NC Computer Science Initiative Update

Mary Hemphill, Ph.D.
Director of K-12 Computer Science & Technology Education
NC Department of Public Instruction

Computer Science Is Future-Proof

No matter how technology transforms the jobs market, computer science expertise will be crucial.

By Lisette Partelow Contributor June 29, 2016, at 8:00 a.m.

This Is the Fastest-Growing Six-Figure Job in America — and it Do

By **PRACHI BHARDWAJ** February 19, 2019

Apps aren't going away anytime soon — and neither are the jobs creating them.

Application developers — the people responsible for developing, building, and updating computer and mobile apps — make a median salary of $101,790 a year, and for the next five to seven years they're going to be more in demand than any other job making a six-figure-salary in the U.S.

The U.S. will be adding 255,140 app developers to the job market between 2016 and 2026, according to occupational projections website Projections Central. That's about 26,000 new openings for app developers every year, in addition to the estimated 60,170 positions opening up each year to replace current app developers.

Top 10 Computer Science Jobs

1. **Software Developer** ($103,560)
2. **Database Administrator** ($87,020)
3. **Computer Hardware Engineer** ($115,120)
4. **Computer Systems Analyst** ($88,270)
5. **Computer Network Architect** ($104,650)
6. **Web Developer** ($67,990)
7. **Information Security Analyst** ($95,510)
8. **Computer & Information Research Scientist** ($114,520)
9. **Computer & Information Systems Analyst** ($139,220)
10. **IT Project Manager** ($86,126)

Source: https://www.thebalancecareers.com/top-jobs-for-computer-science-majors-205

Top 10 "Must Have" Computer Science Skills

1. Public Speaker
2. Strategic Planning
3. Leadership
4. Project Management
5. Marketing Strategy
6. Business Strategy
7. Management
8. Team Building
9. Business Planning
10. Entrepreneurship

Source: *https://www.linkedin.com/pulse/20140701174737-33437571-top-10-must-have-computer-science-skills*

The Computational Thinkers

concepts

Logic
Predicting & analysing

Evaluation
Making judgements

Algorithms
Making steps & rules

Patterns
Spotting & using similarities

Decomposition
Breaking down into parts

Abstraction
Removing unnecessary detail

approaches

Tinkering
Changing things to see what happens

Creating
Designing & making

Debugging
Finding & fixing errors

Persevering
Keeping going

Collaborating
Working together

Computer science is an academic field of study that covers hardware, software, algorithms, & their applications & impacts on society.

Computational thinking is a set of overlapping problem solving skills, which can be used in a variety of different settings.

Public Schools of North Carolina
State Board of Education
Department of Public Instruction

Report to the North Carolina General Assembly

Expand Computer Science Opportunities to All Students in North Carolina K-12 Schools

SL 2017-157, Part VI

Date Due: January 15, 2018
Report # 48

1

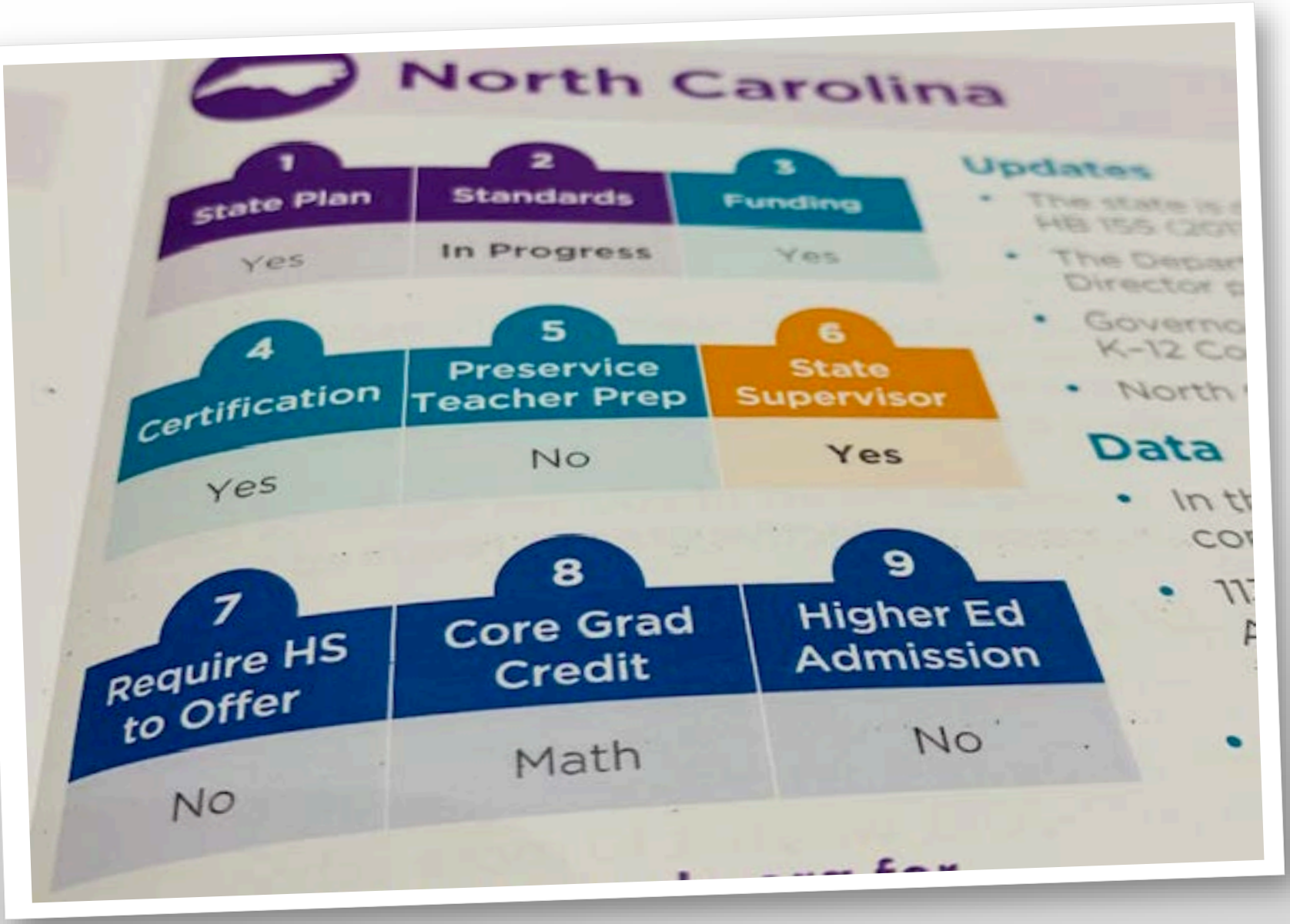
The overall goal of the K-12 Computer Science Initiative is to provide opportunities for all NC students to learn computer science & gain the skills needed to: (1) create & contribute, not just use & consume, in the digital economy; & (2) actively engage as informed citizens in our complex, technology-driven world.

Through collaboration & communication with multiple stakeholders, a coordinated statewide computer science initiative will strengthen pathways from kindergarten to career, address equity gaps, leverage successful programs, & encourage cross-sector partnerships throughout the state.

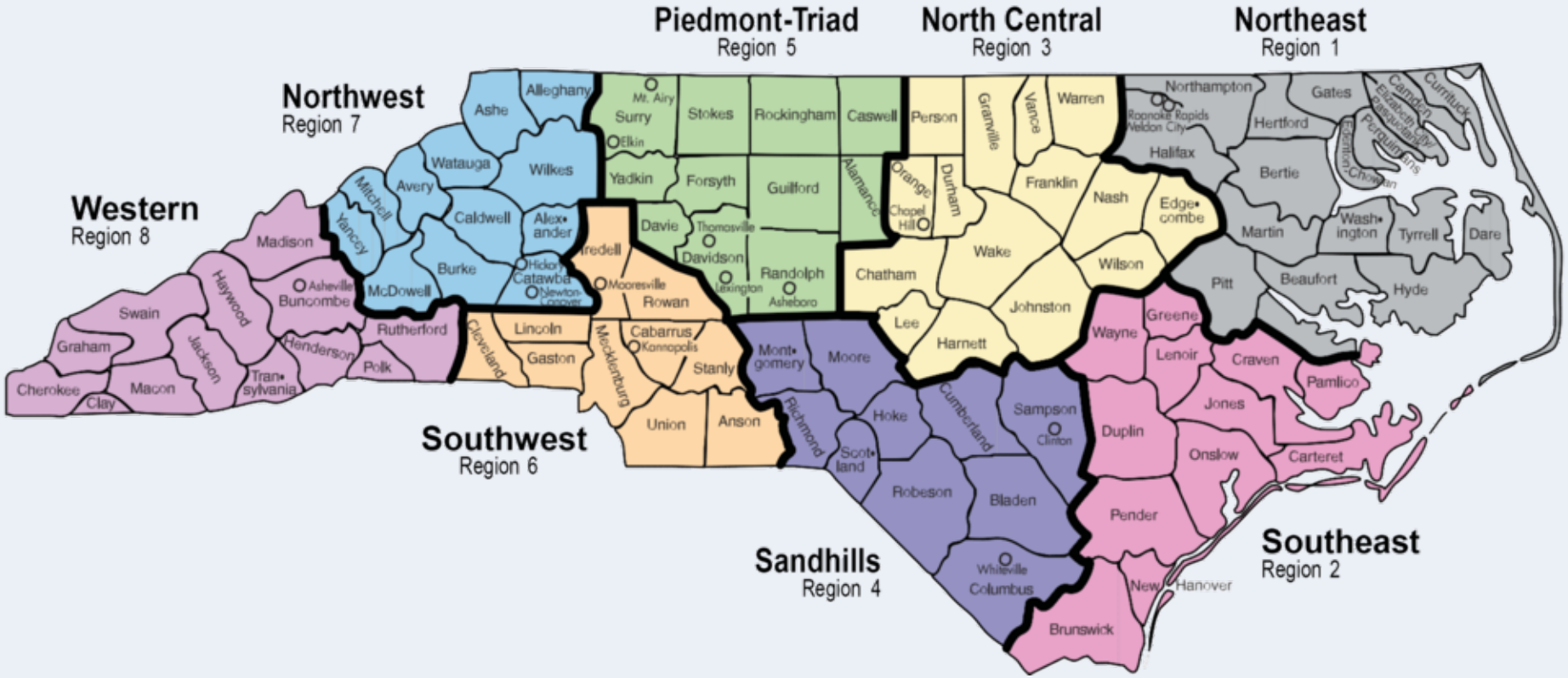
Expand CS Opportunities to All NC Students

Nine State Policies to Expand CS

1. **Create a state plan** for K-12 computer science
2. **Define computer science** and establish rigorous K-12 computer science standards
3. **Allocate funding** for rigorous computer science teacher professional learning and course support
4. **Implement clear certification pathways** for computer science teachers
5. **Create programs** at institutions of higher education to offer computer science to preservice teachers
6. **Establish dedicated computer science positions** in State and Local Education Agencies
7. **Require that all secondary schools offer computer science** with appropriate implementation timelines
8. **Allow computer science to satisfy** a core graduation requirement
9. **Allow computer science to satisfy** an admission requirement at institutions of higher education


North Carolina
1
State Plan
Yes
2
Standards
In Progress
3
Funding
Yes
4
Certification
Yes
5
Preservice Teacher Prep
No
6
State Supervisor
Yes
7
Require HS to Offer
No
8
Core Grad Credit
Math
9
Higher Ed Admission
No
Updates
Data

Mobilizing CS Stakeholders



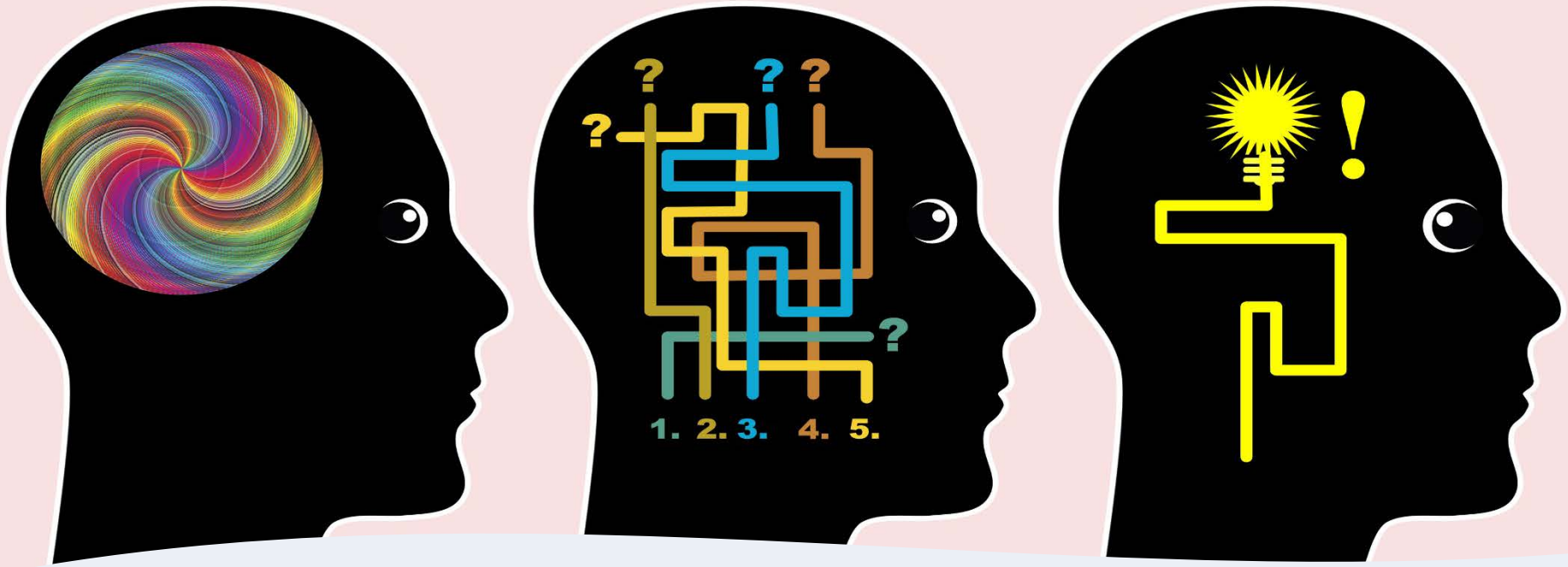
CS representation across the state with the development of K-12 writing & review teams

Mobilizing CS Stakeholders

56 Teacher leaders
9 School leaders
16 District leaders
15 Curriculum leaders
32 External stakeholders

Proposed NC K-12 Computer Science Standards

Concept / Subconcept	Grades K-2 (Ages 5-7) By the end of Grade 2, students will be able to...	Grades 3-5 (Ages 8-11) By the end of Grade 5, students will be able to...	Grades 6-8 (Ages 11-14) By the end of Grade 8, students will be able to...	Introduction to CS By completing this course, students will be able to...	High S... By completing a course, students will be able to...
Computing Systems — Devices	K2-CS-01 Choose appropriate devices to perform a variety of classroom tasks.(P1.1)	35-CS-01 Evaluate the tradeoffs between the features available on digital devices to perform a variety of classroom tasks. (P7.2)	68-CS-01 Understand the design of computing devices, based on an analysis of how users interact with the devices. (P3.3)	ICS-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P4.1)	
Computing Systems — Hardware & Software	K2-CS-02 Describe the function of common physical components of computing systems (hardware) with appropriate terminology. (P7.2) K2-CS-04 (Computing Systems/Hardware and Software) Operate appropriate software to perform a variety of tasks	35-CS-02 Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)	68-CS-02 Design projects that combine hardware and software components to collect and process data. 68-CS-77 Illustrate the process of file management and operating systems.	ICS-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. (P4.1) ICS-CS-03 Explain the roles of operating systems including memory management, data storage/retrieval, process management, and access control.(P7.2)	HS-CS-01 Explain how computing devices manage and allocate shared resources. (P7.2) HS-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components. (P7.2) HS-CS-03 Describe the use of artificial intelligence within computing systems. (P7.2) Moved to AI
Computing Systems — Troubleshooting	K2-TS-77 Describe basic hardware and software problems with accurate terminology. (P6.2, P7.2)	35-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)	68-CS-03 Systematically identify and fix problems with computing devices and their components. (P6.2)	ICS-CS-04 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	HS-CS-77 Utilize guidelines that convey systematic troubleshooting strategies that debug computer systems. (P6.2)
Networks & The Internet — Network Communication & Organization		35-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)	68-NI-04 Analyze different ways that data is transferred across an network and the role of protocols in transmitting data. (P4.4)	ICS-NI-05 Evaluate the relationship between routers, switches, servers, and topology in regards to networks. (P4.1)	HS-NI-04.1 Identify issues of network functionality in computational artifact design. (P3.3, P5.2) HS-NI-04.2 Analyze issues of network functionality in computational artifact design. (P3.3, P5.2)
Networks & The Internet — Cybersecurity	K2-NI-04 Apply knowledge of what passwords are and why we use strong passwords to protect devices and information from unauthorized access. (P7.3)	35-NI-05 Discover your digital footprint and how personal information can be protected.. (P3.1)	68-NI-05 Explain how physical and digital security measures protect electronic information. (P7.2)	ICS-NI-06 Identify examples to illustrate how sensitive data can be affected by malware and other attacks. (P7.2)	HS-NI-05.1 Identify issues of unauthorized access and cyber security in computational artifact design. (P3.3, P5.2) HS-NI-05.2 Analyze issues of unauthorized access and cyber security in computational artifact design. (P3.3, P5.2)
Data & Analysis — Storage	K2-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. (P4.2)	35-DA-??? Choose the correct software to open a file based on its extension.	68-NI-06 Apply multiple methods of encryption to model the secure transmission of information. (P4.4) 68-DA-07 Represent data using multiple encoding schemes. (P4.0)	ICS-NI-07 Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P3.3) ICS-NI-08 Compare various security measures, considering tradeoffs between the usability and security of a computing system. (P6.3) ICS-DA-09 Compare different binary representations of data, including text, sound, images, and numbers. (P4.1) ICS-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. (P3.3)	HS-NI-06 Explain tradeoffs when selecting and implementing cybersecurity recommendations for various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P7.2)
Data & Analysis — Collection, Visualization, & Transformation	K2-DA-06 Collect and present the same data in various visual formats. (P7.1, P4.4)	35-DA-06 Organize and present collected data visually to highlight relationships and support a claim. (P7.1)	68-DA-08 Collect data using computational tools. (P6.3) 68-DA-?? Transform the collected data to make it more useful and reliable.	ICS-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena. (P4.4)	HS-DA-07 Identify patterns in data representing complex systems with select data analysis tools and techniques. (P4.1) HS-DA-08.1 Select appropriate data collection tools and techniques. HS-DA-08.2 Compile data sets that support a claim or communicate information. (P7.2)
Data & Analysis — Inference & Models	K2-DA-07 Make predictions with patterns in data visualizations such as charts or graphs. (P4.1)	35-DA-07 Communicate using data to highlight or propose cause-and-effect relationships. (P7.1)	68-DA-09 Refine computational models based on the data they have generated and/or data collected from a phenomenon or process. (P5.3, 4.4)	ICS-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. (P4.4)	HS-DA-09.1 Identify the ability of models and simulations to test hypotheses. HS-DA-09.2 Determine hypotheses with select models and simulations. (P4.4)
Algorithms	K2-AP-08 Model daily processes with algorithms (sets of step-by- step instructions) to complete tasks. (P4.4)	35-AP-08 Create multiple algorithms for the same task to determine which is the most appropriate. (P6.3, P3.3)	68-AP-10 Implement flowcharts and/or pseudocode to address complex problems as algorithms. (P4.4, P4.1)	ICS-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. (P5.2)	HS-AP-10 Identify artificial intelligence algorithms. (P7.2) HS-AP-11 Solve computational problems with classic algorithms. (P4.2) HS-AP-12 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
Algorithms — Variables	K2-AP-09 Demonstrate how programs store and manipulate data by using numbers or other symbols to represent information.	35-AP-09 Create programs that use variables to store and modify data. (P5.2)	68-AP-11 Create clearly named variables that represent different data types. (P5.1, P5.2) 68-AP-?? Perform operations on the values.	ICS-AP-14 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using	HS-AP-13 Select an appropriate data structure for information of a given problem. (P4.2)
Algorithms — Control	K2-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. (P5.2)	35-AP-10 Create programs that include sequences, events, loops, and conditionals. (P5.2)	68-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	ICS-AP-15 Justify the selection of specific control structures, considering implementation, readability, and program performance. (P5.2) ICS-AP-16 Iteratively develop computational artifacts for practical intent,	HS-AP-14 Illustrate the flow of execution of a recursive algorithm. (P3.2)



Mobilizing CS Stakeholders

ES/MS – focus on K-2 & 3-5 proposed standards & vertical bridge to middle school standards, teaching, & learning

MS – focus on 6-8 proposed standards, current MS CS courses, & vertical bridge to high school standards, teaching, & learning

HS – focus on 9-12 proposed standards, current HS CS courses, & impact on college & career pathways

NC CS Proposed Standards

North Carolina Computer Science Initiative

K-12 CS Standards Proposal

Submitted to State Board of Education

June 3, 2019

Introduction

The NC K-12 Computer Science Standards were developed by the CS Steering Committee (as part of the State Board of Education's Special Committee on Digital Learning and Computer Science) in collaboration with the Department of Public Instruction, the Friday Institute for Educational Innovation at NC State's College of Education, and the Lt. Governor's Office. With cross-sector representation and voices from parents, teachers, school and district administrators, business leaders, non-profit and after-school programs, and national experts on computer science (CS) education, the CS Steering Committee conducted a five-month standards review process from February 2019-June 2019 that included over 1,000 person-hours of research, iteration, and vetting to revise the widely-accepted Computer Science Teachers Association (CSTA) K-12 Standards to best fit the needs of students, teachers, schools, and districts in North Carolina.

The CS Steering Committee was guided by the Report to the North Carolina General Assembly: Expand Computer Science to All Students in North Carolina K-12 Schools (2018). The Report clearly outlines goals for CS in K-12 Education:

> *The overall goal is to provide opportunities for all North Carolina students to learn computer science and gain the skills needed to: (1) create and contribute, not just use and consume, in the digital economy; and (2) actively engage as informed citizens in our complex, technology-driven world. Through collaboration and communication with multiple stakeholders, a coordinated statewide computer science initiative will strengthen pathways from kindergarten to career, address equity gaps, leverage successful programs, and encourage cross-sector partnerships throughout the state.*

The first recommendation of this report is "Rigorous computer science content standards for K-12 students."

Rationale

NC CS Proposed Standards

Proposed NC K-12 Computer Science Standards

Concept	Subconcept	Grades K-2 (Ages 5-7) *By the end of Grade 2, students will be able to...*	Grades 3-5 (Ages 8-11) *By the end of Grade 5, students will be able to...*	Grades 6-8 (Ages 11-14) *By the end of Grade 8, students will be able to...*	Introduction to CS *By completing this course, students will be able to...*	High School – CS Level 1 *By completing a course, students will be able to...*
Computing Systems	Devices	**K2-CS-01** Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. (P1.1)	**35-CS-01** Describe how internal and external parts of computing devices function to form a system. (P7.2)	**68-CS-01** Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. (P3.3)	**ICS-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P4.1)	
	Hardware & Software	**K2-CS-02** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). (P7.2)	**35-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)	**68-CS-02** Design projects that combine hardware and software components to collect and exchange data. (P5.1)	**ICS-CS-02** Compare levels of abstraction and interactions between application software, system software, and hardware layers. (P4.1)	**HS-CS-01** Describe how computing devices manage and allocate shared resources. (P7.2)
		K2-CS-03 Describe basic hardware and software problems using accurate terminology. (P6.2, P7.2)			**ICS-CS-03** Identify the roles of operating systems including memory management, data storage/retrieval, process management, and access control. (P7.2)	**HS-CS-02** Illustrate ways computing systems implement logic, input, and output through hardware components. (P7.2)
	Troubleshooting		**35-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)	**68-CS-03** Systematically identify and fix problems with computing devices and their components. (P6.2)	**ICS-CS-04** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.2)	**HS-CS-03** Describe the use of artificial intelligence within computing systems. (P7.2)
Networks & The Internet	Network Communication & Organization		**35-NI-04** Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. (P4.4)	**68-NI-04** Model different ways that data is transferred across a network and the role of protocols in transmitting data. (P4.4)	**ICS-NI-05** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. (P4.1)	**HS-NI-04** Address issues of network functionality in computational artifact design. (P3.3, P5.2)
	Cybersecurity	**K2-NI-04** Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. (P7.3)	**35-NI-05** Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)	**68-NI-05** Explain how physical and digital security measures protect electronic information. (P7.2)	**ICS-NI-06** Give examples to illustrate how sensitive data can be affected by malware and other attacks. (P7.2)	**HS-NI-05** Address issues of unauthorized access and cyber security in computational artifact design. (P3.3, P5.2)
				68-NI-06 Apply multiple methods of encryption to model the secure transmission of information. (P4.4)	**ICS-NI-07** Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P3.3)	**HS-NI-06** Explain tradeoffs when selecting and implementing cybersecurity recommendations for various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P7.2)

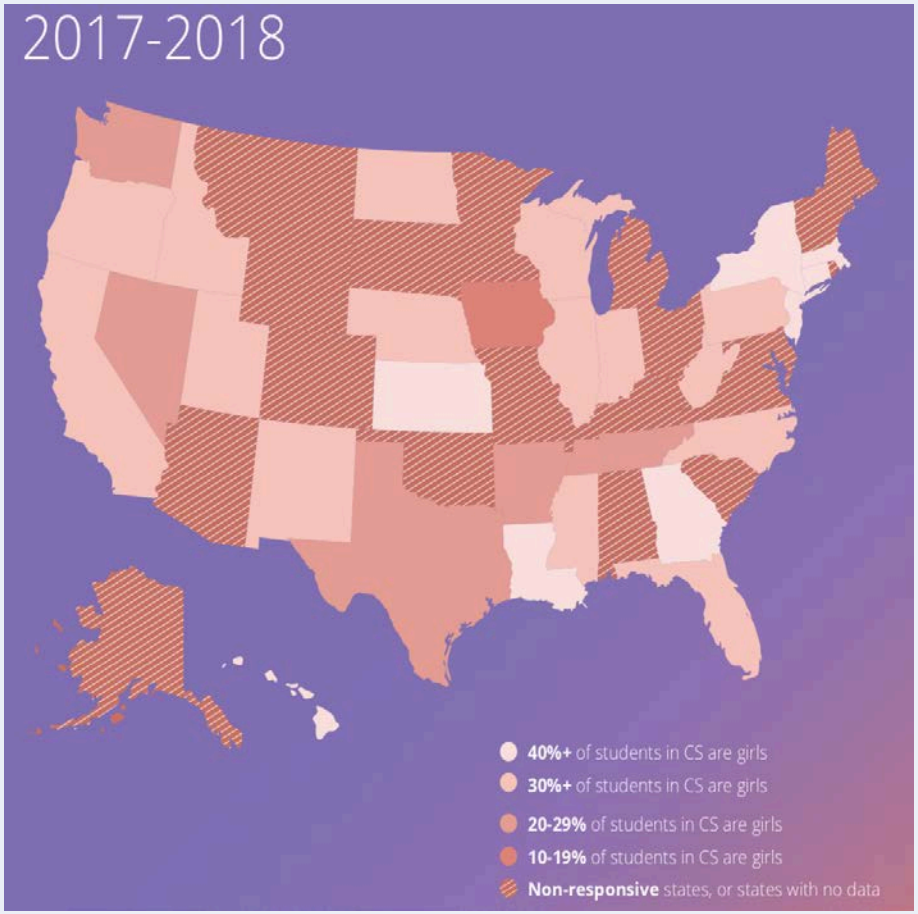
Computing Systems | Networks & the Internet | Data & Analysis | Algorithms & Programming | Impacts of Computing

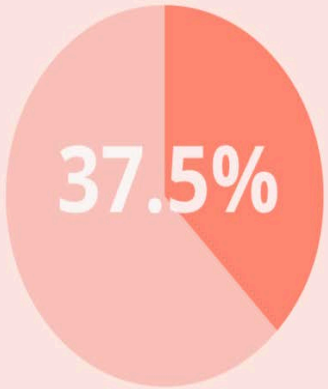
FAQs for NC Computer Science Standards

- What is the difference between Computer Science & Digital Teaching & Learning?
- Where does STEM fit in as it relates to Computer Science?
- How will Computer Science look in grades K-8 in North Carolina?
- Will we continue to have standalone Computer Science courses in middle & high school?
- What can we look forward to in terms of supporting teacher leaders, school leaders, & districts to implement Computer Science in 2020-2021?

Addressing the Gender Gap

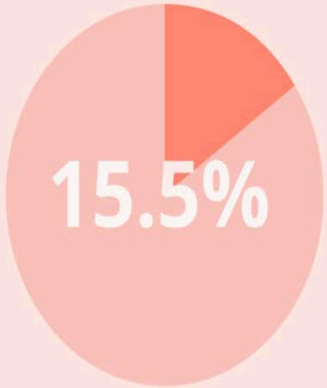
According to states' own data from the last two years, states with policies aimed at increasing the volume of CS classrooms have not seen increased participation by girls. Boys still make up the overwhelming majority of students in CS classrooms.





GIRLS' PARTICIPATION RATE IN K-12 CS IN 2017-18[5]

Nationally, participation by girls in computer science courses in grades K-12 averaged 37.5% of all students.



GIRLS IN HISTORICALLY UNDERREPRESENTED GROUPS (HUGS) PARTICIPATION RATE IN K-12 CS IN 2017-18[6]

Nationally, participation by girls from HUGS in computer science courses in grades K-12 averaged 15.5% of all students.

The State of Girls' Participation in Computer Science in NC

The #IAmCS Campaign aims to move the needle in NC when it comes to the staggering gender & equity gap for NC students in the Computer Science ecosystem.

- show NC K-12 students possibilities for their future with the CS ecosystem in realistic & tangible ways
- feature NC individuals & entities that represent "diversity by industry"
- connect NC-based organizations, entities, & industries to K-12 CS teaching & learning
- increase participation in formal CS education, exploration, & implementation across the state through cross-sector collaboration

21st century learning for 22nd century impact

What are next steps & opportunities for future alignment with the NC CS Initiative?

- Consider local LEA accountability plans for implementation of the CS standards & curricula for the 2020-21 school year
- Early adopters will serve as "prototype districts" & receive support from the CS department at DPI
- CS regional support teams to meet with LEA during 2020-21 school year to guide districts, provide exemplars, & connect with external stakeholders to plan for implementation
- On-going CS Discoveries & CS Principles PD for teachers leaders through Friday Institute; developing CS Fundamentals to meet the needs of K-5 teachers in NC
- #IAmCS ambassadors to be trained in each district to support gender & equity efforts in CS ecosystem

Follow us on Instagram

Computing makes up 2/3 of projected new jobs in STEM

NC had only 1,561 computer science graduates in 2017

5 Core Concepts of CS Standards will focus on

- Computing Systems
- Networks and the Internet
- Data & Analysis
- Algorithms & Programming
- Impacts of Computing

Standards launching 2020-21 school year

COMPUTER SCIENCE & TECHNOLOGY EDUCATION

NC Department of Public Instruction

Through collaboration & communication w/multiple stakeholders, a coordinated statewide CS initiative will:

- Strengthen pathways from kindergarten to career
- Address equity gaps
- Leverage successful programs
- Encourage cross-sector partnerships throughout NC

Only 15% of AP computer science students are Black/African-American or Hispanic/Latino

If you are interested in North Carolina Computer Science Standards development,

Take the survey!

Meet the Team

Linda Lay
CS & IT Education Consultant

Linda.lay@dpi.nc.gov

Meet the Team

Karen McPherson
CS & IT Education Consultant

Karen.mcpherson@dpi.nc.gov

Meet the Team

Russell Strand-Poole
Lead Teacher Facilitator,
Durham Public Schools

Meet the Team

Mary Hemphill
Director of K-12
Computer Science &
Tech Education
Mary.hemphill@dpi.nc.gov