Glen Ridge Public Schools – Computer Science Curriculum

Course Title: Computer Science 1 (CS1)

Subject: Computer Science

Grade Level: 9 - 12

Duration: 0.5 year (approx. 18 weeks)

Prerequisite: None

Elective or Required: Elective

Computer Science, Engineering, and Technology Mission Statement

Since computational thinking and problem solving are integral parts of our lives and 21st century learning, students must be actively involved in their Computer Science, Engineering, and Technology (CSET) education. The CSET curricula will emphasize thinking skills through a balance of computation, intuition, common sense, logic, design, analysis, and technology. Students will use a combination of technology and critical thinking to solve real-world problems. To achieve these goals, students will be taught a standards-based curriculum that is aligned with the New Jersey Curriculum Standards.

Course Description:

Computer Science 1 is designed to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. Rather than focusing the entire course on learning particular software tools or programming languages, the course is designed to focus on the conceptual ideas of computing and help students understand why certain tools or languages might be utilized to solve particular problems.

In Computer Science 1, students will explore such varied topics as programming (largely in Scratch, a block-based programming language, but to a lesser degree in Python, a text-based programming language), web design, hardware/software, problem solving (algorithms), and the implications of computing on society.

Author: Kevin George

Date Submitted: Summer 2016
Resources

Due to the changing nature of computer programming and computer science, a paper textbook is not used for this course. The following online resources are used throughout this course:

- Scratch tutorials: [http://info.scratch.mit.edu/Support](http://info.scratch.mit.edu/Support)
- Scratch cards: [http://info.scratch.mit.edu/Support/Scratch_Cards](http://info.scratch.mit.edu/Support/Scratch_Cards)
- University of California at Berkeley “The Beauty and Joy of Computing” curriculum
- Google’s Python Class: [https://developers.google.com/edu/python/](https://developers.google.com/edu/python/)
- Glen Ridge High School Faculty pages: [http://www.glenridge.org/grhs/faculty.html](http://www.glenridge.org/grhs/faculty.html)

Software Tools

- Google Chrome (or other web browser, such as Safari, Internet Explorer, etc)
- Notepad
- Scratch (scratch.mit.edu)
- jEdit
- Internet access
## Curriculum Standards

### Technology

<table>
<thead>
<tr>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.12.A.1</td>
</tr>
<tr>
<td>8.1.12.B.2</td>
</tr>
<tr>
<td>8.1.12.D.1</td>
</tr>
<tr>
<td>8.1.12.D.2</td>
</tr>
<tr>
<td>8.1.12.D.3</td>
</tr>
<tr>
<td>8.1.12.D.4</td>
</tr>
<tr>
<td>8.1.12.D.5</td>
</tr>
<tr>
<td>8.1.12.E.1</td>
</tr>
<tr>
<td>8.1.12.E.2</td>
</tr>
<tr>
<td>8.1.12.F.1</td>
</tr>
<tr>
<td>8.2.12.A.1</td>
</tr>
<tr>
<td>8.2.12.A.2</td>
</tr>
<tr>
<td>8.2.12.B.1</td>
</tr>
<tr>
<td>8.2.12.B.3</td>
</tr>
<tr>
<td>8.2.12.B.4</td>
</tr>
<tr>
<td>8.2.12.B.5</td>
</tr>
<tr>
<td>8.2.12.C.1</td>
</tr>
<tr>
<td>8.2.12.C.2</td>
</tr>
<tr>
<td>8.2.12.C.4</td>
</tr>
<tr>
<td>8.2.12.C.7</td>
</tr>
<tr>
<td>8.2.12.D.1</td>
</tr>
<tr>
<td>8.2.12.D.3</td>
</tr>
<tr>
<td>8.2.12.D.6</td>
</tr>
<tr>
<td>8.2.12.E.1</td>
</tr>
<tr>
<td>8.2.12.E.2</td>
</tr>
<tr>
<td>8.2.12.E.3</td>
</tr>
<tr>
<td>8.2.12.E.4</td>
</tr>
</tbody>
</table>
### 21<sup>st</sup> Century Life and Careers

<table>
<thead>
<tr>
<th>9.3.IT-PRG.1</th>
<th>Analyze customer software needs and requirements.</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.IT-PRG.2</td>
<td>Demonstrate the use of industry standard strategies and project planning to meet customer specifications.</td>
</tr>
<tr>
<td>9.3.IT-PRG.3</td>
<td>Analyze system and software requirements to ensure maximum operating efficiency.</td>
</tr>
<tr>
<td>9.3.IT-PRG.4</td>
<td>Demonstrate the effective use of software development tools to develop software applications.</td>
</tr>
<tr>
<td>9.3.IT-PRG.5</td>
<td>Apply an appropriate software development process to design a software application.</td>
</tr>
<tr>
<td>9.3.IT-PRG.6</td>
<td>Demonstrate the effective use of software development tools to develop software applications.</td>
</tr>
<tr>
<td>9.3.IT-PRG.7</td>
<td>Perform quality assurance tasks as part of the software development cycle.</td>
</tr>
</tbody>
</table>

### Mathematics

| Math Practices.1 | Make sense of problems and persevere in solving them. |
| Math Practices.2 | Reason abstractly and quantitatively. |
| Math Practices.3 | Construct viable arguments and critique the reasoning of others. |
| Math Practices.4 | Model with mathematics. |
| Math Practices.5 | Use appropriate tools strategically. |
| Math Practices.6 | Attend to precision. |
| Math Practices.7 | Look for and make use of structure. |
| Math Practices.8 | Look for and express regularity in repeated reasoning. |

### Social Studies

| 6.3.12.D.1 | Analyze the impact of current governmental practices and laws affecting national security and/or individual civil rights/privacy |

### Visual and Performing Arts

| 1.1.12.D.2 | Translate literary, musical, theatrical, and dance compositions by using them as stimulus/inspiration for corresponding visual artworks. |

### English Language Arts

<p>| NJSLSA.R1 | Read closely to determine what the text says explicitly and to make logical inferences and relevant connections from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text. |
| NJSLSA.R2 | Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas. |
| NJSLSA.R4 | Interpret words and phrases as they are used in a text, including determining technical, connotative, and figurative meanings, and analyze how specific word choices shape meaning or tone. |
| NJSLSA.R7 | Integrate and evaluate content presented in diverse media and formats, including visually and quantitatively, as well as in words. |
| RI.11-12.4 | Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text (e.g., how Madison defines faction in Federalist No. 10). |
| RI.11-12.7 | Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem. |</p>
<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJSLSA.W4</td>
<td>Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.</td>
</tr>
<tr>
<td>NJSLSA.W6</td>
<td>Use technology, including the Internet, to produce and publish writing and to interact and collaborate with others.</td>
</tr>
<tr>
<td>NJSLSA.W9</td>
<td>Draw evidence from literary or informational texts to support analysis, reflection, and research.</td>
</tr>
<tr>
<td>W.11-12.4</td>
<td>Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.)</td>
</tr>
<tr>
<td>W.11-12.6</td>
<td>Use technology, including the Internet, to produce, share, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.</td>
</tr>
</tbody>
</table>
Unit 1: Human-Computer Interaction

Approximate # Of Weeks: 1.5

Essential Questions:
- What is a computer?
- What is computing?
- What is the difference between hardware and software, and what are examples of each?
- What are the dangers of personal data being put online?
- What is a computer program?
- How is technology developed?
- What makes technology change?

Upon completion of this unit students will be able to:
- Explain and give examples of the concepts of computer and computing.
- Describe the uses for computer hardware components
- Choose hardware components for various types of users.
- Explain how computers are used for communications
- Recognize various forms of communication as data exchange
- Describe the implications of data exchange on social interactions
- Consider privacy of data that they create.
- Explain the concept of a computer program.
- Explain how technology is developed and changes over time to meet human needs and wants.

Interdisciplinary Standards
- 8.1.12.D(2-4); 8.1.12.E.2; 8.1.12.F.1; 8.2.12.A(1-2); 8.2.12.B(1, 4-5); 8.2.12.C(2, 4); 8.2.12.D.6; 8.2.12.E(2, 4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Class Activity: Start with a journal entry on, “How Many Computers are in the Room?” Have students write responses to the question in a journal and then share the response with their neighbor. Ask a few student pairs to share their responses. Some students may have counted only the desktop computers in the room while others may have recognized that there are other items that are computers as well. After this discussion, have students write further examples in their journal and the class will discuss again. (Examples of computers include: Macintosh, Windows PC, cell phone, mp3 player, most appliances (television, coffee maker, washer, dishwasher, etc.), cars, medical
equipment, planes, watches, cash registers, ATMs, traffic lights, scoreboards, humans, and calculators.)

- **Class Discussion:** What is a computer? What is computing?
- **Project:** Computer Buying Project Assignment: Each student will interview a family member or friend to find out what features that person would like to have if they were buying a new personal computer (e.g., What will be the uses of the computer? What are the space constraints? What is the price range? etc.). Students will investigate computer buying options and their components (e.g., Processor, Operating System, Memory, Hard Drive, Optical Drive, Monitor, Video Card, Sound Card, Speakers, Keyboard, Mouse, Modem). Then students will give them at least 4 options and then give them advice on which one to buy. The final product can be a: PowerPoint, Debate, Skit, Video, or Other approved product.

- **Class Activity:** Communication Methods Activity: Students complete Communication Methods Chart. For each of a set of examples (e.g., "Breaking up with a significant other (boyfriend/girlfriend)", "Asking parents' permission to do something when you think they will likely say 'no'", "Figure out where and when to meet a friend to see a movie", etc.), students will fill in a table with which method of communication (e.g., Texting, Phone call, Talking in person, Facebook, Twitter, Email) they would choose for the given scenario and why. Teacher will take a poll for each category to see what form of communication is most popular for each scenario, and lead a class discussion.

- **Project:** Privacy Activity: This activity is designed to make students examine the data they "give off" at all times. Students will read a scenario assigned to you (many of these are based on real stories: "A boss sees an employee who called in 'sick' in a picture that someone posted on Facebook. In this picture the employee is partying the night before. The boss fires the employee.", "A company who has contracts with the Federal Government doesn’t want to hire you because a Facebook friend leaves lots of enthusiastic 'legalize marijuana' postings on your wall." Students will prepare a 3-5 minute presentation for the class that includes answers to the following: In your scenario, did the people have the right to use the information they did or should it have been private? Are the people who were affected by the use of this information at fault? What might other people think about who you are based on the data you’ve made available in different places/spaces such as Facebook, Twitter, texting, Netflix, email, at the grocery store, etc.? Is it an accurate impression of who you are?

- **Activity:** Algorithm Writing: Teacher will split the class into pairs. Each member will write an algorithm for an everyday task (tying shoes, sharpening a pencil, doing jumping jacks, etc.). Then, partners will
perform each other’s algorithms literally, spotlighting the flaws and ambiguities in the algorithm, while allowing the writer to make corrections.

- Project: Development and Evolution of Technology: In groups, students will pick a technology and write a short paper (a couple paragraphs) on the invention and development of the technology over time, and changes made to meet the needs of consumers. Class discussion.
- Unit test and quiz.

**Enrichment Activities:**
- Students will explore the concept of Artificial Intelligence. They will experiment with various chatbots and determine which come closest to passing the Turing Test.

**Methods of Assessments/Evaluation:**
- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources/Including Online Resources**
- Teacher Webpage
- Resources listed above in *Resources* section
Unit 2: Problem Solving

Approximate # Of Weeks: 2

Essential Questions:
- What are the steps to solving a problem?
- What is an algorithm?
- What is binary and why is it important in computer science?
- What are efficient algorithms for searching through data?
- What are efficient algorithms for sorting data?
- What is a minimal spanning tree and why is it important in computer science?

Upon completion of this unit students will be able to:
- Name and explain the steps in the problem-solving process
- Solve a problem by applying the problem-solving process
- Explain what the word algorithm means.
- Express a solution using standard design tools
- Determine if a given solution successfully solves a stated problem
- Count forward and backward in binary
- Explain why binary numbers are important in computer science
- Describe the linear search algorithm
- Describe the binary search algorithm
- Explain conditions under which each search (linear or binary) might be appropriate
- Define sorted and unsorted lists
- Describe various sorting algorithms
- Compare various sorting algorithms
- Solve a minimal spanning tree
- Draw a graph to solve a problem

Interdisciplinary Standards
- 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Class Activity: Candy Bar Activity: Divide students into pairs. Give each group a candy bar. Explain that their task is to determine how many "breaks" it will take to break the candy bar into 12 equal pieces. Demonstrate the process by: 1) breaking the bar into two pieces, 2) then stacking the two pieces together and breaking or cutting the two pieces into four. Students write the number of breaks they think it will
take to break the bar into 12 equal pieces and then test their answer. Now, extend breaking the candy into N pieces.

- **Lecture:** Teacher instructs on the 4 steps to solving a problem: 1) Understand the problem; 2) Make a plan to solve the problem—use pictures, charts, graphs, systematic lists, objects, or act out the solution to help you devise a plan to solve the problem (an algorithm); 3) Carry out the plan—one the plan is conceived and understood, follow the plan (if you have planned well, this is the easy part!); 4) Review and reflect on how the problem was solved.

- **Journal Entry:** Handshake and Fencepost Activity: Follow worksheet to administer class handshake and fencepost problems (i.e., how many handshakes would it take to for \( n \) people in a room to shake hands once and only once with each other person in the room, etc.)

- **Class Activity:** CS Unplugged Count the Dots Activity (Binary): Follow packet (downloaded from [http://csunplugged.com](http://csunplugged.com)) to lead this activity where students will learn to count in binary.

- **Class Activity:** Tower Building Activity: Donald Trump wants to build a 100 meter high tower as quickly as possible. He has unlimited resources and an unlimited budget and is willing to spend any amount to get the job done. He has chosen to build the tower with blocks that are 1 meter tall. What is the quickest way to build this tower? After students attempt, teacher will use Legos or checker pieces to model the quickest solution (foreshadows binary vs linear search).

- **Class Activity:** Linear vs Binary Search: Look up a word in the dictionary using linear and then binary search. Which is quicker?

- **Class Activity:** CS Unplugged Lightest and Heaviest Activity (Sorting): Follow packet (downloaded from [http://csunplugged.com](http://csunplugged.com)) to lead this activity where students will learn about sorting algorithms. And adjustment could (and probably should) be made to use numbered cards instead of jars of sand.

- **Class Activity:** CS Unplugged Muddy City Activity (Minimal Spanning Trees): Follow packet (downloaded from [http://csunplugged.com](http://csunplugged.com)) to lead this activity where students will learn about minimal spanning trees.

- **Unit test and quiz.**

**Enrichment Activities:**

- **Journal Entry:** Students will do further exercises in the CS Unplugged Count the Dots packet (Sending Secret Messages; Email and Modems; Counting Higher than 31; More on Binary Numbers).

- **Journal Entry:** Students will do further reading in the CS Unplugged Lightest and Heaviest packet (about the other kinds of sorts) and compare each of the sorts.
• Lab: Work with a partner to make graphs for which your partner must develop a minimal spanning tree.

Methods of Assessments/Evaluation:
• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources/Including Online Resources
• Candy Bars
• Legos/Checkers
• CS Unplugged Packets (Count the Dots; Lightest and Heaviest; Muddy City)
• Teacher Webpage
• Resources listed above in Resources section
Unit 3: Programming in Scratch (7 Sub-Units)

Approximate # Of Weeks: 8.5

Unit 3.1: Introduction to the Scratch Environment

Approximate # Of Weeks: 0.5

Essential Questions:
- What is programming?
- What are the main sections of the Scratch development environment and how are they used?
- How is a Scratch project created, saved, modified and executed?
- What are sprites, blocks, and stacks?
- What are the geometric properties of the Stage?
- What are the main categories and types of blocks in the Scratch toolbox?
- What is a script and how is it created and used?
- How can motion blocks be used to animate a sprite?
- How can looks blocks be used to add user interaction?
- How are the “When clicked”, “When key pressed”, “Broadcast” and “repeat” control blocks used?
- What is a comment and why is it useful?
- How is the “Ask User block in the “sensing tab used to get information from the user?

Time Permitting:
- How can sound blocks be used to add music and sound to a Scratch project?
- How can pen blocks be used to draw lines, shapes and text?
- How can the paint editor be used to create or modify sprites?

Upon completion of this unit students will be able to:
- Create, modify and execute applications using Scratch.
- Understand and use the main sections of the Scratch development environment.
- Describe the steps for retrieving, modifying and saving a Scratch program.
- Explain and use terminology associated with Scratch programming.
- Explain the main categories and types of blocks in the Scratch toolbox.
- Apply knowledge of the categories and types of blocks to create scripts.
- Explain and use the geometric properties of the stage to move sprites.
- Explain and use the blocks in the motion and looks tabs to create a simple animation.
- Explain and use the “When Clicked”, “When key pressed”, “Broadcast” and “repeat” control blocks.
• Add comments to program code.
• Explain and use the blocks in the sensing tab to interact with the user.

**Time Permitting:**
• Explain and use the blocks in the pen tab to create a program that draws shape and text.
• Explain and use the blocks in the sound tab to add sound and music to a project.

**Interdisciplinary Standards**
• 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

**Activities (All STEAM Activities):**
• Lecture and class discussion.
• Guided and independent practice thru labs.
• Lab: Sprite and User interaction.
• Lab: Using the Sprite Editor.
• Lab: Moving, changing directions and changing costumes.
• Lab: "Gliding” sprite.
• Project: Alphabet Project.
• Lab: Broadcast and When I Receive.
• Lab: Ask User and Answer.

**Enrichment Activities:**
• Lab: Drawing Shapes.
• Lab: Drawing Shapes with Ask User

**Methods of Assessments/Evaluation:**
• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources/Including Online Resources**
• Teacher Webpage
• Resources listed above in **Resources** section
Unit 3.2: Variables, Data Types, and Operations

Approximate # Of Weeks: 1.5

Essential Questions:
- What is a variable?
- How are variables created and deleted?
- What is a data type?
- How is a value assigned to a variable?
- How is the value of a variable changed?
- What is a variable monitor?
- What are the arithmetic operations available in Scratch?
- What are the String operations available in Scratch?
- How are arithmetic expressions requiring multiple operations written and evaluated?

Upon completion of this unit students will be able to:
- Explain and use variables.
- Explain and use numeric and String data types.
- Explain and use the String operators (join, length, letter of).
- Explain and use the arithmetic operators.
- Explain and code statements containing nested arithmetic and String operations.
- Display the value of a variable on the stage using a variable monitor.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Lecture and class discussion.
- Guided and independent practice thru labs.
- Lab: User Defined Variables.
- Lab: Countdown.
- Project: Unicorn.
- Lab: Test Average.
- Lab: Area and Perimeter.
- Lab: Piggy Bank.
- Lab: Baseball.
- Lab: Initials.
- Lab: Mad Libs.
- Activity: Given code for a program, students will predict the output of the program.
- Activity: Given code for a program, students will find and correct the logic error.
**Enrichment Activities:**
- Lab: Digits.
- Lab: Time Converter.
- Lab: Ideal Weight.

**Methods of Assessments/Evaluation:**
- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources/Including Online Resources**
- Teacher Webpage
- Resources listed above in *Resources* section
Unit 3.3: Decision Statements (ifs) and Random Numbers

Approximate # Of Weeks: 1.5

Essential Questions:
- How can decision statements be used to control the execution of program statements?
- What are relational operators?
- What are Boolean operators?
- How are random numbers used in programming?
- How do an “if” statement and an “if else” statement compare?
- How do the Boolean operators, “and” and “or” compare?
- How are nested decision statements evaluated?
- What are some of the common errors programmers make when coding decision statements and how can these errors be avoided?

Upon completion of this unit students will be able to:
- Explain and use the relational operators.
- Explain and use the Boolean operators.
- Explain and use decision structures to control the flow of a program.
- Explain and use a random number generator.
- Explain and use nested decision statements.
- Develop code with correct and efficient use of conditional control structures.
- Explain and correct errors in programs containing decision statements.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Lecture and class discussion.
- Guided and independent practice thru labs.
- Lab: Random Numbers.
- Lab: Decision Statements – Guessing Game.
- Project: Dice.
- Lab: Grade.
- Lab: Old Enough.
- Labs: Improved Guessing Game / Grade / Age.
- Lab: Collision Detection – If Touching.
- Project: Collision.
- Lab: Given code for a program containing decision statements, students will predict the output of the program.
- Lab: Given code for a program containing decision statements, students will find and correct the logic error.
- Project: Lucky Ones
• Project: Rock Paper Scissors

Enrichment Activities:
• Lab: Even or Odd.
• Project: Maze.
• Project: Slot Machine.
• Project: Jumping Sprite.

Methods of Assessments/Evaluation:
• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources/Including Online Resources
• Teacher Webpage
• Resources listed above in Resources section
Unit 3.4: Procedures – Custom Blocks and Parameters

Approximate # Of Weeks: 0.5

Essential Questions:
- What is a procedure?
- How is Scratch’s “Make a Block” feature used to create procedures (custom blocks)?
- How are values passed to a custom block?
- How do procedures (custom blocks) help simplify program development?

Upon completion of this unit students will be able to:
- Explain and use the term procedure.
- Explain and use Scratch custom blocks.
- Create custom blocks that accept parameter values.
- Explain and call a custom block.
- Pass a parameter to a custom block.
- Describe and develop code using custom blocks.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); R1.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Lecture and class discussion.
- Guided and independent practice thru labs.
- Lab: “Move in Shape” Block – Modified “Draw Shape” as custom block
- Lab: “Draw Line from x1, y1 to x2, y2” Block.

Enrichment Activities:
- Project: Brick Wall project.

Methods of Assessments/Evaluation:
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources/Including Online Resources
- Teacher Webpage
- Resources listed above in Resources section
Unit 3.5: Iteration Statements (Loops)

Approximate # Of Weeks: 1

Essential Questions:
- What is an iteration statement (loop)?
- What is a repeat until loop?
- How does a repeat until loop differ from a repeat or forever loop?
- How do loops help simplify program development?
- What are common mistakes to avoid when working with loops?

Upon completion of this unit students will be able to:
- Explain and use loops.
- Compare and contrast repeat until, repeat, and forever loops.
- Explain and avoid accidental infinite loops.
- Use loops to simplify program code.
- Explain and avoid common errors that programmers make with loops.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Lecture and class discussion.
- Guided and independent practice thru labs.
- Lab: Repeat Until.
- Lab: Practice with Repeat Until.
- Lab: Draw a Squirrel.
- Lab: Countdown with Repeat Until.
- Project: Unique Random Numbers.
- Lab: Given code for a program containing loops, students will predict the output of the program.
- Lab: Given code for a program containing loops, students will find and correct the logic error.

Enrichment Activities:
- Project: Modified guessing game (play until correct).

Methods of Assessments/Evaluation:
- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources/Including Online Resources
• Teacher Webpage
• Resources listed above in Resources section
Unit 3.6: Lists and Common List Algorithms

Approximate # Of Weeks: 1

Essential Questions:
- What is a list?
- What is the empty list?
- What are the common list operations (traverse, insert, replace, delete, etc.)?
- How do lists help simplify program code?
- What are some of the common algorithms involving lists?
- What are some of the common errors programmers make when using lists and how can these errors be avoided?

Upon completion of this unit students will be able to:
- Explain, create and use a list.
- Explain and use an empty list.
- Explain and use the common operations on lists (insert, replace, delete, etc.).
- Use a loop to traverse a list.
- Explain and code some of the common list algorithms such as swapping values, finding the highest, lowest and average value of a set and searching for an element in a list.
- Use lists to simplify program code.
- Explain and avoid common errors that programmers make with lists.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Lecture and class discussion.
- Guided and independent practice thru labs.
- Lab: Make a list of names.
- Lab: Read a list of names.
- Lab: Practice with Index Variables – traverse a list in different orders, by different increments, or find values that meet certain criteria.
- Lab: Play with the names in the list.
- Lab: Delete names from list.
- Lab: Range of elements.
- Lab: Given code for a program containing a list, students will predict the output of the program.
- Lab: Given code for a program containing a list, students will find and correct the logic error.
- Class blog entries in response to teacher prompts.
**Enrichment Activities:**
- Lab: Processing a sentence.
- Project: Swap List Values.
- Project: Sort the List.

**Methods of Assessments/Evaluation:**
- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources/Including Online Resources**
- Teacher Webpage
- Resources listed above in Resources section
Unit 3.7: Programming in Scratch Final Project

Approximate # Of Weeks: 2.5

Essential Questions:
- What are some of the challenges encountered in developing computer applications?
- How can all the concepts and techniques learned in the previous programming units be applied to developing a complex game application?
- What are the issues to consider when designing applications involving user interactions?
- How can applications be tested to minimize or eliminate errors?
- How can applications be documented to help the user and other programmers understand and use the application?

Upon completion of this unit students will be able to:
- Explain and use the application development process.
- Apply all of the programming concepts and techniques learned in the previous programming units to design and develop a complex game application.
- Explain issues associated with user interface design.
- Explain and use effective testing practices.
- Develop documentation for a complex application.

Interdisciplinary Standards
- 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

Activities (All STEAM Activities):
- Working in pairs, students will submit a proposal for an original game application.
- Working in pairs, students will design, develop and test an original game application.
- Working in pairs, students will write specifications to document their original application.
- Students will share their projects with the class.

Enrichment Activities:
- None.

Methods of Assessments/Evaluation:
- Assessing students’ daily progress and participation.
- Grading completed proposal, project, and presentation as per teacher rubric.

23
Resources/Including Online Resources

- Teacher Webpage
- Resources listed above in Resources section
Unit 4: Programming in Python

Approximate # Of Weeks: 2.5

Essential Questions:
- What is Python?
- What is an interpreter?
- How does one find help in Python?
- How does one create a source file in Python?
- How does one print output in Python?
- What is a comment and why is it useful?
- What is a variable?
- How are variables created and deleted?
- What is a data type?
- How is a value assigned to a variable?
- How is the value of a variable changed?
- What are the arithmetic operations available in Python?
- What are the String operations available in Python?
- How can functions and methods be used to work with Strings?
- How is input received from the user?
- How can data types be converter to one another?
- How can decision statements be used to control the execution of program statements?
- What are relational operators?
- What are Boolean operators?
- How do "if", "if else", and "if elif" statements compare?
- How do the Boolean operators, "and" and "or" compare?
- How are nested decision statements evaluated?
- What is an iteration statement (loop)?
- What is a while loop?
- How do loops help simplify program development?

Time Permitting:
- What is a list?
- What are the common list operations (traverse, insert, replace, delete, etc.)?
- How do lists help simplify program code?
- What is a function?
- How are functions created in Python?
- How are values passed to a function?
- How can a function return a value?
- How do procedures (custom blocks) help simplify program development?
- How can files be used to store persistent data?

Upon completion of this unit students will be able to:
- Where to find the Python interpreter
- How to enter commands in the interpreter
- Explain and use the built-in help feature in the Python interpreter
• Explain and use an editor to make python source code files
• Explain and use console output
• Explain and use comments
• Explain and use variables
• Explain and use numeric and String data types
• Explain and use the arithmetic operators
• Explain and use the String operators
• Explain and use String functions (len, [], slices, in)
• Explain and use String methods (upper, find)
• Explain and use keyboard input
• Explain and use data type conversion
• Explain and use the relational operators
• Explain and use the Boolean operators
• Explain and use decision structures to control the flow of a program
• Explain and use nested decision statements
• Explain and use while loops
• Explain and use break statements to end loops early

**Time Permitting:**
• Explain, create and use a list.
• Explain and use the common operations on lists (insert, replace, delete, etc.).
• Use a for loop to traverse a list (or other collections).
• Use lists to simplify program code.
• Explain and use the term function
• Explain and use Python functions
• Create functions that accept parameter values
• Create functions that return values
• Explain and call functions
• Pass a parameter to a function
• Explain and use file input/output for storing persistent data

**Interdisciplinary Standards**
• 8.1.12.B.2; 8.1.12.D.5; 8.1.12.F.1; 8.2.12.C(1, 4); 8.2.12.D.3; 8.2.12.E(1, 3-4); 9.3.IT-PRG.1-8; 6.3.12.D.1; 1.1.12.D.2; NJSLSA (R1-2, R4, R7, W4, W6, W9); RI.11-12 (4, 7); W.11-12 (4, 6); Math.Practices.1-8

**Activities (All STEAM Activities):**
• Lecture and class discussion.
• Guided and independent practice thru labs.
• Lab: Hello World.
• Lab: Greeting with Name.
• Lab: Test Average.
• Lab: Area and Perimeter.
• Lab: Piggy Bank.
• Lab: Baseball.
• Lab: Initials.
• Lab: Mad Libs.
• Lab: Grade.
• Lab: Old Enough.
• Lab: Practice with while Loops.
• Lab: Countdown with while Loop.
• Project: Unique Random Numbers.

**Time Permitting:**
• Lab: Make a list of names.
• Lab: Read a list of names.
• Lab: Practice with Index Variables – traverse a list in different orders, by different increments, or find values that meet certain criteria.
• Lab: Play with the names in the list.
• Lab: Delete names from list.
• Lab: Range of elements.
• Lab: area Function
• Lab: max Function
• Lab: between Function
• Lab: moreThan3Letters Function
• Lab: isEven Function
• Lab: isNegative Function

**Enrichment Activities:**
• Lab: Processing a sentence.
• Project: Swap List Values.
• Project: Sort the List.

**Methods of Assessments/Evaluation:**
• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources/Including Online Resources**
• Teacher Webpage
• Resources listed above in **Resources** section
Unit 5: Web Design

Total Approximate # Of Weeks: 3.5

Unit 5.1: Basic Web Design (Basic Elements and CSS)

Approximate # Of Weeks: 2

Essential Questions:
- How is a storyboard used for planning a web page?
- What software can be used to create a web page?
- What is HTML?
- What are the basic elements of a web page?
- How is CSS used to add formatting to a web page?
- How can inline styles and/or classes be used to add formatting to individual elements or classes of elements?
- How can the same CSS styles be used on multiple pages?
- How can one create a web page that incorporates all of the design elements previously studied?

Upon completion of this unit students will be able to:
- Create a storyboard for a web page
- Create a web page based on a storyboard
- Create an html page with a title and a body
- Create an html page with paragraph tags, headings, line breaks, and horizontal lines
- Create an html page that includes emphasis and strong tags
- Create an internal style sheet with CSS
- Create a web page that uses an internal style sheet
- Add inline styles to a web page
- Add classes and class styles to a web page
- Create an html page that links to a separate CSS file
- Use html tags and CSS styling elements to separate style from structure
- Create a web page that incorporates all of the design elements previously studied

Interdisciplinary Standards
Activities (All STEAM Activities):

- Class Activity: Teacher leads class in piece-wise fashion with the introduction of each new element (see objectives above for elements covered). Students follow on a practice page. Then, after each major chunk of information, students are given the opportunity to update their personal website with the new techniques.

- Class Activity: Gallery Walk: At various stages in the unit, students should walk from computer to computer observing each other’s personal websites, and giving feedback. Students should receive feedback and consider revising their sites.

- Project: Mid-Unit Project: Students will work in pairs to create a storyboard for a multi-page website, on one of the following topics:
  - A career
  - A worldwide or community problem
This idea will be the basis for their website that they will add to later in their Unit Project. As they create the storyboard (a different sheet of paper for each page), they should make note of features they would like to add that they have not learned about yet. Remind students that their design should consider usability by a diverse set of users. Share with neighbors and obtain feedback to incorporate. Finally, create a first draft of an html file and an external CSS file for the home page of their website. The home page should make use of all of the html tags and CSS style elements learned to date.

- Unit test and quiz.

Enrichment Activities:

- Project: Students will spend extra time working on a more involved personal website.

Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources/Including Online Resources

- Teacher Webpage
- Resources listed above in Resources section
Unit 5.2: Intermediate Web Design (Hyperlinks and Layout Elements)

Approximate # Of Weeks: 1.5

Essential Questions:
- How are links used to connect different web pages?
- How does one place a table of data on a web page?
- How does one use a table to organize the content of a web page?
- How can CSS be used to format a table?
- How can lists be used to organize a web page?
- How can web pages be compartmentalized and formatted using Divs?
- What more advanced methods are there for organizing the layout of a web page with a more complex design?
- How can one create a website that incorporates all of the design elements previously studied?

Upon completion of this unit students will be able to:
- Create an html page that includes hyperlinks
- Use table, row, and column tagging in an html page
- Add CSS styling to an html table
- Use ordered and unordered list tagging in an html page
- Add CSS styling to an html list
- Use grid elements in CSS div placement
- Add layout styles to a web page
- Create a complete website that incorporates all of the design elements previously studied

Interdisciplinary Standards

Activities (All STEAM Activities):
- Class Activity: Teacher leads class in piece-wise fashion with the introduction of each new element (see objectives above for elements covered). Students follow on a practice page. Then, after each major chunk of information, students are given the opportunity to update their personal website with the new techniques.
- Project: Unit Project: Students will work in pairs to finish the website they began in their Mid-Unit project, following a rubric. They will first edit the storyboard for their multi-page website that they created in their Mid-Unit Project. At this point, they should try to implement the notes they took during the Mid-Unit Project on features they would like
to add that they had not learned about yet. Students should consider usability by a diverse set of users. They should do a final draft of their home page, during this project, and the rest of their at least 3-page site, using all of the elements they have learned in these units.

- Unit test and quiz.

**Enrichment Activities:**
- Lab: Students will use the w3schools CSS Navigation Bar tutorial (http://www.w3schools.com/css/css_navbar.asp) to learn how to create a menu. Students will then include a navigation bar in their Unit Project.

**Methods of Assessments/Evaluation:**
- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources/Including Online Resources**
- Teacher Webpage
- Resources listed above in *Resources* section