印发的普通高中课程方案和课程标准实验稿，指导了十余年来普通高中课程改革的实践；
- 2013年，教育部启动了普通高中课程修订工作。本次修订深入总结21世纪以来我国普通高中课程改革的宝贵经验，充分借鉴国际课程改革的优秀成果，努力将普通高中课程方案和课程标准修订成既符合我国实际情况，又具有国际视野的纲领性教学文件，构建具有中国特色的普通高中课程体系。

普通高中信息技术课程标准（2017年版2020年修订）

课程性质

普通高中
信息技术课程标准
(2017年版 2020年修订)
中华人民共和国教育部制定



普通高中信息技术课程是一门旨在全面提升学生信息素养，帮助学生掌握信息技术基础知识与技能、增强信息意识、发展计算思维、提高数字化学习与创新能力、树立正确的信息社会价值观和责任感的基础课程。课程围绕高中信息技术学科核心素养，精炼学科大概念，吸纳学科领域的前沿成果，构建具有时代特征的学习内容；课程兼重理论学习和实践应用，通过丰富多样的任务情境，鼓励学生在数字化环境中学习与实践；课程倡导基于项目的学习方式，将知识建构、技能培养与思维发展融入到运用数字化工具解决问题和完成任务的过程中；课程提供学习机会，让学生参与到信息技术支持的沟通、共享、合作与协商中，体验知识的社会性建构，增强信息意识，理解信息技术对人类社会的影响，提高信息社会参与的责任感与行为能力，从而成为具备较高信息素养的中国公民。

普通高中信息技术课程标准（2017年版2020年修订）

一、课程性质与基本理念 1

(一) 课程性质 / 1

(二) 基本理念 / 2

二、学科核心素养与课程目标 5

(一) 学科核心素养 / 5

(二) 课程目标 / 7

三、课程结构 8

(一) 设计依据 / 8

(二) 结构 / 9

(三) 学分与选课 / 10

四、课程内容 12

(一) 必修课程 / 12

普通高中信息技术课程标准（2017年版2020年修订）

(二) 选择性必修课程 / 18

(三) 选修课程 / 34

五、学业质量 39

(一) 学业质量内涵 / 39

(二) 学业质量水平 / 39

(三) 学业质量水平与考试评价的关系 / 43

六、实施建议 44

(一) 教学与评价建议 / 44

(二) 学业水平考试命题建议 / 55

(三) 教材编写建议 / 59

(四) 地方和学校实施本课程的建议 / 63

附录 66

附录1 信息技术学科核心素养水平划分 / 66

附录2 项目教学案例 / 72

普通高中信息技术课程标准（2017年版2020年修订）

基本理念

- 1. 坚持立德树人的课程价值观，培养具备信息素养的中国公民
- 2. 设置满足学生多元需求的课程结构，促进学生的个性化发展
- 3. 选择体现时代性和基础性的课程内容，支撑学生信息素养的发展
- 4. 培育以学习为中心的教与学关系，在问题解决过程中提升信息素养
- 5. 构建基于学科核心素养的评价体系，推动数字化时代的学习创新

普通高中信息技术课程标准（2017年版2020年修订）

核心素养和课程目标

（一）学科核心素养

学科核心素养是学科育人价值的集中体现，是学生通过学科学习而逐步形成的正确价值观、必备品格和关键能力。高中信息技术学科核心素养由信息意识、计算思维、数字化学习与创新、信息社会责任四个核心要素组成。它们是高中学生在接受信息技术教育过程中逐步形成的信息技术知识与技能、过程与方法、情感态度与价值观的综合表现。四个核心要素互相支持，互相渗透，共同促进学生信息素养的提升，具体内涵表述如下。

（二）课程目标

高中信息技术课程旨在全面提升全体高中生的信息素养。课程通过提供技术多样、资源丰富的数字化环境，帮助学生掌握数据、算法、信息系统、信息社会等学科大概念，了解信息系统的基本原理，认识信息系统在人类生产与生活中的重要价值，学会运用计算思维识别与分析问题，抽象、建模与设计系统性解决方案，理解信息社会特征，自觉遵循信息社会规范，在数字化学习与创新过程中形成对人与世界的多元理解力，负责、有效地参与到社会共同体中，成为数字化时代的合格中国公民。

K-12 Computer Science Framework

- In October 2016, the United States "K-12 Computer Science Framework" (K-12 Computer Science Framework) was officially released (referred to as "framework").
- "Framework" describes the primary collection of computer science **concepts and practices**, and is the **basis for** the development of the new national computer science education **standards** in the United States.
- The framework provides information for standards and curriculum, professional development, and the implementation of computer science pathways.

K-12 Computer Science Framework

K12 COMPUTER SCIENCE FRAMEWORK



Table of Contents

Acknowledgments	ii
Executive Summary	1
1. A Vision for K-12 Computer Science	7
2. Equity in Computer Science Education	21
3. Development Process	39
4. Navigating the Framework	55
5. Practices Including Computational Thinking	65
6. Concepts Including Crosscutting Concepts	85
7. Guidance for Standards Developers	123
8. Implementation Guidance: Curriculum, Course Pathways, and Teacher Development	145
9. Computer Science in Early Childhood Education	181
10. The Role of Research in the Development and Future of the Framework	199
Appendices	229
Appendix A: Feedback and Revisions	231
Appendix B: Biographies of Writers and Development Staff	245
Appendix C: Glossary	259
Appendix D: Early Childhood Research Review	269
Appendix E: Bibliography of Framework Research	277
Appendix F: Frequently Asked Questions	291
Photo Credits	297

- <https://k12cs.org/wp-content/uploads/2016/09/K%20Computer-Science-Framework.pdf>

A Vision for K-12 Computer Science

- The **K-12 Computer Science Framework** envisions a future in which students
 - critically engage in public discussion on computer science topics;
 - develop as learners, users, and creators of computer science knowledge and artifacts;
 - better understand the role of computing in the world around them;
 - and learn, perform, and express themselves in other subjects and interests.
 - 1批判地参与有关计算机科学主题的公开讨论;2发展为计算机科学知识和工件的学习者、用户和创造者;3更好地了解计算在他们周围世界中的作用;4在其他学科和兴趣中学习、执行和表达自己。

The K-12 computer science framework

The Concepts and Practices of the K-12 Computer Science Framework

Core Concepts

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Core Practices

1. Fostering an Inclusive Computing Culture
2. Collaborating Around Computing
3. Recognizing and Defining Computational Problems
4. Developing and Using Abstractions
5. Creating Computational Artifacts
6. Testing and Refining Computational Artifacts
7. Communicating About Computing

- <https://k12cs.org>

Crosscutting concept in K-12 CS Framework

- 1. Abstraction
- 2. System Relationships
- 3. Human–Computer Interaction
- 4. Privacy and Security
- 5. Communication and Coordination
- 框架还将抽象的和领域通用的思想应用于不同的核心概念，称为跨领域概念，这些跨领域概念提供了跨不同核心概念的主题联系^[4]，包括：
 - 1抽象:将过程或一组信息简化为一组重要特征以供计算使用的结果;
 - 2系统关系:系统的各个部分相互依存，并且出于共同目的而组织的关系;
 - 3人机交互:计算机技术的设计和使用，着重于人(用户)与计算机之间的接口;
 - 4隐私和安全性:隐私是隔离信息并有选择地表达信息的能力，安全是指围绕信息系统的保障措施;
 - 5通信与协调:计算过程的特点是不同代理之间为实现共同结果而进行可靠的通信。

Comparison of core concepts in the framework of computer education in China and USA

China	USA
数据	数据分析 Data and Analysis
算法	算法和程序实现 Algorithms and Programming
信息系统	计算机系统 Computing Systems 网络和互联网 Networks and the Internet
信息社会	计算的影响 Impacts of Computing

Core practices of K-12 computer science framework in USA including computational thinking



Comparison of core practices in the framework of computer education in China and USA

China	USA
信息意识	围绕计算进行合作 Collaborating around computing
数字化学习和创新	关于计算的交流 Communicating about computing
计算思维	认识和定义计算问题:recognizing and defining computational problems 开发和使用抽象:developing and using abstractions 创建计算工件:creating computational artifacts 测试和完善计算工件:testing and refining computational artifacts
信息社会责任	建立包容性计算文化 Fostering an inclusive computing culture

2. Comparison between curriculum standards in two countries

CSTA K-12 Computer Science Standards

- 2017: Computer Science Teachers Association provided CS Standards on standard, descriptive statement, subconcept and practice.
- <https://csteachers.org/page/standards>

High school information technology curriculum module design

表3 我国高中信息技术课程模块设计

类别	模块设计
必修课	模块1:数据与计算 模块2:信息系统与社会 模块1:数据和数据结构 模块2:网络基础 模块3:数据管理和分析
选择必修课	模块4:人工智能初步 模块5:三维设计与创意 模块6:开源硬件项目设计
选修课	模块1:算法初步 模块2:移动应用设计

Algorithm and Programming

- 算法和程序实现属于必修课程 模块 1:数据与计算。
- 我国的课程内容要求如下^[3]:
- 1从生活实例出发，概述算法的概念与特征，运用恰当的描述方法和控制结构表示简单算法；
- 2掌握一种程序设计语言的基本知识，使用程序设计语言实现简单算法，通过解决实际问题，体验程序设计的基本流程，感受算法的效率，掌握程序调试与运行的方法；
- 3通过人工智能典型案例的剖析，了解智能信息处理的巨大进步和应用潜力，认识人工智能在信息社会中的重要作用。

Subconcept in Algorithm and Programming

- Algorithm
- Variables
- Control
- Modularity
- Program Development

The core learning objectives of algorithm and programming in the 3A level of CSTA K-12 computer science curriculum standard in USA

Identifier	Grades	Standard	Concept	Subconcept	Practice(s)
3A-AP-13	9-10	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	>	Algorithms & Programming	Algorithms Creating
3A-AP-14	9-10	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	>	Algorithms & Programming	Variables Abstraction
3A-AP-15	9-10	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.	>	Algorithms & Programming	Control Computational Problems
3A-AP-16	9-10	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	>	Algorithms & Programming	Control Creating
3A-AP-17	9-10	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	>	Algorithms & Programming	Modularity Computational Problems
3A-AP-18	9-10	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	>	Algorithms & Programming	Modularity Creating
3A-AP-19	9-10	Systematically design and develop programs for broad audiences by incorporating feedback from users.	>	Algorithms & Programming	Modularity Creating
3A-AP-20	9-10	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	>	Algorithms & Programming	Program Development Communicating
3A-AP-21	9-10	Evaluate and refine computational artifacts to make them more usable and accessible.	>	Algorithms & Programming	Program Development Testing
3A-AP-22	9-10	Design and develop computational artifacts working in team roles using collaborative tools.	>	Algorithms & Programming	Program Development Collaborating
3A-AP-23	9-10	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	>	Algorithms & Programming	Program Development Communicating

	标准描述	子概念	相应的实践内容
3A-AP-13	通过利用先前的知识和个人兴趣，创建利用算法来解决计算问题的原型。	算法	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3A-AP-14	使用列表来简化解决方案，概括计算问题，而不是重复使用简单变量。	变量	4. 1 从一组相互关联的过程或复杂现象中提取共同特征。
3A-AP-15	当权衡涉及实现、可读性和程序性能时，证明选择特定控制结构的理由，并说明所做选择的利弊。	控制	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3A-AP-16	设计并迭代开发计算工件，以实现实际意图、个人表达，或通过使用事件发起指令来解决社会问题。	控制	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3A-AP-17	使用诸如过程、模块和/或对象之类的结构，通过系统分析将问题分解为较小的组件。	控制	3. 2 将复杂的实际问题分解为可管理的子问题，这些子问题可以集成现有的解决方案或过程。
3A-AP-18	通过使用程序中的子程序、数据和子程序的组合或独立但相互关联的程序来创建工作。	模块化	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3A-AP-19	通过整合用户的反馈，系统地设计和开发针对广大受众的程序。	模块化	5. 1 使用迭代过程来计划计算工件的开发，其中包括对计划的反思和修改，同时要考虑关键特征，时间和资源限制以及用户期望。
3A-AP-20	在使用资源（例如库）时，评估约束或限制计算工件使用的许可。	程序开发	7. 3 通过遵守知识产权和声明成果的归属，负责任地表达思想。
3A-AP-21	评估和完善计算工件，以使其更加可用和可访问。	程序开发	6. 3 多次评估和优化计算工件以增强其性能、可靠性、可用性和可访问性。
3A-AP-22	团队协作时，使用协作工具设计和开发计算工件。	程序开发	2. 4 评估并选择可用于在项目上进行协作的技术工具。
3A-AP-23	在开发复杂程序时使用文本、图形、演示文稿和/或演示文档来记录设计决策。	程序开发	7. 2 使用与预期的受众和目的相符的适当术语描述、证明和撰写计算过程和解决方案

- **3A 级是适用于所有高中学生的计算机科学标准**
- **9-10 年级，14-16岁**

The core learning objectives of algorithm and programming in the 3B level of CSTA K-12 computer science curriculum standard in USA

Identifier	Grades	Standard	Concept	Subconcept	Practice(s)
3B-AP-08	11-12	Describe how artificial intelligence drives many software and physical systems.	➤ Algorithms & Programming	Algorithms	Communicating
3B-AP-09	11-12	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.	➤ Algorithms & Programming	Algorithms	Creating
3B-AP-10	11-12	Use and adapt classic algorithms to solve computational problems.	➤ Algorithms & Programming	Algorithms	Abstraction
3B-AP-11	11-12	Evaluate algorithms in terms of their efficiency, correctness, and clarity.	➤ Algorithms & Programming	Algorithms	Abstraction
3B-AP-12	11-12	Compare and contrast fundamental data structures and their uses.	➤ Algorithms & Programming	Variables	Abstraction
3B-AP-13	11-12	Illustrate the flow of execution of a recursive algorithm.	➤ Algorithms & Programming	Control	Computational Problems
3B-AP-14	11-12	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	➤ Algorithms & Programming	Modularity	Creating
3B-AP-15	11-12	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	➤ Algorithms & Programming	Modularity	Abstraction
3B-AP-16	11-12	Demonstrate code reuse by creating programming solutions using libraries and APIs.	➤ Algorithms & Programming	Modularity	Creating
3B-AP-17	11-12	Plan and develop programs for broad audiences using a software life cycle process.	➤ Algorithms & Programming	Program Development	Creating
3B-AP-18	11-12	Explain security issues that might lead to compromised computer programs.	➤ Algorithms & Programming	Program Development	Communicating
3B-AP-19	11-12	Develop programs for multiple computing platforms.	➤ Algorithms & Programming	Program Development	Creating
3B-AP-20	11-12	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.	➤ Algorithms & Programming	Program Development	Collaborating
3B-AP-21	11-12	Develop and use a series of test cases to verify that a program performs according to its design specifications.	➤ Algorithms & Programming	Program Development	Testing
3B-AP-22	11-12	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	➤ Algorithms & Programming	Program Development	Creating
3B-AP-23	11-12	Evaluate key qualities of a program through a process such as a code review.	➤ Algorithms & Programming	Program Development	Testing
3B-AP-24	11-12	Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	➤ Algorithms & Programming	Program Development	Communicating

标识	标准描述	子概念	相应的实践内容
3B-AP-08	描述人工智能如何驱动许多软件和物理系统。	算法	7. 2 使用与预期的受众和目的相符的适当术语描述、证明和撰写计算过程和解决方案
3B-AP-09	实现人工智能算法，以便与人类对手玩游戏或解决问题。	算法	5. 3 修改现有工件以改进或定制它。
3B-AP-10	使用并改编经典算法来解决计算问题。	算法	4. 2 评估现有技术功能并将其纳入新设计中。
3B-AP-11	评估算法的效率、正确性和清晰度。	算法	4. 2 评估现有技术功能并将其纳入新设计中。
3B-AP-12	比较和对比基本数据结构及其用途。	变量	4. 2 评估现有技术功能并将其纳入新设计中。
3B-AP-13	说明递归算法的执行流程。	控制	3. 2 将复杂的实际问题分解为可管理的子问题，这些子问题可以集成现有的解决方案或过程。
3B-AP-14	使用学生创建的组件（例如过程、模块和/或对象）构造问题的解决方案。	模块化	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3B-AP-15	分析大规模计算问题，并确定可应用于解决方案的可概括模式。	模块化	4. 1 从一组相互关联的过程或复杂现实中提取共同特征。
3B-AP-16	通过使用库和API创建程序实现解决方案来演示代码重用。	模块化	5. 3 修改现有工件以改进或定制它。
3B-AP-17	使用软件生命周期过程为广大受众规划和开发程序。	程序开发	5. 1 使用迭代过程来计划计算工件的开发，其中包括对计划的反思和修改，同时要考虑关键特性、时间和资源限制以及用户期望。
3B-AP-18	说明可能导致计算机程序受损的安全性问题。	程序开发	7. 2 使用与预期的受众和目的相符的适当术语描述、证明和撰写计算过程和解决方案。
3B-AP-19	为多个计算平台开发程序。	程序开发	5. 2 创建用于实际意图、个人表达或解决社会问题的计算工件。
3B-AP-20	在小组软件项目中使用版本控制系统，集成开发环境（IDE）以及协作工具和实践（代码文档）。	程序开发	2. 4 评估并选择可用于在项目上进行协作的技术工具。
3B-AP-21	开发并使用一系列测试用例来验证程序是否按照其设计规范执行。	程序开发	6. 1 通过考虑所有场景并使用测试用例来系统地测试计算工件。
3B-AP-22	修改现有程序以添加其他功能，并讨论有意和无意的含义（例如，破坏其他功能）。	程序开发	5. 3 修改现有工件以改进或定制它。
3B-AP-23	通过诸如代码审查之类的过程评估程序的关键质量。	程序开发	6. 3 多次评估和优化计算工件以增强其性能、可靠性、可用性和可访问性。
3B-AP-24	比较多种程序实现语言，并讨论它们的功能如何使它们适合解决不同类型的问题。	程序开发	7. 2 使用与预期的受众和目的相符的适当术语描述、证明和撰写计算过程和解决方案。

- **3B 级标准是适用于学习完所有高中必学的内容后，还希望学习更多计算机科学课程的学生。**
- **11-12 年级，16-18岁**

Comparison

- Similarities:both emphasizes designing and expressing algorithms based on the needs of solving problems, mastering programming languages, and using programming languages to implement algorithms to solve practical problems.
均强调根据解决问题的需要设计和表示算法，掌握程序设计语言，利用程序语言实现算法以解决实际问题。
- Difference:美国的课程标准在计算思维的培养方面，充分通过实践活动全面培养学生的程序开发能力。其核心目标的制定非常详尽和具体，每一个核心目标通过子概念和核心实践进行阐述，基于此标准也较易于制定相应的课程。

3. Advanced Placement Courses

AP Course in China

- 中国教育学会联合高等教育出版社共同发起并组织实施了“中国大学先修课程(Chinese Advanced Placement, CAP)试点项目”。CAP 提供了计算机基础与前沿科目的基础课程，包括语言设计和大学计算机基础课程，但没有与之对应的考试 http://cap.icourses.cn/#cap_course;
- 北京大学考试院支持建设了中国大学先修课程(Advanced Pre-University Courses, AC)，在全国部分中学选修课体系内开设大学程度的选修课 <http://www.ioe.pku.edu.cn/> (目前无法访问)

AP Course in USA

- Computer Science A
- AP CSP(Computer Science Principles) In development since 2008, AP Computer Science Principles was created with significant support from the National Science Foundation and more than 50 leading high school and higher education computer science educators who piloted the course at their institutions. This rigorous process of development and testing produced a course that reflects the latest scholarship in the field and provides students with a relevant and engaging learning experience.
- AP CSP 考试采用包括多项选择和自由回答的笔试方式，重视学生对于程序设计理念的掌握，而不是细枝末节的语法知识。
- <https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course>



- INCLUDES
- ✓ Course framework
 - ✓ Instructional section
 - ✓ Sample exam questions
 - ✓ Create performance task guidelines

AP® Computer Science Principles

COURSE AND EXAM DESCRIPTION

Effective
Fall 2020

v **Acknowledgments**

1 **About AP**

4 **AP Resources and Supports**

6 **Instructional Model**

7 **About the AP Computer Science Principles Course**

7 College Course Equivalent

7 Prerequisites

COURSE FRAMEWORK

11 **Introduction**

13 **Course Framework Components**

15 Computational Thinking Practices

17 Course Content

20 **Course at a Glance**

23 **Big Idea Guides**

24 Using the Big Idea Guides

27 **BIG IDEA 1:** Creative Development

41 **BIG IDEA 2:** Data

57 **BIG IDEA 3:** Algorithms and Programming

97 **BIG IDEA 4:** Computer Systems and Networks

109 **BIG IDEA 5:** Impact of Computing

INSTRUCTIONAL APPROACHES

129 **Selecting and Using Course Materials**

132 **Instructional Strategies**

140 **Developing Computational Thinking Practices**

152 **Using Strategies for Collaboration**

153 **Differentiating Computer Science Instruction**

CURRICULUM ALIGNMENT

157 **Curriculum Alignment**

158 **Unit at a Glance**

EXAM INFORMATION

163 **Exam Overview**

172 **Sample Exam Questions**

STUDENT HANDOUTS

189 **Create Performance Task**

197 **Guidelines for Completing the Create Performance Task**

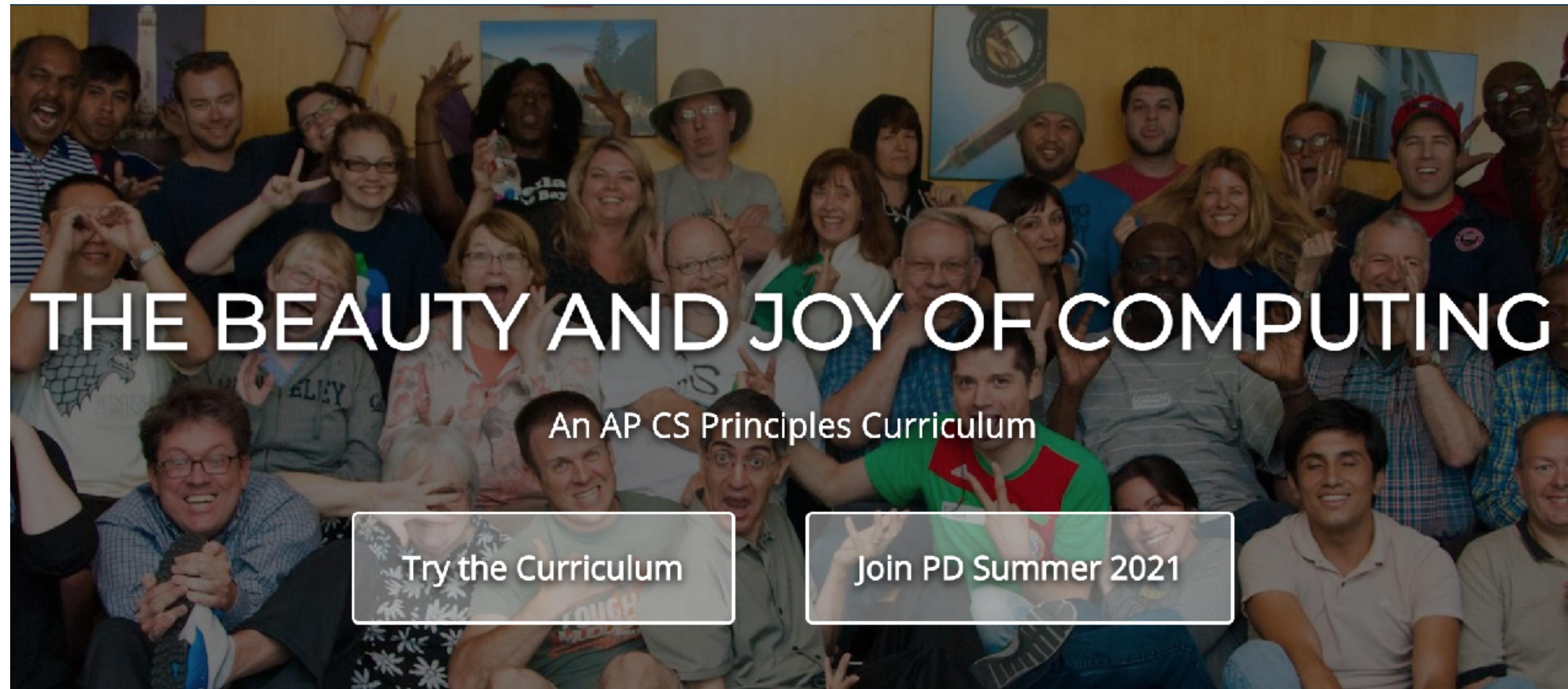
APPENDIX

205 **APPENDIX 1: AP CSP Exam Reference Sheet**

213 **APPENDIX 2: AP CSP Conceptual Framework**

AP Course in USA

- AP Computer Science Principles (AP CSP) represents a college-level introduction to computer science and influenced the development of the K-12 Computer Science Framework.
- Many of the writers and advisors of the K-12 Computer Science Framework were involved in the development of AP CSP



Concepts Crosswalk with AP CSP Principles

K–12 CS Framework Concepts	AP CSP Big Ideas
<i>see the framework's practices</i>	Creativity
Abstraction (crosscutting concept)	Abstraction
Networks and the Internet	Internet
Data and Analysis	Data
Impacts of Computing	Impacts
Algorithms and Programming	Algorithms
	Programming

Practices Crosswalk with AP CSP Principles

K-12 CS Framework Practices	AP CSP Practices
Recognizing and Defining Computational Problems	Connecting Computing Analyzing Problems and Artifacts
Creating Computational Artifacts	Creating Computational Artifacts
Testing and Refining Computational Artifacts	
Developing and Using Abstractions	Abstracting
Communicating About Computing	Communicating
Collaborating Around Computing	Collaborating
Fostering an Inclusive Computing Culture	

The influence to University Education

大量研究表明,美国先修计算机科学考试的成绩与学生入学后的表现呈正相关,这意味着先修课程为后续大学学习打下了良好的知识、计算思维和方法论基础。而目前我国的先修课程侧重于传授知识,例如最新的大纲中包括了人工智能、物联网等新兴技术知识。课程评价方法包括笔试和上机测试,暂时无法为大学教育提供具有参考意义的判断方法^[9]。我国需要建立一个更完善的计算机先修课程体系,改善课程目标、内容和评价方法,并重视发展学生良好的计算机思维能力,从而为其大学生涯奠定良好的基础。

The influence to University Education

- 计算机先修课程除了可以帮助未来选择计算机相关专业的学生打好基础，对需要利用计算机作为工具的其他专业学生也有着非常重要的作用。
 - 一方面，帮助学生奠定计算机科学知识的基础，另一方面培养学生运用计算机科学思维方式解决问题的能力。
- 我国高中计算机教育发展不平衡，导致大一学生信息素养差异较大，增加了大学计算机基础教育的难度。如果我国高中计算机教育能同时重视基础知识的传授和思维能力的培养，计算机先修课成绩便可以为大学教育提供具有参考意义的判断基础，也可以实现中学与大学计算机教育的良好过渡，降低大学计算机基础教育的难度，为学生迎接新的挑战提供更多空间。
- 对教师而言，减少目前偏重基础知识的课程内容是大学计算机基础课教育的新挑战，例如从赋能其他专业的角度开设计算机课程，从而为非计算机相关专业学生的未来发展打下良好基础。

4. Suggestions for Computer Science Education

Seymour Papert's theory of Constructionism

- Seymour Papert regarded learning as a reconstruction rather than a spread of knowledge and he mentioned that the effectiveness of learning is reflected in the construction of meaningful products by learners in the process of engaging in a certain activity.
- 西蒙·派珀特认为学习是一种重建，而不是知识的传播，他提到学习的有效性体现在学习者在从事某种活动的过程中构造有意义的产品上。

Use Constructionism to refine the framework

- adjust the computing education framework to make it more readable and understandable and facilitate more people to participate in. 调整计算机教育框架的宏观设计，使其更具有可读性和易理解性，并促进包括中学教师在内的更多人参与其中
- pay attention to the description of details in the design of curriculum standards, and solve the problem of insufficient guidance in current ordinary high school curriculum standards due to insufficient explanation and lack of consideration of actual teaching conditions. 在课程标准设计上注重细节的描述，解决目前普通高中课程标准由于解释不够明晰、未考虑实际教学情况而存在的指导性不足问题。
- 为系统培养学生的实践能力，还需要增加程序开发的方法论内容，并增设软件开发和抽象实践环节的课程设计。

References

- Association for Computing Machinery, Computer Science Teachers Association, Cyber Innovation Center, National Math and Science Initiative. K-12 computer science framework[EB/OL]. <https://k12cs.org/wp-content/uploads/2016/09/K% E2%80%9312-Computer-Science- Framework.pdf>
- University of California, Berkeley(2020). The beauty and joy of computing[EB/OL]. <https://bjc.berkeley.edu>.
- CS For All. CS for all[EB/OL]. <https://www.csforall.org>.
- PAPERT S. Constructionism:A new opportunity for elementary science education [EB/OL]. http://nsf.gov/awardsearch/showAward?AWD_ID=8751190.

Thanks and Questions