NEW YORK STATE EDUCATION DEPARTMENT
MIDDLE LEVEL CAREER AND TECHNICAL EDUCATION

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

COMPUTATIONAL THINKING CONTENT MODULE

UPDATED MAY 2023

NYS ED .gov New York State EDUCATION DEPARTMENT Knowledge › Skill › Opportunity

MODULE DESCRIPTION

Computational thinking involves posing and solving problems in ways that can be carried out by a computer. It includes concepts, such as algorithms and variables, and practices, such as abstraction, decomposition, data analysis, modeling, and simulation. These are vital not only to the design and development of computer programs, but also to the strategic use of computational power to solve problems across disciplines.

In this module, students will learn how to use computational thinking to solve problems. They will define and analyze problems by extracting details and repeatable patterns in collected data sets. Modeling and visualization techniques will be employed to assist in design, define needs, and make data more easily understood. Using iterative development, students will break larger problems into smaller, more manageable components. They will learn the fundamentals of writing algorithms, using variables and control flow statements with logical sequencing. Using algorithms, students will produce a solution in the form of pseudocode, flowcharts, or code.

GUIDING QUESTION

What computational thinking concepts and practices can students use to solve problems?

MODULE CONTENT

1. DECOMPOSITION AND ABSTRACTION

STUDENTS WILL:

a) Break down ideas, problems, or processes into components to set the stage for deeper analysis (decomposition)
b) Examine decomposed component parts to find patterns like similarities, repetition, conditional relationships, or nested relationships (pattern recognition)
c) Group component parts by general characteristics with unnecessary details filtered out (generalization)
d) Chain together simple, singular tasks or steps with other processes to accomplish something more complex in order to make it more useful (modularity)
e) Use interfaces to help complete tasks, while hiding details of the overall process

2. MODELING AND VISUALIZATION

STUDENTS WILL:

a) Develop models or simulations to show changes in output when there are changes in inputs
b) Utilize logic-thinking to analyze information related to a task, develop solutions to problems based on facts, and test the effectiveness of a solution and make revisions
c) Use models or visualizations to design and develop a program that includes a variable and control structures for creative expression or solving a problem.
d) Show evidence of the design process that incorporates user feedback and preferences.

3. ALGORITHMS

STUDENTS WILL:

a) Design algorithms with instructions that are general, clear, well-formed, complete, and capable of being executed as intended without confusion
b) Organize steps of an algorithm by logical constructs such as IF statements, loops, and calls to other procedures (control flow)
c) Create variables to store information and understand how data is passed into inputs, processed, and returned thought outputs (inputs, variables, outputs)
d) Understand where, when, why, and how to apply algorithms and which algorithm to apply in a given context (application)

4. PROGRAMMING

STUDENTS WILL:

a) Break down large tasks into smaller pieces
b) Design, develop, and test code in repeated cycles
c) Create applications using an editor to write code, run, and show a program’s output, logging any errors that arise (development environment)
d) Clearly set team roles, save versions along the way, and use parallel versions to manage the complexity of working in a larger algorithm or abstraction (collaboration)
e) Produce solutions in the form of pseudocode, flowcharts, and/or code

5. CAREERS INVOLVING COMPUTATIONAL THINKING

STUDENTS WILL:

a) Explain roles and functions of individuals engaged in careers involving computational thinking
b) Investigate education, training requirements, and opportunities for career paths in computing fields

c) Assess personal employability skills for careers in computing and evaluate personal suitability for such careers

ILLUSTRATIVE ACTIVITIES BY THEME MODULE

These activities are intended to serve as examples of how the content in this module could be tied to each of the six middle level themes.

CAREER AND COMMUNITY OPPORTUNITIES

INTERVIEW A COMPUTING PROFESSIONAL

Students will have a conversation with a professional in the computer science field (virtually or in person). Students prepare by brainstorming questions to ask regarding the computational thinking skills required and to assess their employability skills, personal interests, and suitability for these careers. They will then complete a reflection on the conversation based on the information gathered.

COMMUNICATION AND INTERPERSONAL RELATIONSHIPS

DANCING ALGORITHMS

Students collaborate to write an algorithm for a dance. Algorithms should include loops and control flow statements. Students (and teachers) can run the algorithm in front of the class.

FINANCIAL AND CONSUMER LITERACY

CAREER COSTS AND BENEFITS

Students explore career opportunities and examine the cost-benefits of that career in connection to their lifestyle. Students then produce a model, algorithm, or application to analyze, track, and manage money for future planning.

HEALTH, SAFETY, AND WELLNESS

WELLNESS TRACKERS

Students identify a personal need within the scope of health, safety, and wellness that could be served through a wearable device (i.e., pedometer, smart watch). Students will utilize computational thinking concepts and practices to program a microcontroller (i.e., Micro:bit or Adafruit Circuit Playground Express) that collects data via hardware inputs such as accelerometer, microphone, or other sensors and produces an output to address this need.

PROBLEM SOLVING AND INNOVATION

ALGORITHMS TO SOLVE DAILY PROBLEMS

Students identify a problem that can be solved by developing a program through physical computing (robot, drone, microcontroller). Students will utilize computational thinking and the iterative process to research and brainstorm to develop a solution with an algorithm. Students collaborate with their peers to write the prototype code to complete the task based on their plan and document the program components. Students test, debug, and optimize the program.

SUSTAINABILITY

SUSTAINABLE TECHNOLOGY PURCHASES

Students will examine the technologies they currently use in the classroom (i.e., laptops) and/or their community. They will collect data to understand how and why the technologies are used, evaluate the positive and negative impacts, and create visualizations to persuade the school board or a community organization to include student voice and choice in making sustainable technology purchases.

STANDARDS ADDRESSED

NEW YORK STATE CAREER DEVELOPMENT AND OCCUPATIONAL STUDIES (CDOS) STANDARDS

STANDARD 1: CAREER DEVELOPMENT

Students will be knowledgeable about the world of work, explore career options, and relate personal skills, aptitudes, and abilities to future career decisions

STANDARD 2: INTEGRATED LEARNING

Students will demonstrate how academic knowledge and skills are applied in the workplace and other settings

STANDARD 3A: UNIVERSAL FOUNDATION SKILLS

Students will demonstrate mastery of the foundation skills and competencies essential for success in the workplace

COMMON CAREER TECHNICAL CORE STANDARDS

CAREER READY PRACTICES

1. Act as a responsible and contributing citizen and employee
2. Apply appropriate academic and technical skills
3. Attend to personal health and financial well-being
4. Communicate clearly and effectively with reason
5. Consider the environmental, social, and economic impacts of decisions
6. Demonstrate creativity and innovation
7. Employ valid and reliable research strategies
8. Utilize critical thinking to make sense of problems and persevere in solving them
9. Model integrity, ethical leadership, and effective management
10. Plan education and career paths aligned to personal goals
11. Use technology to enhance productivity
12. Work productively in teams while using cultural global competence

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY STANDARDS

COMPUTATIONAL THINKING

- 4-6.CT.1: Develop a computational model of a system that shows changes in output when there are changes in inputs.
- 7-8.CT.1: Compare the results of alternative models or simulations to determine and evaluate how the input data and assumptions change the results.
- 4-6.CT.2: Collect digital data related to a real-life question or need.
- 7-8.CT.2: Collect and use digital data in a computational artifact.
- 4-6.CT.3: Visualize a simple data set in order to highlight relationships and persuade an audience.
- 7-8.CT.3: Refine and visualize a data set in order to persuade an audience.
- 4-6.CT.4: Decompose a problem into smaller named tasks, some of which can themselves be decomposed into smaller steps.
- 7-8.CT.4: Write a program using functions or procedures whose names or other documentation convey their purpose within the larger task.
- 4-6.CT.5: Identify and name a task within a problem that gets performed multiple times while solving that problem, but with slightly different concrete details each time.
- 7-8.CT.5: Identify multiple similar concrete computations in a program, then create a function to generalize over them using parameters to accommodate their differences.
- 4-6.CT.6: Compare two or more algorithms and discuss the advantages and disadvantages of each for a specific task.
- 7-8.CT.6: Design, compare, and refine algorithms for a specific task or within a program.
- 4-6.CT.7: Identify pieces of information that might change as a program or process runs.

- 7-8.CT.7: Design or remix a program that uses a variable to maintain the current value of a key piece of information.
- 4-6.CT.8: Develop algorithms or programs that use repetition and conditionals for creative expression or to solve a problem.
- 7-8.CT.8: Develop or remix a program that effectively combines one or more control structures for creative expression or to solve a problem.
- 4-6.CT.9: Explain each step of an algorithm or program that includes repetition and conditionals for the purposes of debugging.
- 7-8.CT.9: Read and interpret code to predict the outcome of various programs that involve conditionals and repetition for the purposes of debugging.
- 4-6.CT.10: Describe the steps taken and choices made to design and develop a solution using an iterative design process.
- 7-8.CT.10: Document the iterative design process of developing a computational artifact that incorporates user feedback and preferences.

NETWORKS AND SYSTEMS DESIGN

- 4-6.NSD.1: Propose improvements to the design of a computing technology based on an analysis of user interactions with that technology.
- 7-8.NSD.1: Design a user interface for a computing technology that considers usability, accessibility, and desirability.
- 4-6.NSD.2: Model how computer hardware and software work together as a system to accomplish tasks.
- 7-8.NSD.2: Design a project that combines hardware and software components.
- 4-6.NSD.3: Determine potential solutions to solve hardware and software problems using common troubleshooting strategies.
- 7-8.NSD.3: Identify and fix problems with computing devices and their components using a systematic troubleshooting method or guide.

RESOURCES

Disclaimer: Posting of resources on this form does not constitute an endorsement from the New York State Education Department nor does it imply that the following resources are mandatory or the only ones that can be used. Teachers and administrators ensure that resources align with local policies and are responsible for choosing the resources and have the final authority, in alignment with local policies, to choose and utilize the resources that best meet the needs of their students. Questions regarding compliance with Education Law 2D should be directed to your administrator and/or chief information officer.

NYS COMPUTER SCIENCE AND DIGITAL FLUENCY (CS&DF) LEARNING STANDARDS

http://www.nysed.gov/curriculum-instruction/computer-science-and-digital-fluency-learning-standards

This webpage contains multiple resources for the CS&DF Standards. Including an excel version of the standards, glossary of terms, standards examples, as well as At-a-Glance documents by grade band.

NYS SMART START GRANT RESOURCES

http://www.nysed.gov/edtech/smart-start-grant-program

This resource includes teacher-created artifacts curated through the Smart Start Grant cohorts that focus on computer science, engineering, and educational technology.

NYS LEARNING TECHNOLOGY GRANT (LTG) RESOURCES

http://www.nysed.gov/edtech/learning-technology-grants-ltg

Resources include artifacts curated by multiple grant recipient districts that include personalized learning, technology integration, STEM activities, blended learning, computer science, and more.

CAREER AND TECHNICAL EDUCATION TECHNICAL ASSISTANCE CENTER OF NEW YORK

http://nyctecenter.org/

The Career and Technical Education Technical Assistance Center (CTE TAC) operates under a state contract to assist the New York State Education Department (NYSED) in carrying out its mission of improving the quality, access, and delivery of career and technical education through research-based methods and strategies resulting in broader CTE opportunities for all students.