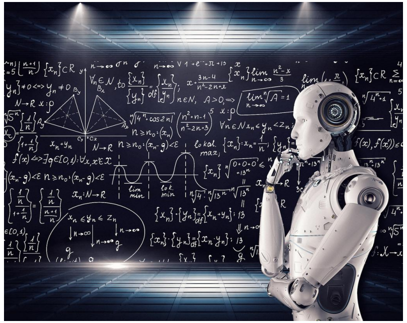


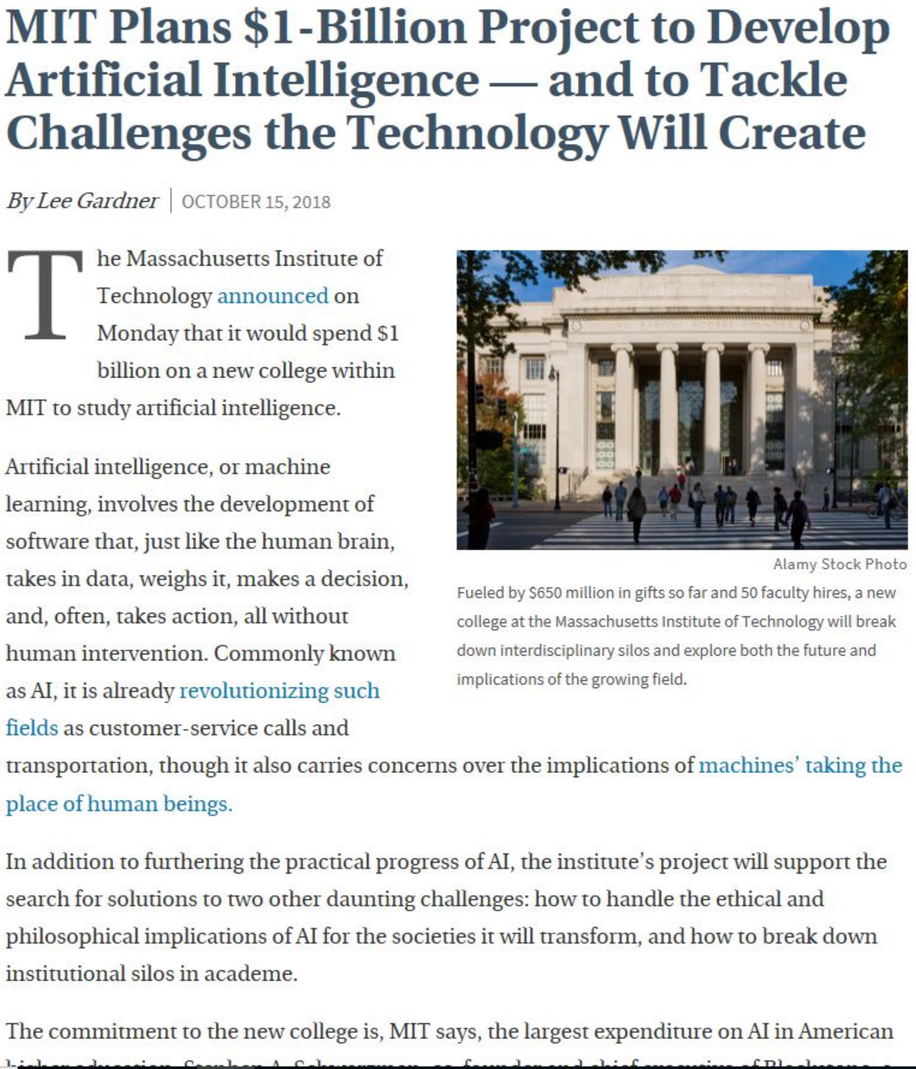
AI in grades 9-12

High School = preparation

For career and/or college

Interdisciplinary impacts:
social, ethical, legal, economic, environmental

Basic literacy:
Remove the “magic”



MIT Plans $1-Billion Project to Develop Artificial Intelligence — and to Tackle Challenges the Technology Will Create

By Lee Gardner | OCTOBER 15, 2018

The Massachusetts Institute of Technology announced on Monday that it would spend $1 billion on a new college within MIT to study artificial intelligence.

Alamy Stock Photo

Fueled by $650 million in gifts so far and 50 faculty hires, a new college at the Massachusetts Institute of Technology will break down interdisciplinary silos and explore both the future and implications of the growing field.

Artificial intelligence, or machine learning, involves the development of software that, just like the human brain, takes in data, weighs it, makes a decision, and, often, takes action, all without human intervention. Commonly known as AI, it is already revolutionizing such fields as customer-service calls and transportation, though it also carries concerns over the implications of machines’ taking the place of human beings.

In addition to furthering the practical progress of AI, the institute’s project will support the search for solutions to two other daunting challenges: how to handle the ethical and philosophical implications of AI for the societies it will transform, and how to break down institutional silos in academe.

The commitment to the new college is, MIT says, the largest expenditure on AI in American

Two different levels in high school

CTE Pathways

AI for ALL:
What should every student know? (3A)

Big Idea #1 - Computers can access data about the world.
Big Idea #2 - Agents maintain internal representations/models of the world and use them for reasoning.
Big Idea #3 - Computers can learn from data.
Big Idea #4 - AI systems strive to interact comfortably with humans.
Big Idea #5 - AI applications can impact society in positive and negative ways.

AI specialty track:
For CS beyond an introduction (3B)

Big Idea #1 - Create & evaluate code to access data.
Big Idea #2 - Develop agents with internal representations/models of the world and use them for reasoning.
Big Idea #3 - Create code to learn from data.
Big Idea #4 - Develop, evaluate, and improve AI systems to interact with humans.
Big Idea #5 - Debate how AI applications can impact society in positive and negative ways.

Integration or Stand Alone?

Possible integration ideas

AI for ALL:

What should every student know? (3A)

Big Idea #1 - Computers can access data about the world. **engineering/technology**

Big Idea #2 - Agents maintain internal representations/models of the world and use them for reasoning. **Science**

Big Idea #3 - Computers can learn from data. **Math**?

Big Idea #4 - AI systems strive to interact comfortably with humans. **?**

Big Idea #5 - AI applications can impact society in positive and negative ways. **Social studies**

Integration vs. Stand Alone pros and cons

Common high school requirements and trends

Typical required coursework for graduation:

- 4 years English
- 4 years Math
- 3 years Science
- 3 years Social Studies
- 2 years Foreign Language
- 1 year phys ed, health, fine arts, technology, financial literacy, etc.
- 3-11 electives
    - CTE pathways: 3-6 courses

High schools offer students 6-8 courses per year.

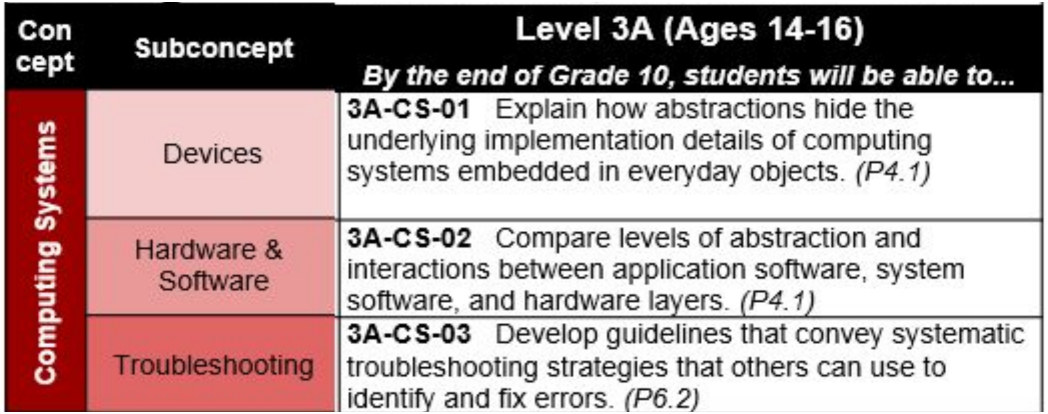
Trends:

- Early release
- College in high school (dual enrollment, AP, etc)
- Only teach what is assessed
- STEM / CTE pathways / student internships and apprenticeships

Attracting and retaining students:

Personal relevance, job opportunities, creativity, humor, inquiry, interdisciplinary, weighted average, friends.

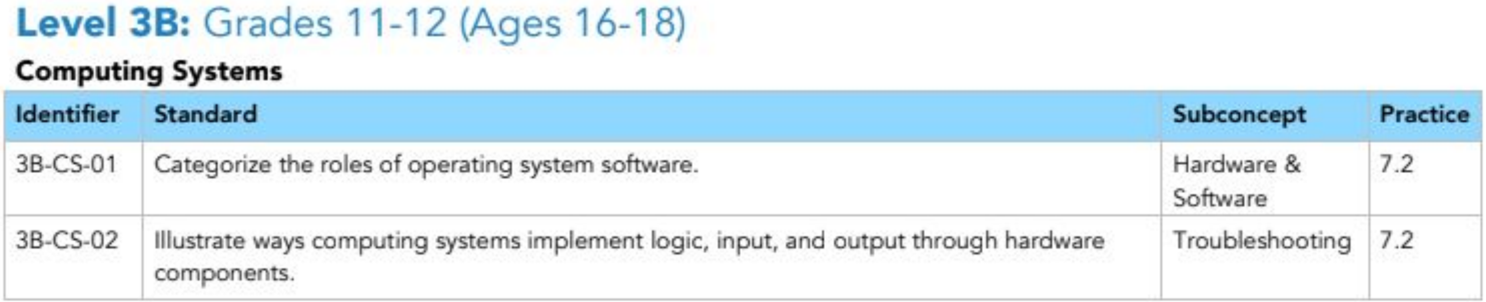
Computing Systems at the high school level



Con cept	Subconcept	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to...*
Computing Systems	Devices	**3A-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. *(P4.1)*
	Hardware & Software	**3A-CS-02** Compare levels of abstraction and interactions between application software, system software, and hardware layers. *(P4.1)*
	Troubleshooting	**3A-CS-03** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. *(P6.2)*

3A: levels of abstraction, troubleshooting

3B: operating systems, hardware

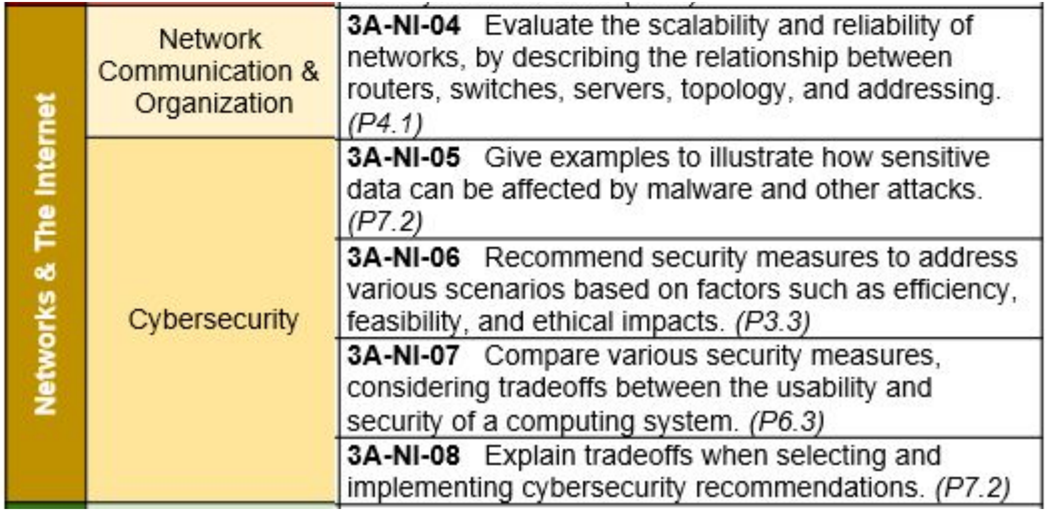


Level 3B: Grades 11-12 (Ages 16-18)

Computing Systems

Identifier	Standard	Subconcept	Practice
3B-CS-01	Categorize the roles of operating system software.	Hardware & Software	7.2
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.	Troubleshooting	7.2

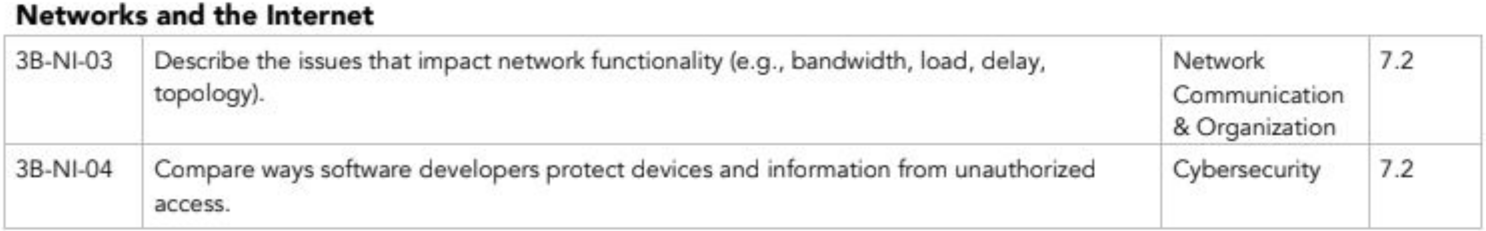
Networks and internet concept and subconcepts



Networks & The Internet	Network Communication & Organization	**3A-NI-04** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. *(P4.1)*
	Cybersecurity	**3A-NI-05** Give examples to illustrate how sensitive data can be affected by malware and other attacks. *(P7.2)*
		3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. *(P3.3)*
		3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. *(P6.3)*
		3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations. *(P7.2)*

3A: network design, basic cyber protection

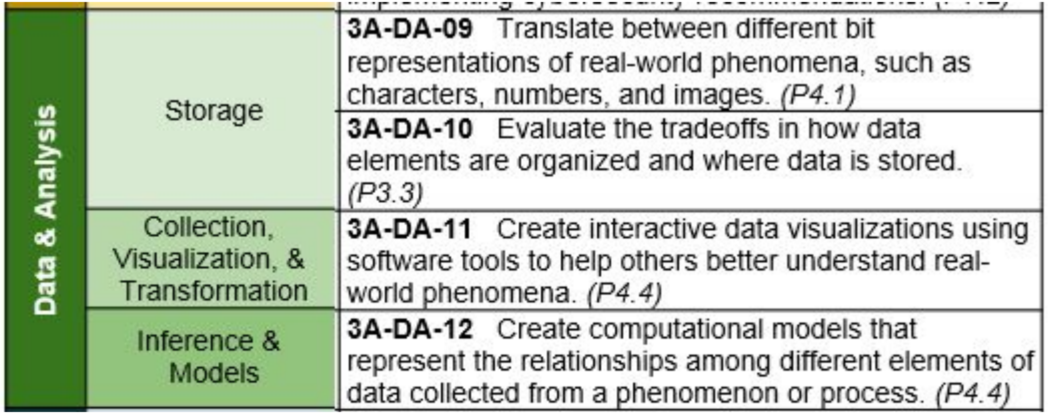
3B: network functionality and details, cybersecurity software development



Networks and the Internet

3B-NI-03	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).	Network Communication & Organization	7.2
3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.	Cybersecurity	7.2

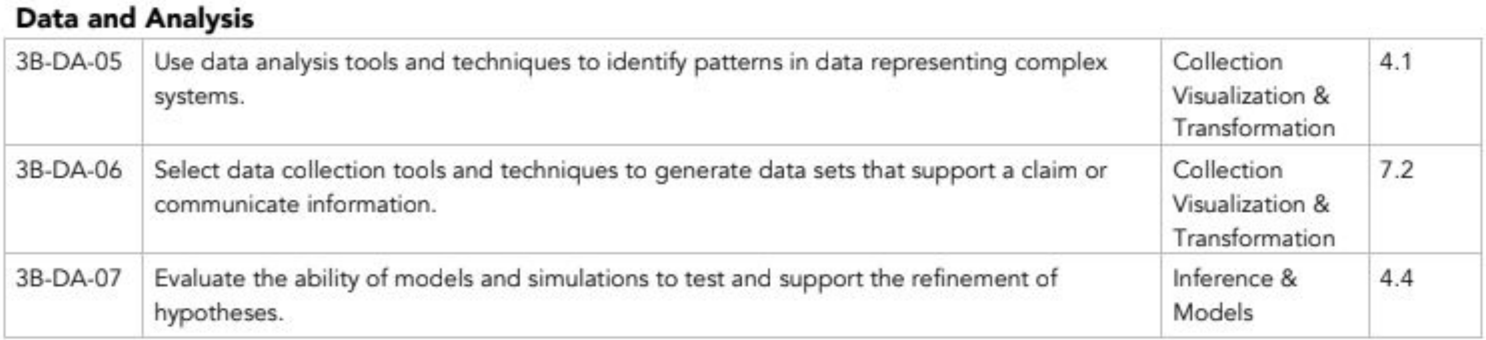
Data



Data & Analysis	Storage	**3A-DA-09** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. *(P4.1)*
		3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. *(P3.3)*
	Collection, Visualization, & Transformation	**3A-DA-11** Create interactive data visualizations using software tools to help others better understand real-world phenomena. *(P4.4)*
	Inference & Models	**3A-DA-12** Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. *(P4.4)*

3A: data representation, data storage, visualizations, create models

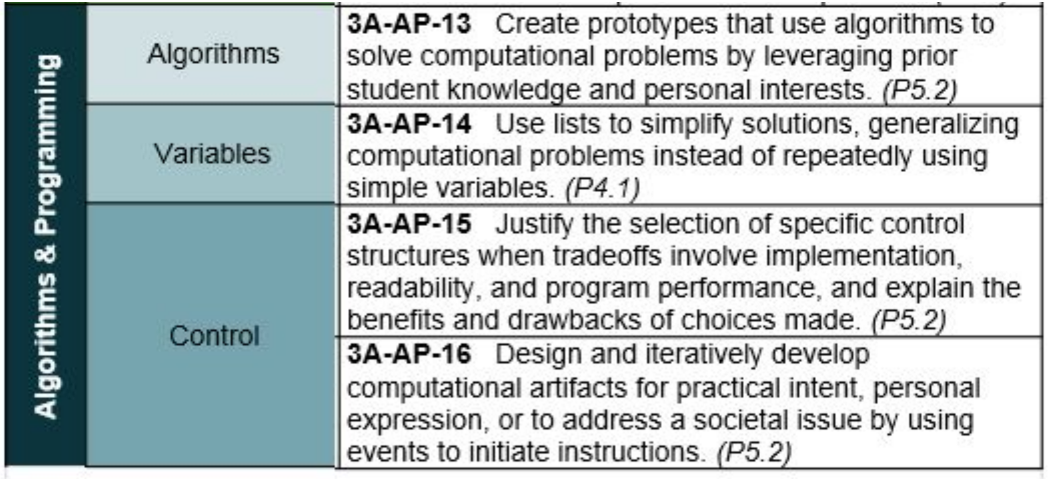
3B: complex systems, evaluation of data collection, models to support a hypothesis



Data and Analysis

3B-DA-05	Use data analysis tools and techniques to identify patterns in data representing complex systems.	Collection Visualization & Transformation	4.1
3B-DA-06	Select data collection tools and techniques to generate data sets that support a claim or communicate information.	Collection Visualization & Transformation	7.2
3B-DA-07	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	Inference & Models	4.4

Algorithms

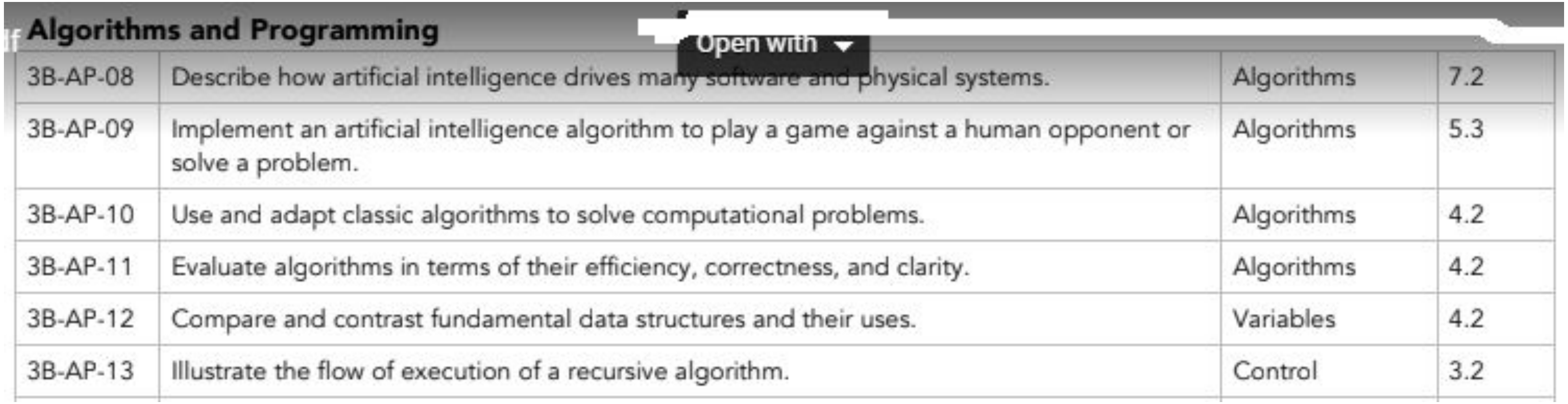


Algorithms & Programming	Algorithms	**3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. *(P5.2)*
	Variables	**3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. *(P4.1)*
	Control	**3A-AP-15** Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. *(P5.2)*
		3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. *(P5.2)*

3A: basic program design

3B: specific reference to AI

Evaluate algorithms, data structures, recursion

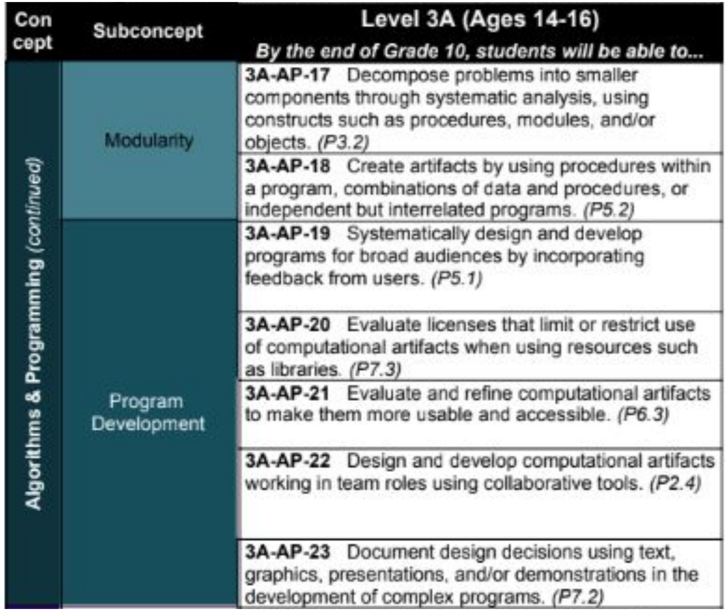


Algorithms and Programming

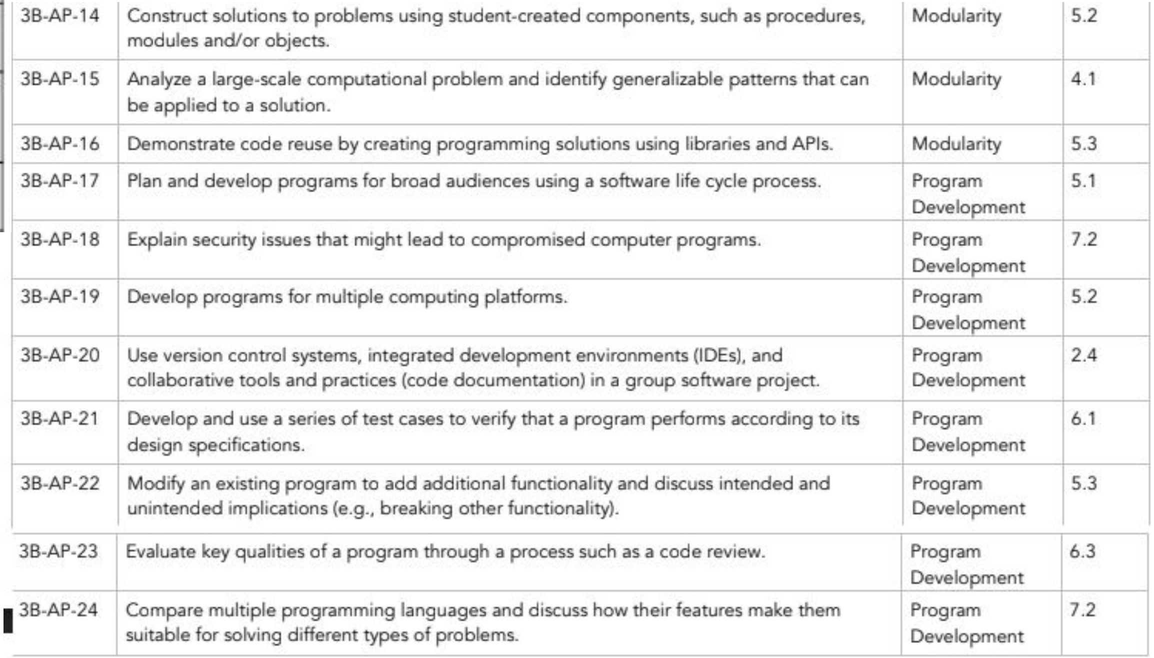
3B-AP-08	Describe how artificial intelligence drives many software and physical systems.	Algorithms	7.2
3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.	Algorithms	5.3
3B-AP-10	Use and adapt classic algorithms to solve computational problems.	Algorithms	4.2
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.	Algorithms	4.2
3B-AP-12	Compare and contrast fundamental data structures and their uses.	Variables	4.2
3B-AP-13	Illustrate the flow of execution of a recursive algorithm.	Control	3.2

Modularity and Program Development

Focus on developing actual programs in 3B



Con cept	Subconcept	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to...*
Algorithms & Programming *(continued)*	Modularity	**3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. *(P3.2)*
		3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. *(P5.2)*
	Program Development	**3A-AP-19** Systematically design and develop programs for broad audiences by incorporating feedback from users. *(P5.1)*
		3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. *(P7.3)*
		3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. *(P6.3)*
		3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. *(P2.4)*
		3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. *(P7.2)*



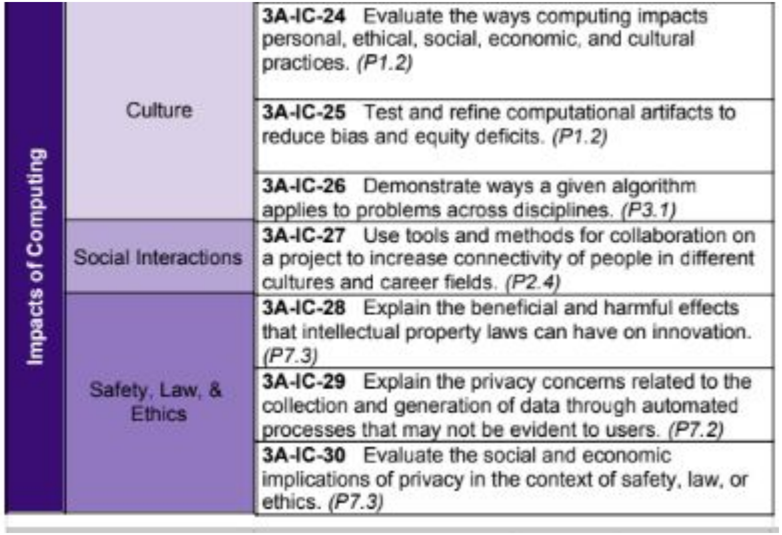
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	Modularity	5.2
3B-AP-15	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	Modularity	4.1
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.	Modularity	5.3
3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.	Program Development	5.1
3B-AP-18	Explain security issues that might lead to compromised computer programs.	Program Development	7.2
3B-AP-19	Develop programs for multiple computing platforms.	Program Development	5.2
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.	Program Development	2.4
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.	Program Development	6.1
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Program Development	5.3
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.	Program Development	6.3
3B-AP-24	Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	Program Development	7.2

Impacts

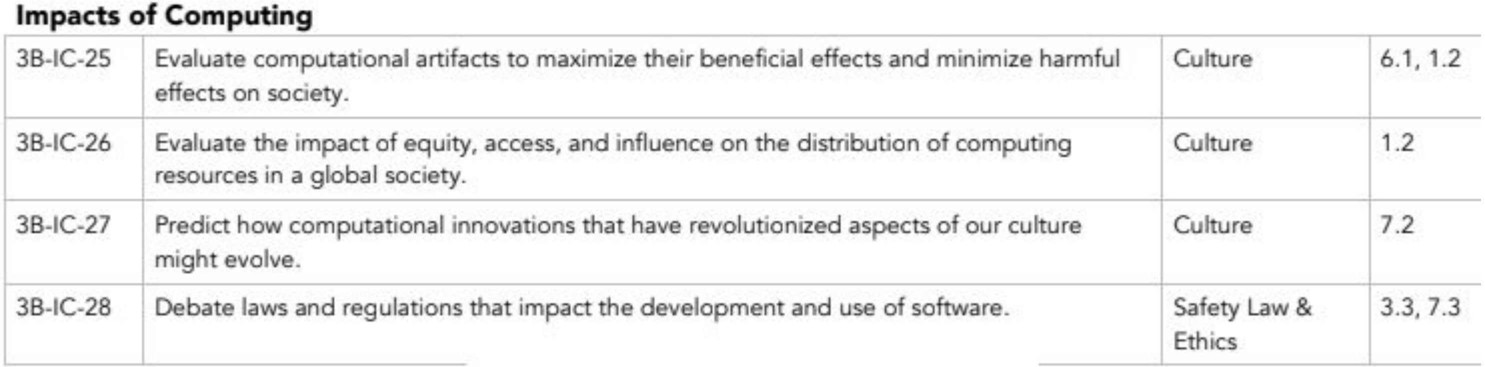
Subtle differences:

Culture: evaluate and test vs. maximizing benefits and minimizing risks

Safety, Law & Ethics: explain and evaluate vs. debate



Impacts of Computing	Culture	**3A-IC-24** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. *(P1.2)*
		3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. *(P1.2)*
		3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. *(P3.1)*
	Social Interactions	**3A-IC-27** Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. *(P2.4)*
	Safety, Law, & Ethics	**3A-IC-28** Explain the beneficial and harmful effects that intellectual property laws can have on innovation. *(P7.3)*
		3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. *(P7.2)*
		3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. *(P7.3)*



Impacts of Computing

3B-IC-25	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.	Culture	6.1, 1.2
3B-IC-26	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	Culture	1.2
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.	Culture	7.2
3B-IC-28	Debate laws and regulations that impact the development and use of software.	Safety Law & Ethics	3.3, 7.3