Randolph Township Schools
Randolph Middle School

Introduction to Programming Curriculum

"Technology is best when it brings people together."

-Matt Mullenweg

Department of Science, Technology, Engineering, and Math
Gabriel Maffei
Supervisor

Curriculum Committee
Ralph Scimeca
Ned Sheehy

Curriculum Developed:
July 2018

Date of Board Approval:
September 4, 2018

Randolph Township Schools
Department of Science, Technology, Engineering, and Mathematics
Introduction to Programming

Table of Contents

Randolph Township Schools

Mission Statement

We commit to inspiring and empowering all students in Randolph schools to reach their full potential as unique, responsible and educated members of a global society.

Randolph Township Schools
Affirmative Action Statement

Equality and Equity in Curriculum

The Randolph Township School district ensures that the district's curriculum and instruction are aligned to the state's standards. The curriculum provides equity in instruction, educational programs and provides all students the opportunity to interact positively with others regardless of race, creed, color, national origin, ancestry, age, marital status, affectional or sexual orientation, gender, religion, disability or socioeconomic status.

N.J.A.C. 6A:7-1.7(b): Section 504, Rehabilitation Act of 1973; N.J.S.A. 10:5; Title IX, Education Amendments of 1972

RANDOLPH TOWNSHIP BOARD OF EDUCATION
EDUCATIONAL GOALS
VALUES IN EDUCATION

The statements represent the beliefs and values regarding our educational system. Education is the key to self-actualization, which is realized through achievement and self-respect. We believe our entire system must not only represent these values, but also demonstrate them in all that we do as a school system.

We believe:

- The needs of the child come first
- Mutual respect and trust are the cornerstones of a learning community
- The learning community consists of students, educators, parents, administrators, educational support personnel, the community and Board of Education members
- A successful learning community communicates honestly and openly in a non-threatening environment
- Members of our learning community have different needs at different times. There is openness to the challenge of meeting those needs in professional and supportive ways
- Assessment of professionals (i.e., educators, administrators and educational support personnel) is a dynamic process that requires review and revision based on evolving research, practices and experiences
- Development of desired capabilities comes in stages and is achieved through hard work, reflection and ongoing growth

Randolph Township Schools
Department of Science, Technology, Engineering, and Mathematics
Introduction to Programming

Introduction

In Introduction to Programming, 7th and 8th grade students will be introduced to computational thinking skills to help develop knowledge of how to deconstruct problems into smaller pieces. Computational thinking stretches far beyond learning coding concepts or how computers work. Students will learn problem-solving methods that require logic, sequencing, and trial and error. According to the International Society of Technology Education (ISTE) and the Computer Science Teachers of America, computational thinking also encourages perseverance, self-assurance, and broad-mindedness for uncertainty. Shuchi Grover, an education researcher who studies STEM in K-12 with Stanford University and the nearby SRI International Institute, says coding uses computational thinking because it requires modeling, using if-then logic, testing, debugging, and retesting.

Key components to this class include:

- organizing and analyzing data
- representing data through models and simulations
- automating solutions through algorithms (a series of ordered steps)
- generalizing the problem-solving process to apply it elsewhere

These concepts also transfer to other, non-computer science content areas. During coding lessons, the focus will be placed as much on initial ideologies, such as patterns and reason, as on coding language and technical skills. Coding will also be used as a method of communication. Through this course students will create prototype apps, collages, and websites that reflect their beliefs and understanding on subjects that interest them.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Curriculum Pacing Chart
Introduction to Programming

SUGGESTED TIME ALLOTMENT	UNIT NUMBER	CONTENT - UNIT OF STUDY
2 weeks	**I**	**Computational Problem Solving**
4 weeks	**II**	**Interactive Games and Animation**
3 weeks	**III**	**Web Development**

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT I: Computational Problem Solving

<table>
<tr><td rowspan="8">STANDARDS / GOALS:

<u>NJCCCS Technology:</u>
8.1.8.A.1 Demonstrate knowledge of a real world problem using digital tools.

8.1.8.A.3 Use and/or develop a simulation that provides an environment to solve a real world problem or theory.

8.1.8.D.4 Assess the credibility and accuracy of digital content.

8.1.8.D.5 Understand appropriate uses for social media and the negative consequences of misuse.

8.1.8.E.1 Effectively use a variety of search tools and filters in professional public databases to find information to solve a real world problem.

8.2.8.A.2 Examine a system, consider how each part relates to other parts, and discuss a part to redesign to improve the system.

8.2.8.C.1 Explain how different teams/groups can contribute to the overall design of a product.

8.2.8.E.2 Demonstrate an understanding of the relationship between hardware and software.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Effective problem solving involves efficient processes and strategies.</td><td>• What strategies and processes can be used to become a more effective problem solver?</td></tr>
<tr><td>To solve a problem, computers require data along with a defined function.</td><td>• What does a computer need to solve problems effectively?</td></tr>
<tr><td>Computing systems solve problems using algorithms and have defined inputs and outputs.</td><td>• How do people and computers approach problems differently?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Processes such as defining, preparing, trying, and reflecting can be useful in the task of problem solving.

Strategies can make solving a problem more efficient.

A computer is a device that requires an input and solves a task by providing an output.</td><td>Students will be able to:

Examine different processes involved in problem solving.

Recall strategies to help solve problems effectively.

Define a computer and identify past and present devices that fit the definition. ex: abacus, calculator</td></tr>
</table>

8.2.8.E.3 Develop an algorithm to solve an assigned problem using a specified set of commands and use peer review to critique the solution. **8.2.8.E.4** Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms). **CSTA:** **1B-AP-08** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. **2-AP-10** Use flowcharts and/or pseudocode to address complex problems as algorithms. **1B-AP-11** Decompose (break down) problems into smaller, manageable sub-problems to facilitate the program development process. **2-AP-15** Seek and incorporate feedback from team members and users to refine a solution that meets user needs. **1B-AP-16** Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation and review stages of program development. **2-AP-17** Systematically test and refine programs using a range of test cases. **2-AP-18** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	An input is any data or resource a computer uses while an output is any message or product created while solving a problem. Complex and difficult to define problems require problem-solving processes to be solved efficiently. Processing occurs when a computer turns input information into output information and is essential to computational problem solving. An algorithm is a set of instructions a computer uses to process information. Correctly processing information requires a detailed understanding of the relationship of the input compared to the output along with the criteria and constraints of the process. Software applications can solve a defined problem by processing different inputs to desired outputs through the use of brainstorming, defining, preparing, trial, and reflection.	Identify inputs and outputs of several devices. Describe different situations where problem-solving processes are helpful. Illustrate what processing means and how it works. Construct an algorithm to complete a simple task. Evaluate a more complex process and arrange an algorithm to process inputs into a desired output. Develop inputs and outputs required to solve a real-world application problem and produce an application proposal that will solve a personal and relevant problem.

1B-CS-01 Describe how internal and external parts of computing devices function to form a system. **1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. **2-IC-20** Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	**VOCABULARY AND KEY TERMS:** Defining, Preparing, Trying, Reflecting, Computer, Input, Output, Processing, Algorithm, Problem-Solving Strategies, Problem-Solving Processes, Computational, Problem Solving	

KEY LEARNING EVENTS AND INSTRUCTION:

- Students will explore problem-solving processes and their application to computer systems.
- Students will analyze the relationships between a computer system's input and output information.
- Students will create a prototype application utilizing inputs, outputs, and algorithms.

ASSESSMENT EVIDENCE: *Students will show their learning by the following, including, but not limited to:*

- Do now
- Reflection worksheets
- Processing handout
- Application proposal poster

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT I: Computational Problem Solving

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
2 Weeks	**UNIT I: Computational Problem Solving** Problem Solving Process Computational Processing and Algorithms Application Proposal	BOOKS: Readings taken from various relevant sources Suggested Supplies: Computers Measuring and drafting tools: rulers, triangle straight edges, graphing paper Various objects from around the technology classroom Suggested Activities: Problem Solving Challenge Activity Using the Problem Solving Process What is a Computer? Inputs and Outputs Storage and Processing Propose an App

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT II: Interactive Animations and Games

<table>
<tr><td rowspan="6">STANDARDS / GOALS:

<u>NJCCCS Technology:</u>
8.1.8.A.1 Demonstrate knowledge of a real world problem using digital tools.

8.1.8.C.1 Collaborate to develop and publish work that provides perspectives on a global problem for discussions with learners from other countries.

8.1.8.D.1 Understand and model appropriate online behaviors related to cyber safety, cyber bullying, cyber security, and cyber ethics including appropriate use of social media.

8.1.8.D.2 Demonstrate the application of appropriate citations to digital content.

8.1.8.D.3 Demonstrate an understanding of fair use and Creative Commons to intellectual property.

8.1.8.D.4 Assess the credibility and accuracy of digital content.

8.1.8.D.5 Understand appropriate uses for social media and the negative consequences of misuse.

8.2.8.A.2 Examine a system, consider how each part relates to other parts, and discuss a part to redesign to improve the system.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Computer systems can be used to store, retrieve, and modify data and graphics.</td><td>• How can a computing device be used to assist individuals, society, or the world?</td></tr>
<tr><td>Computer software can be used to model and teach individuals how to perform a complex task.</td><td>• How can an individual program a computer to teach others a task?</td></tr>
<tr><td>Computer software can benefit the user and creator in various ways such as entertainment.</td><td>• How can programming promote positivity and empathy through designing entertainment applications?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Different software applications exist to perform various tasks.

Each software application is designed for a specific task; similar applications can complete similar tasks with different tools/manufacturers.</td><td>Students will be able to:

Identify a task and find the appropriate software application for the task.

Classify different applications by their function and identify which software application to use for different products.</td></tr>
</table>

8.2.8.A.3 Investigate a malfunction in any part of a system and identify its impacts. **8.2.8.A.5** Describe how resources such as material, energy, information, time, tools, people, and capital contribute to a technological product or system. **8.2.8.B.2** Identify the desired and undesired consequences from the use of a product or system. **8.2.8.A.2** Examine a system, consider how each part relates to other parts, and discuss a part to redesign to improve the system. **8.2.8.C.1** Explain how different teams/groups can contribute to the overall design of a product. **8.2.8.C.2** Explain the need for optimization in a design process. **8.2.8.C.3** Evaluate the function, value, and aesthetics of a technological product or system, from the perspective of the user and the producer. **8.2.8.C.6** Collaborate to examine a malfunctioning system and identify the step-by-step process used to troubleshoot, evaluate and test options to repair the product, presenting the better solution. **8.2.8.E.1** Identify ways computers are used that have had an impact across the range of human activity and within different careers where they are used.	Tools in Scratch are used to manipulate and create sprites (characters) along with modifying elements. The background of the scene is where all sprites are located and what the user views. When a sprite is created, it can be moved around the stage and its properties can be modified. Scripts are programs written in Scratch that control the functions of the application. Script commands are divided into different sections based on its function. Scripts are executed by using the 'Events' functions. The direction of the code is directed by the 'Control' functions such as 'if then', 'repeat', and 'forever'. Scratch can sense objects in an application such as determining if it comes in contact with another Sprite or sense the color of objects. Variables are places to store data which Scratch can use to perform calculations by using logic and conditional statements to modify numbers. Stages change the view a user sees and can expand the scope and add complexity to a program. Costumes are different appearances for sprites; when changed in sequence they create the illusion of motion.	Define the various tools in Scratch and identify their use. Create a unique background by using the drawing tools. Construct a sprite. Change the look or motion of a sprite by executing an Event. Change the direction of the code by using a conditional statement. Program a sprite to interact with another sprite. Create a variable that is incremented each time an action takes place. Change stages when an action takes place. Make a sprite animate by coding different costumes.

8.2.8.E.2 Demonstrate an understanding of the relationship between hardware and software. **8.2.8.E.3** Develop an algorithm to solve an assigned problem using a specified set of commands and use peer review to critique the solution. **8.2.8.E.4** Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms).	Tutorials in Scratch can help introduce concepts and skills. Programs can help communicate information and ideas. **VOCABULARY AND KEY TERMS:** Sprite, Costume, Event, Control, Script, Motion, Looks, Variable, 'if then else', Wait, Operators, Data, Conditional Statements, Loops, Events, Debugging	Compare being guided through a set tutorial with having an undefined goal. Plan and create an interactive collage.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students will engage in exploratory programing activities to build coding skills utilizing the Scratch platform.
- Students will construct creative programs to perform a defined task.

ASSESSMENT EVIDENCE: *Students will show their learning by the following, including, but not limited to:*

- Completing tutorials on software applications
- Completing matching activity with software and function
- Coding a Sprite to perform simple tasks
- Coding a Sprite to show different looks
- Coding a Sprite to perform different tasks using Controls

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT II: Interactive Games and Animation

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
4 Weeks	**UNIT II: Interactive Games and Animation** Introducing Scratch Scratch Challenges: Exploring Scratch Challenges: Animations Scratch Challenges: Stories and Games	<u>BOOKS:</u> Readings taken from various relevant sources <u>Suggested Supplies:</u> Computers Scratch 2.0 desktop application Measuring and drafting tools: rulers, triangle straight edges, graphing paper Various objects from around the technology classroom <u>Suggested Activities:</u> Design Journal Scratch Surprise Critique Group Step-By-Step 10 Block Debug it! About Me Orange Square, Purple Circle It's Alive Extensions

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT III: Web Development

<table>
<tr><td rowspan="8">STANDARDS / GOALS:

<u>NJCCCS Technology:</u>

8.1.8.A.1 Demonstrate knowledge of a real world problem using digital tools.

8.1.8.A.3 Use and/or develop a simulation that provides an environment to solve a real world problem or theory.

8.1.8.D.1 Understand and model appropriate online behaviors related to cyber safety, cyber bullying, cyber security, and cyber ethics including appropriate use of social media.

8.1.8.D.4 Assess the credibility and accuracy of digital content.

8.1.8.D.5 Understand appropriate uses for social media and the negative consequences of misuse.

8.1.8.E.1 Effectively use a variety of search tools and filters in professional public databases to find information to solve a real world problem.

8.2.8.A.2 Examine a system, consider how each part relates to other parts, and discuss a part to redesign to improve the system.

8.2.8.C.1 Explain how different teams/groups can contribute to the overall design of a product.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>The Web (Internet and Cloud Storage) is a place where users can share information instantly across large distances.</td><td>• How can information shared instantly change the world around us?</td></tr>
<tr><td>Coding languages describe in a standard language what and how information is displayed and processed.</td><td>• How can the implementation and development of a coding language change the way it is used?</td></tr>
<tr><td>Adding pleasing ascetics and functionally practical designs can improve user experience and increase interest.</td><td>• How can text, images, and colors communicate content and structure on a web page?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Websites are content pages that are shared for the benefit of the user who is viewing the page along with the creator who constructed the page.

HTML is a coding language used to put content and structure into webpages.</td><td>Students will be able to:

Describe how commonly used websites benefit both user and creator.

Identify problems HTML solves.</td></tr>
</table>

8.2.8.C.3 Evaluate the function, value, and aesthetics of a technological product or system, from the perspective of the user and the producer. **8.2.8.C.6** Collaborate to examine a malfunctioning system and identify the step-by-step process used to troubleshoot, evaluate and test options to repair the product, presenting the better solution. **8.2.8.E.1** Identify ways computers are used that have had an impact across the range of human activity and within different careers where they are used. **8.2.8.E.2** Demonstrate an understanding of the relationship between hardware and software. **8.2.8.E.4** Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms).	Webpages are the final product for viewing while the HTML that builds the pages describes each element that will be viewed in the page with tags. Tags describe what type of content will be displayed; a heading is a type of content. Lists can be created in HTML using both ordered and unordered lists. As the code on a webpage develops, it must be organized in order to be efficiently read and edited. Strategies such as using comments, separating similar areas with whitespace, and indenting subgroups can make code more user friendly for editing purposes and less prone to errors. Web links enable users to access other pages including other websites. A home page and subpages are constructed and moved into the same folder; using web links, these pages can be quickly accessed with the use of a navigation bar. Media that is not personally created, must be free to use, and be given proper attribution on webpages. Personal information online can be accessed by anyone; information such as addresses and phone numbers should never be shared while other information can be.	Compare webpages with its HTML. Manipulate HTML tags to construct a heading. Categorize information and create HTML lists. Rearrange and edit code to avoid errors and implement comments, whitespace, and indentations. Assemble links to various subpages and other relevant websites. Create a multipage website to express interests and ideas. Originate where media came from and integrate proper attribution on webpages. Identify and explain what personal information is safe to share digitally and what information may be hazardous.

	VOCABULARY AND KEY TERMS: HTML, Tags, Links, Element, Comments, Whitespace, Indentations, Navigation Bar, Error, Developer, Content, Structure, Information Sharing, User Friendly, Web Links	

KEY LEARNING EVENTS AND INSTRUCTION:

- Students will utilize HTML tutorials to analyze website's code.
- Students will create personalized websites utilizing HTML and CSS code.

ASSESSMENT EVIDENCE: *Students will show their learning by the following, including, but not limited to:*

- Do Now
- Reflection worksheet
- Code.org tutorial and lessons
- Building a multipage personal website

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Introduction to Programming
UNIT III: Web Development

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
3 Weeks	**UNIT III: Application and Animation Design** Common Websites and Their Uses HTML & CSS Personal Website	<u>BOOKS:</u> Readings taken from various relevant sources <u>Suggested Supplies:</u> Computers Notepad Notepad++ Measuring and drafting tools: rulers, triangle straight edges, graphing paper Various objects from around the technology classroom <u>Suggested Activities:</u> Exploring Websites Websites for Expression Intro to HTML Headings Digital Footprint Lists Intellectual Property and Images Clean Code and Debugging Multi-Page Websites

APPENDIX A

Resources:

Suggested Web Addresses:

www.google.com
www.code.org
scratch.mit.edu
www.w3schools.com

Software Names:

Microsoft Word
Microsoft PowerPoint
Microsoft Excel
Scratch 2.0
Code.org Web Application
Notepad
Notepad++
Internet Sources

APPENDIX B

Assessments

Propose an App

Checklist for Propose an App

Criteria	Yes/No	Comments
The problem is well-defined, including a target audience, details of the problem, and how to tell it has been solved.		
The app is clearly described including what it does and why someone would use it.		
The outputs of the app are clearly described and could be used to address the problem.		
The inputs of the app could be used to produce the outputs.		
The way that the information is processed to produce the output is clearly described.		
Stored information is listed and is appropriate for the functionality of the app.		
The peer review provides useful and constructive feedback.		
Peer review feedback has clearly been incorporated into the final version of the project.		
The final presentation includes all information required by the project guide.		

Personal Website

Checklist for Personal Website

Personal website:	Yes/No	Comments
Pages		
There are at least three web pages in the site.		
Every page contains DOCTYPE, <html> element, <head> element, and <body> element.		
All text in the page is contained inside elements.		
Images		
There are at least three images.		
All images have an alt attribute.		
All images have attribution underneath the image with available information about the author, title, and source.		
Image include information about the license it was published under either in text (eg CC-BY) or using a badge from the Creative Commons website.		
Headers and Footers		
Each page contains a header with the page title and a navigation bar that links to every other page in the site.		
Each page contains a footer that tells users the license under which the website is published.		
Element Types		
There is at least one list.		
There are at least two different sizes of headings.		
There is at least one paragraph.		
Safety and Responsibility		
The website does not give away any personally identifiable information (Full name, address, etc.)		
All content from outsides sources is cited.		