

Tech Gr 3 Unit 2 - Problem Solving through Programming

Content Area: **Technology**
Course(s): **STEM-Technology**
Time Period: **Two Weeks**
Length: **2nd Marking Period**
Status: **Published**

Stage 1 - Learning Outcomes

Unit Overview

Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming: All students will develop an understanding of the nature and impact of technology, engineering, technological design, computational thinking and the designed world as they relate to the individual, global society, and the environment.

Strand E. Computational Thinking: Programming: *Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.*

Enduring Understandings

- Computer programming is a tool used to help us solve problems, create, and design.

Essential Questions

- How does computer programming help us solve problems, create, and design?

Interdisciplinary Connections

1. Common Core Literacy

- | | |
|---------------------------------|--|
| LA.3.CCSS.ELA-Literacy.CCRA.R.7 | Integrate and evaluate content presented in diverse media and formats, including visually and quantitatively, as well as in words. |
| LA.3.CCSS.ELA-Literacy.CCRA.W.6 | Use technology, including the Internet, to produce and publish writing and to interact and |

| | |
|----------------------------------|---|
| | collaborate with others. |
| LA.3.CCSS.ELA-Literacy.CCRA.W.10 | Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences. |
| LA.3.CCSS.ELA-Literacy.CCRA.SL1 | Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively. |
| CCSS.ELA-Literacy.W.3.10 | Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences. |
| CCSS.ELA-Literacy.RF.3.4.a | Read grade-level text with purpose and understanding. |
| CCSS.ELA-Literacy.SL.3.1 | Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly. |

2. Common Core Math

- CCSS.MATH.PRACTICE.MP1 Make sense of problems and persevere in solving them.
- CCSS.MATH.PRACTICE.MP2 Reason abstractly and quantitatively.
- CCSS.MATH.PRACTICE.MP3 Construct viable arguments and critique the reasoning of others.
- CCSS.MATH.PRACTICE.MP5 Use appropriate tools strategically.
- CCSS.MATH.PRACTICE.MP6 Attend to precision.
- CCSS.MATH.PRACTICE.MP7 Look for and make use of structure.

3. Career Ready Practices

- CRP11. Use technology to enhance productivity.
- CRP6. Demonstrate creativity and innovation.
- CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.

Stage 2 - Assessment

Formative Assessment

Teacher observation of students' progress

Teacher observation of completed projects

Teacher observation of student cooperation and class discussion

Matrix of skill development

End of Unit/Benchmark Assessments

Students will use their Google Apps account to login to CODE.ORG and will provide evidence of completion of at least 12 lessons in Course 2.

<https://studio.code.org/s/course2>

Teacher must setup a "class code" in advance of students logging on with their Google accounts. Students will enter this class code after logging in the first time (using their Google Apps account) but before beginning any work. After students have completed their work, teacher can view/print evidence of what Code.org lessons students have completed.

Stage 3 - Learning Plan

Suggested Activities

- Blockly
- BotLogic <http://botlogic.us/play>
- Code an Arduino (microcontroller) using some of the projects in this book: Sylvia's Super-Awesome Project Book: Super-Simple Arduino (Volume 2)
- Code using a Raspberry Pi device. low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.
<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- Code.org <http://code.org>
- Coding Symboloo created by Shannon Miller <http://www.symboloo.com/mix/coding-coding-coding>
- Daisy the Dinosaur app <https://itunes.apple.com/us/app/daisy-the-dinosaur/id490514278?mt=8>
- Flocabulary Top 10 Reasons to Code Video <https://www.youtube.com/watch?v=78-RllzwDHo>
- Have a student "program" another student to move through the classroom - Kevin Jarrett lesson here <http://blogs.ncs-nj.org/k4stemlab/?p=1792>
- Human Computer Activity: <http://csunplugged.org/activities/>
- Lightbot <http://lightbot.com/hocflash.html>
- Madewithcode <https://www.madewithcode.com/>

- Participate in the Hour of Code on Code.org or Tynker
- Play code monkey <https://www.playcodemonkey.com/>
- Purple Mash Coding Section (subscription) <https://www.2simple.com/purple-mash>
- Scratch by MIT <https://scratch.mit.edu/>
- Teacher discussion of vocabulary (algorithm, program, debug, loops, events, procedures, memory, storage, processing, software, coding, procedure, data)
- Teacher discussion of why we should learn coding - we use technology everyday and it is good to view "behind the scenes," because more careers will rely on computer programming, and because coding makes you think differently - try it out, make mistakes, and learn that way!
- Tynker <https://www.tynker.com/>
- What most schools don't teach - Why Code? Code.org video <https://www.youtube.com/watch?v=nKIu9yen5nc>

LESSON STRATEGIES

- Interactive mini lecture/notes – Students take notes, answer questions, turn and talk, participate in class discussions, look up information in text/online.
 - Do Nows – At start of class, students answer questions, reflect on learning, work on typing.com
 - Video Clips- Shown to aid learning by providing a visual, engage the class
 - Demonstrations- To enhance student learning, ignite curiosity, spur discussion, provide a visual, engage the class
 - Research questions- Students are given a broad question to research. Students answer the question and provide evidence for their responses. (Independent or collaborative)
 - Journaling- Responses to various ideas, thoughts, class activities, and content.
 - Reflecting on Learning- Students self reflect on their learning and “rate” themselves on a learning scale.
- Blockly
- BotLogic <http://botlogic.us/play>
- Code an Arduino (microcontroller) using some of the projects in this book: Sylvia's Super-Awesome Project Book: Super-Simple Arduino (Volume 2)
 - Code using a Raspberry Pi device. low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- Code.org
- Coding Symboloo created by Shannon Miller <http://www.symboloo.com/mix/coding-coding-coding>
- Daisy the Dinosaur app <https://itunes.apple.com/us/app/daisy-the-dinosaur/id490514278?mt=8>
- Demonstrations- To enhance student learning, ignite curiosity, spur discussion, provide a visual, engage the class
- Do Nows – At start of class, students answer questions, reflect on learning, work on typing.com

- Flocabulary Top 10 Reasons to Code Video <https://www.youtube.com/watch?v=78-RllzwDHo>
- Have a student "program" another student to move through the classroom - Kevin Jarrett lesson here <http://blogs.ncs-nj.org/k4stemlab/?p=1792>
- Human Computer Activity: <http://csunplugged.org/activities/>
- Journaling- Responses to various ideas, thoughts, class activities, and content.
- LESSON STRATEGIES: Interactive mini lecture/notes – Students take notes, answer questions, turn and talk, participate in class discussions, look up information in text/online.
- Lightbot <http://lightbot.com/hocflash.html>
- Madewithcode <https://www.madewithcode.com/>
- Participate in the Hour of Code on Code.org or Tynker
- Play code monkey <https://www.playcodemonkey.com/>
- Purple Mash Coding Section (subscription)
- Reflecting on Learning- Students self reflect on their learning and “rate” themselves on a learning scale.
- Research questions- Students are given a broad question to research. Students answer the question and provide evidence for their responses. (Independent or collaborative)
- Scratch by MIT
- Teacher discussion of vocabulary (algorithm, program, debug, loops, events, procedures, memory, storage, processing, software, coding, procedure, data)
- Teacher discussion of why we should learn coding - we use technology everyday and it is good to view "behind the scenes," because more careers will rely on computer programming, and because coding makes you think differently - try it out, make mistakes, and learn that way!
- Tynker <https://www.tynker.com/>
- Video Clips- Shown to aid learning by providing a visual, engage the class
- What most schools don't teach - Why Code? Code.org video <https://www.youtube.com/watch?v=nKlu9yen5nc>

Supportive Strategies

1. Special Education

- Employ assistive technology as needed (For example, use of Dyslexie font, high contrast or screen magnification on Chromebook, or spoken text features)
- Graphic Organizers
- Modifications on IEP
- Provide written and oral directions, utilizing visuals and exemplars. (For example, teacher will model on ActivBoard)
- Reduction in workload
- Repetition and Reinforcement of classroom material
- Strategic Grouping for all group work

2. ESL

- Employ assistive technology as needed (For example, online translation or Language text settings on Chromebook)
- For collaborative assignments, appropriate roles will be assigned. (For example, time-keeper, activity starter)
- Make content culturally relevant
- Partner English Learners with Strong English Speakers
- Provide written and oral directions for all lessons, utilizing visuals and exemplars
- Repeat classroom procedure and routines as much as possible to reinforce language learning
- Visual Aids

3. Student at risk of failure

- Employ assistive technology as needed (For example, use of Dyslexie font, high contrast or screen magnification on Chromebook, or spoken text features)
- Flexible acceptance of missing/lost/incomplete assignment
- Strategic Grouping for all group work

4. Gifted and Talented

- Higher level learners will be provided with more intellectually demanding learning activities. (For example, allowing students to complete as many levels as they wish in Course 2 in Code.org)
- Higher Order Questioning
- Utilize different reading levels appropriate for students

Core Instructional Materials/Resources

- Bee-Bot <https://www.bee-bot.us/>
- Chromebooks, iPads or other internet-connected devices
- Code.Org Inspirational Videos - <https://code.org/educate/inspire>
- CS Unplugged website - <http://csunplugged.org/>
- See list of coding websites listed in “Suggested Activities” section.