Powered by [redacted]

A STEAM Resource for Educators in a Digital World • 2024

# ROOT & STEM

## Grade 10 Computer Studies

**CODING FOR CULTURE**

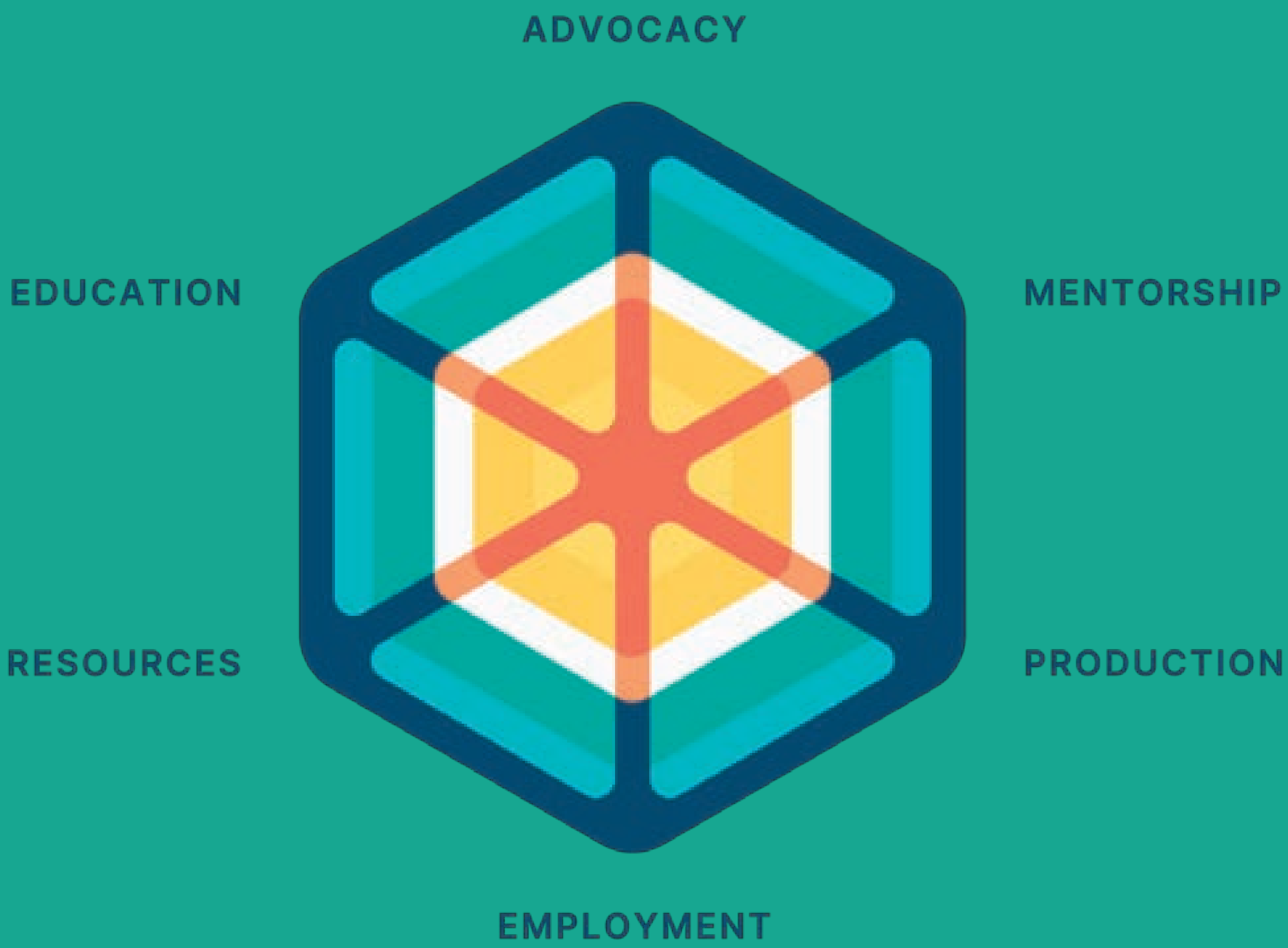
Finding rhythm in the classroom

**PRINTED WITH PURPOSE**

The vision behind Makers Making Change

**HOW TO HOST A HACKATHON**

Simple ways to get started




# AMPERE LIFE CYCLE

Ampere follows a life cycle model to support the core phases of a person's learning journey in STEAM education. We strive to provide educators and students with opportunities and resources each step of the way.

To learn more about what we do, visit our website at

**amp.ca**

Powered by

# ROOT & STEM

## ABOUT AMPERE

Ampere, a not-for-profit organization, incorporates STEAM into unique learning applications that promote storytelling, health, wellness, and growth in rural and remote communities. At its core, Ampere embraces diversity and creates opportunities in order to empower all people.

## DIGITAL TAXONOMY

Computer Science Education is more than just coding. A comprehensive approach to it includes learning skills and competencies from each of the following areas. Look for the icons at the bottom of each article or lesson plan for suggested curriculum connections.

### CODING AND PROGRAMMING

- Algorithms
- Data Structure
- Modelling and Abstraction
- Modularity
- Debugging

### COMPUTING AND NETWORKS

- Hardware and Software
- File Management
- Troubleshooting
- Digital Connectivity
- Cybersecurity

### DATA

- Data and Its Uses
- Organizing Data
- Assessing Information
- Ownership and Governance
- AI and Machine Learning

### TECHNOLOGY AND SOCIETY

- Safe Communication
- Ethics, Safety, and the Law
- Tech and the Environment
- History and Technology
- Tech and Well-being

### DESIGN

- User Design
- Visual Design
- Universal Design

Powered by Root & STEM *is printed on certified Forest Stewardship Council paper, to reduce our impact on the natural world.*

*This publication is funded in part by the Government of Ontario and the Government of Canada.*

Ontario Canada

*Powered by Root & STEM*
(ISSN 2563-6979) is published by **Ampere**

87 Adelaide St N, Lindsay, Ontario K9V 4L4
**@root_stem@amp.ca**
**rootandstem.ca**

Printed in Oshawa, Ontario, by Maracle, Inc

**Publisher**
Ryan Oliver

**Editorial Lead**
Chelsea Kowalski

**Editorial Assistant**
Nadia Scott

**Copy Editor**
Nicholas Davies

**Design & Layout**
Ayaya

**Art Direction**
Stephanie Amell, Emily Canfield, Kelly Eng, Mariana Fernandez, Alyssa Rowe

**Cover Illustration**
Chantal Jung

**Logo Design**
X-ing Design

**Contributors**
Ayesha Akhlaq, Sharon Aschiaek, Tanner Big Canoe, Austin Cozicar, Brian McLachlan

Follow Us **@amperecanada**

# Contents

## CHANTAL JUNG

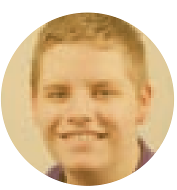
*Cover Illustration*

Chantal Jung (she/they) is a Nunatsiavut Inuk and self-taught multimedia artist and writer who focuses on collage art, zines, video, and film. She is originally from Happy Valley-Goose Bay, Newfoundland (Nunatsiavut), and currently resides as a guest on Ramaytush Ohlone land (San Francisco). Chantal has produced animated work for the musician Black Belt Eagle Scout and for the Bartow Project, and her writing and artwork have been featured in the *Inuit Art Quarterly* and on the Inuit Art Foundation's website.

## JON CORBETT

*Guest Editorial • Page 6*

Jon Corbett is Assistant Professor with Lived Indigenous Experience in the School of Interactive Art & Technology at Simon Fraser University. His research focuses on Indigenous forms of expression through "Indigitalization," which he describes as a computational creative practice that braids together Indigenous and decolonial computing practices facilitated through traditional and computer-based expressive media art forms.



## AUSTIN COZICAR

*Printed with Purpose • Page 7*

Austin Cozicar is a Communications Specialist for the Neil Squire Society based in the Lower Mainland region of British Columbia. With over 10 years of experience in journalism and communications, Austin is proud to use his storytelling abilities in the non-profit space.

## BRIAN MCLACHLAN

*Digital Footprint • Page 10*
*A.lways I.naccurate • Page 12*

Brian McLachlan is the cartoonist of *Complete the Quest: The Poisonous Library*, a combination role-player game and graphic novel, and *Draw Out the Story: Ten Secrets to Creating Your Own Comics*. He makes comic tutorials on Deep Thought Balloon, his YouTube channel.

## SHARON ASCHAIEK

*Coding for Culture • Page 13*

Sharon Aschaiek, the principal of Higher Ed Communications, writes about the education space, producing articles about research breakthroughs, successful alumni, and innovative practices.

## TANNER BIG CANOE

*Activity Guide: Unplugged • Page 14*

Tanner Big Canoe is a Computer Science graduate from Queen's University. He is the Director of Productions at Ampere and is always looking for ways for software to serve as a solution to problems with users at the center of innovation.

# Improvement through Indigenous Perspectives

As an artist and professor, I weave together nehiyaw (Plains Cree) cultural narratives with technology. My work focuses on integrating Indigenous heritage and cultural practices with modern technologies, thereby personalizing technology for Indigenous individuals and communities. To further these goals, I contribute my expertise in Indigenous-focused computing by serving as a fact-checker for the Government of Ontario's Grade 10 Computer Studies curriculum. In this role, I ensure the curriculum is inclusive, celebrates diverse student identities, and promotes equity, thus shaping a welcoming and culturally diverse technological landscape for all users.

In computer studies, fostering inclusivity and diversity is essential for creating meaningful learning experiences that resonate with all students. The Ontario government's consultation with Indigenous and subject-matter experts like myself is crucial, as it ensures the educational landscape in computer studies is equitable and inclusive, and that it makes the world of information and communication technology more welcoming and diverse, so that every student sees their culture and identity reflected and valued in their learning journey.

By embracing Indigenous ways of knowing, educators can create inclusive learning experiences that empower students to explore the intersection of technology, culture, and society. As we strive towards educational equity and cultural diversity, incorporating Indigenous perspectives enriches the educational landscape and prepares students from all cultural contexts to navigate our increasingly interconnected and multicultural world. Indigenous perspectives in the development of the Grade 10 Computer Studies curriculum offer many benefits, including cultural relevance, community engagement, and culturally responsive pedagogy.

## Cultural Relevance and Contextualization

Acknowledging how perspectives of computers and technology differ between Western and Indigenous peoples is crucial. For many Indigenous groups, the Land is central to our culture and spirituality, and it shapes what we worry about and what we elevate as important. Unlike mainstream Western perspectives, which often prioritize technological advancement and economic growth, Indigenous communities prioritize holistic approaches centred on sustainability, biodiversity, and intergenerational stewardship. Therefore, by highlighting how Indigenous communities utilize technology in traditional practices such as storytelling and cultural preservation, students gain a deeper appreciation for the diverse applications of computer studies.

## Community Engagement and Collaboration

Collaborating with Indigenous communities in curriculum development demonstrates integrity and mutual respect in cultural exchange. Educators gain invaluable insights into culturally relevant teaching methodologies, learning resources, and pedagogical approaches by consulting with Indigenous Elders, knowledge keepers, and community leaders. Furthermore, involving students from diverse cultural backgrounds in co-designing projects and assignments ensures that curriculum content reflects their lived experiences and perspectives.

## Culturally Responsive Pedagogy

Adopting culturally responsive pedagogy enables educators to create learning environments that honour cultural ways of knowing and learning styles. Incorporating storytelling, experiential learning, and land-based activities into computer-studies instruction resonates with students and enhances their engagement and academic success. By valuing diverse languages, symbols, and oral traditions, educators demonstrate respect for all cultures, which can give students a greater sense of belonging and empowerment. By prioritizing Indigenous worldviews, educators can place computer studies within cultural contexts that promote a deeper appreciation for the diverse applications of technology. Additionally, integrating Indigenous perspectives encourages interdisciplinary connections and promotes a more holistic understanding of technology's societal impacts.

Ultimately, embracing Indigenous perspectives prepares students from all cultural backgrounds to navigate an interconnected, multicultural world that fosters educational equity and cultural diversity.

– JON CORBETT

COMPUTING AND NETWORKS

TECHNOLOGY AND SOCIETY

COURTESY OF MAKERS MAKING CHANGE

A Makers Making Change team member helps two young Makers build an assistive device at a build event

# Printed with Purpose

## Makers Making Change workshops teach students to create assistive technologies for people with disabilities

**BY AUSTIN COZICAR**

Assistive technologies help people with disabilities live independently and increase their quality of life. Think of a wheelchair for someone with a spinal cord injury to move around, a screen reader that allows a person with visual impairment to use a computer, or even a large button switch to make playing with a toy easier for a child with a motor disability.

A problem, however, is that a lot of commercially available assistive technologies are quite expensive. The cost of that seemingly simple button switch for a toy, for example, can easily exceed $100 from a commercial supplier. Cost makes things difficult, especially for those who might need multiple assistive devices.

Makers Making Change (MMC), a program of the Canadian non-profit Neil Squire Society, has been making assistive technology more accessible for four decades. By hosting a library of open-source designs—anyone can make, customize, and use the designs as they see fit—that can often be built with a 3D printer and some tools, MMC engages volunteers to build devices, thereby significantly reducing costs. Devices can be as simple as a 3D-printed ball that fits over a pen to make writing easier for people with arthritis, to more complex solutions like mouth-operated joysticks that allow people who do not have the use of their hands to operate phones or game systems.

And through its STEM with Purpose initiative, MMC is engaging students across Canada to join the cause. MMC has developed a series of STEM workshops for classrooms that cover the various aspects of the creation process, from using computer aided design (CAD) software to building devices. Students get to employ their skills to tackle real-world challenges, all the while increasing their awareness of a wide range of disabilities and the need for assistive technology.

"It's an interesting concept to students, because usually they're not thinking about, or they don't know anything about, accessibility or assistive tech," says Courtney Cameron, MMC's East Region Coordinator.

The workshops can be tailored to a range of ages, with events that are appropriate for elementary school students up to Grade 12 and beyond. Teachers can choose or combine the workshops that are most applicable to their classes. Sessions can be run by teachers independently or with assistance from MMC staff. Many of the workshops involve learning hands-on skills like soldering, which is always a hit with students.

"On a really basic level, teaching a kid to solder is really cool," says Cameron. "It's really cool how [hands-on learning] can bring out different sides of students, or reach those students who aren't always into academics."

In the intro to 3D printing workshop, students learn all about the basics of the tech so they can print devices on their own at home or at school. The workshop also shows students how to use CAD software to design assistive technology that can be 3D-printed.

"We usually try to match up the software to whatever the school has access to," says Cameron, noting that they often use software like TinkerCAD and Autodesk Fusion 360. "If a school is working with a certain software or certain CAD, we can make our workshop work with that."

One of the most popular workshops is the build event. Using 3D-printed parts and other supplies, students work together to build assistive devices, which in turn are donated to local hospitals and people with disabilities.

The circuits and electronics workshops and microcontroller workshops introduce students to the workings of electronic devices. In some of the more advanced workshops, students upload and test coding to make the devices operate.

Finally, design challenge workshops see students working together to use their problem-solving skills to design solutions for real-world issues faced by people with disabilities, ending with a working prototype.

"Teaching people how to build stuff really empowers them to take it into their own hands," says Cameron. "It's this idea of passing on these skills, so people can take it back and really affect their community." &

COMPUTING AND NETWORKS

TECHNOLOGY AND SOCIETY

DESIGN

**For more information on Makers Making Change's STEM with Purpose workshops, including a STEM Workshop Planner, visit *makersmakingchange.com/s/stem* or email *info@makersmakingchange.com.***

# How to Host a Hackathon

BY AMPERE

One of the most effective ways to come up with solutions for issues is to consider them in a team, where a variety of perspectives, opinions, and expertise can be shared. In the field of technology and innovation, problems are explored in an engaging and collaborative way: the hackathon.

Hackathons are designed to bring developers, designers, innovators, and other experts together to solve specific problems. They typically last for 24 to 48 hours and are often planned as fun events, but they can be very effective for solving problems in the areas they address.

Hackathons have a variety of applications, and have been used to advance medical tech, mobile apps, APIs, and more. The events offer several advantages over traditional processes that approach innovation and problem-solving. Hackathons are inclusive and agile; they promote collaboration; and they are a great way to engage learners with technology in a creative, play-based learning environment.

Interested in creating a hackathon in your community? Here's a guide to preparing for and hosting these special events.

## Find a Suitable Location

Finding a suitable location to host a hackathon can take time. It may be wise to reach out early in the planning process to contact partner organizations that may be willing to donate or rent their space. Some questions to ask when considering whether a space is appropriate or not include:

1. Are there enough electrical outlets for all the necessary computers and devices?
2. Is the wi-fi fast enough, and can it support all participants using it at one time?
3. Is the space large enough for teams or individuals to work independently?

## Registration

Eventbrite, or similar event management websites, are great tools to use for participant registration. Ideally, you should publicize your event about eight weeks before it takes place. Besides names, contact details, and skill-level questions, also ask applicants for information regarding important health or dietary considerations, whether they require transportation to and/or from the event, accessibility needs, permission to take photographs, and whether they would like to hear about future events.

## Skill Level

The skill level of participants will help you determine the theme of the hackathon, its activities, and will guide how participants are assigned to teams. Creating teams with participants who have different levels of experience in hackathons—as well as different strengths in areas like leadership, art, and coding—can make a significant difference in the learning experience. Include questions in the registration process that ask participants to select their skill level on a five-point scale. Here is an example:

What is your experience with coding and technology?

1. I can figure out how to fill out an Eventbrite registration form, but that's about it
2. I'm definitely comfortable using a computer but haven't done any coding
3. I've tried a bit of coding by following lessons with something like Hour of Code or Scratch
4. I have a pretty good understanding of code and have made a few cool things with it
5. Code is my life

## Parameters

Required materials vary depending on the ages of the participants and the tasks and project expectations of your hackathon. All hackathons require computers or laptops (don't forget the chargers!), and a reliable internet connection. If you plan to give participants a challenge involving a specific device or technology—like building an app compatible with Alexa (the Amazon device) for example—make sure every team has access to and a basic understanding of the device or technology in question. If education is required for the coding languages, tools, or devices being used, offer a training workshop. Make sure facilitators are prepared to check in with the teams and offer support as needed.

**Questions to Support Team-Building:**

- What are your strengths and passions?
- How do you want to contribute to this project?
- Do you work best alone or with a partner? (Some individuals may want to do pair-programming, for which two participants work on one computer)
- Do you want to be involved in all aspects of the project (programming, presenting, creative brainstorming, copywriting) or do you have specific preferences?

Hackathons for younger participants might require different guidelines than those for a hackathon with teenagers. This might include limiting expectations to programming without incorporating an additional device (like Alexa). Younger participants may also benefit from more time, for example a week-long hackathon including more workshops and space to learn, while older participants may be comfortable with just two or three days for the event. When it comes to team building for young participants, prepare questions with simpler language, such as:

- What do you find fun?
- What do you want to learn?
- What is a programming task you already know how to do?
- Do you want to learn several new skills or would you rather focus on developing one?

### Choose a Task

There are at least two ways to think about challenging participants when deciding on the task that your hackathon will focus on. You might provide all participants with the same technology and challenge them to develop something unique or, inversely, participants might all be given the same problem to solve, each using different hardware and software. For example, for the first Ampere (formerly Pinnguaq) hackathon, all participants were given an Alexa Echo Dot device and the MIT App Inventor tool and asked to create a unique app.

Emphasize to participants that they are building a prototype, and that it does not have to be complete nor even work properly at the end of the hackathon. Nevertheless, to give participants the best chance at feeling satisfied with their work, try to provide enough time for them to complete the project.

### Create a Schedule

After you have determined the task or theme, the next step is to break it down into an activity program. The first steps in the program should teach participants about the tech they will be using. Depending on their experience and skill level, providing some basic background information may be helpful. For example, during the Pinnguaq hackathon, participants were taught some basic AI knowledge, how to set up and use their Alexa devices, and how to make a simple app using MIT App Inventor. If you choose a theme that revolves around creating a video game, for example, it might be helpful to teach some game design theory, coding with specific tools, and creating art and designs for games.

When working with young learners, especially if they are in teams, scheduling some time for socialization, perhaps including a few icebreaker games, at the start of the hackathon is a great way to help participants get to know one another. Once participants are more comfortable, give them time to brainstorm ideas. Ideally, this is done after they have been informed about the project, task, or goal, but before they have learned about any restrictions, such as any technological limitations. This gives them the opportunity to plan a really unique project that isn't limited by concerns related to technology.

After brainstorming, allow enough time for participants to work on their project. For this part of the hackathon, it is important to schedule enough staff to ensure that all participants have access to adequate support. On the start of the final day, brief participants on the goals for their presentation, and have staff encourage them to spend some of their work time preparing to present their work. At the end of the final day, make sure enough time is left to say thanks to all learners, staff who participated, and community partners who donated support, as well as to award the prize to the winners.


COURTESY OF AMPERE

### Promotion and Connecting with the Community

**Social Media**

Leveraging popular social media platforms will help you reach a wide audience. Post regular updates, engaging content, teasers, and behind-the-scenes glimpses leading up to the event. Use relevant hashtags and encourage participants to share their excitement online. Create and share valuable content related to the hackathon's theme or industry. This might include blog posts, infographics, videos, or interviews with industry experts.

**Awareness**

Participating in relevant online forums, social media groups, and tech communities is a great way to attract participants. Engage with members of these groups and share details about the hackathon, making sure to follow any community guidelines for promotion. Email marketing campaigns are another great way to promote your event. Send targeted emails to potential participants, previous attendees, partners, and sponsors. Personalize the messages, highlight the benefits of participating in the hackathon, and include a clear call for registration.

Above all, make sure to keep fun at the core of everything that happens for both participants and the organizers to ensure skills are learned and memories are created. &

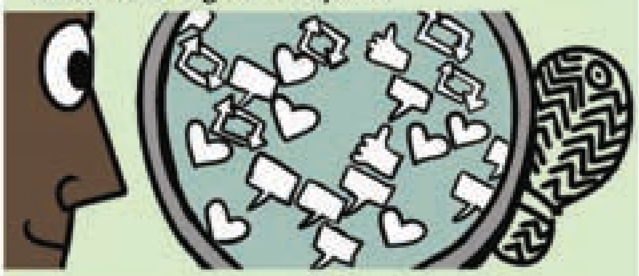
CODING AND PROGRAMMING
COMPUTING AND NETWORKS
TECHNOLOGY AND SOCIETY

# DIGITAL FOOTPRINT

Every time you like a social media post, leave a review, move from one website to another, log in to a website, or do anything at all online, you are being tracked. Wherever you go, you leave a digital footprint, which is recorded by your operating system, your browser, and the websites you visit. This information can be used to personalize your online experience in useful ways, like providing ads or other suggestions you might be interested in given the online path you have taken. It can also be used to make other sorts of assessments about you, however, and you might not realize what kinds of personal information you are revealing, nor to whom.

Think you know what kind of internet trail you leave? Test your knowledge about the effects of a digital footprint in this True or False test.

**You don't need to be a hacker to look at someone's digital footprint.**



True. Parts of your social media profiles, like who your friends are, which posts you've liked, your comments on public posts, etc., can often be seen with no effort. To manage more closely who can see detailed information about you, make your profiles private.

**A smart car can't share information.**

False. Smart cars collect data about driving habits—how far people drive, their speed on corners, etc. The auto companies that make those cars and operate their data systems can share that information with insurance companies.

**Collecting online data without permission is punishable by law.**

True (in theory). Around the world, countries and federations have laws in place that allow them to fine corporations for illicit data collection. Huge companies have been fined many times—to the tune of more than $2 billion in Meta's case—but, because the profit from selling data often outweighs the cost of the fines, they frequently continue the illegal activity.

**Gathering your likes and ratings allows algorithms to understand what kind of content you engage with so they can focus on directing you to content and products you already prefer and enjoy.**


True. But that also means they are less likely to direct you to movies, songs, television shows, etc. that might expand your horizons by introduce you to new or different knowledge and ideas.

**Having your phone's location sharing feature turned on can help you recover it if you lose it or it is stolen.**

True. By using apps like Google's Find My Device and Apple's Find My, lost phones can be located easily. But, having the location sharing feature on all the time if you have your phone with you means apps can find out where you are and where you go. You can manage this by only sharing your phone's location data with apps that really need the information and that you trust.

**If you use private or incognito windows, websites can't track you.**

False. Some browsers offer separate tabs or windows that come with the promise that none of your information or data is saved when you use them for browsing. These services offer more privacy than regular browsing windows because your history isn't recorded, your login information isn't saved, and cookies and site data are automatically deleted. Your internet service provider and operating system can still track your browsing, however, and the websites you visit can still see and identify you by your IP address.

**Companies that gather data to learn more about user patterns rely on cookies.**


True. Cookies (not the baked kind!) are crumbs of information about you that a website remembers, like your login data, where you are located, the websites you go to when you leave theirs, the links you click on while on their site, etc. Some jurisdictions, like California, for example, have laws that require websites to obtain consent from users to operate cookies, but it is still common for websites to use cookies regardless of users' preferences.

**Strong passwords (for your email account, for example) always protect your accounts from being hacked.**

False. Most hacking comes from phishing. This is a scam in which you receive a message from a hacker pretending to be someone you know or a representative of a familiar organization that asks you to reveal sensitive information, like your password, without realizing it's a set-up. Sometimes, fake versions of real websites are even created to trick people into entering passwords into the lookalike, password-gobbling sites.

by Brian McLachlan

DATA TECHNOLOGY AND SOCIETY


A. LWAYS I. NACCUR-ATE
by Brian McLachlan
Whoa. Someone's hooked this hologram room up to A.I. from the year 2024.
Program, make a throne room!
Huh? How does it get it so wrong?
Low-paid workers fed it texts, chats, and images but the algorithm can't understand if material contains an error, a joke, a bias, or if it's missing context, etc.
And what even is a chair? If I sit on it, is a tree stump a chair? If put it over my head in the rain, does it become an umbrella instead? What about a reflection of a chair?
I guess I know it when I see it, but making me think about it hurts my head.
Program, Make a throne room with a painting of Denise as queen.
That's not cool, generating pictures of me without my permission.
Hey, we've drawn each other before. For fun.
Yeah, but using a photorealistic generator to make me fit your imagination is creepy.
It does hit different. Sorry.
ENERGY ALERT!
Using A.I. has cost us 40% more power! The cooling cost is huge.
Cancel program.
This is the thing they were worried was going to take over the world?
Yeah. I guess they never imagined it would be brain slimes.

# Coding for Culture

STEAM-based music contest illuminates Indigenous narratives

BY SHARON ASCHAIEK


Your Voice Is Power winners Nodin Outten-Joseph and Lucia Almeida with the Honourable Marilou McPhedran during their visit to be recognized at the Senate of Canada

*"They tried to extinguish our flames but we never left / They said that we are the holders of blame but we never left"*

These lyrics to a thoughtful rap song by recent Toronto high school graduate Nodin Outten-Joseph, who is of Indigenous and Black heritage, reflect his views on the perseverance of Indigenous identity and the importance of honouring our cultural ancestors.

The song was a grand winner of the 2022–23 edition of Your Voice Is Power, a contest for middle- and high school students in Canada that combines music-making, computer science, and Indigenous affairs. Sponsored by Amazon, the initiative engages students in developing their science, technology, engineering, and math (STEM) skills by using coding software to remix music by Indigenous artists, while also teaching them about the history and resilience of Indigenous Peoples in Canada.

"Music is a unique way to engage students in dipping their toes into coding," says contest founder and organizer Michael Furdyk of TakingITGlobal, a charity that is administering the contest's 2023–24 edition. "It's also a great way to teach students about the experiences of Indigenous Peoples in Canada and the accomplishments of Indigenous artists."

Furdyk says Your Voice Is Power emerged in response to the lack of diversity in STEM fields, in which visible minorities continue to be under-represented. Meeting this challenge was a perfect fit for his charity, which uses technology, creativity, and community to design and deliver programs that engage youth.

His solution was to adapt a US version of the contest for Canadian students and centre the experiences and music of Indigenous Peoples. He launched the contest in 2022 with the support of Amazon Music and Amazon Future Engineer, a program focused on increasing computer science education for students from underserved communities.

Participants in Your Voice Is Power learn to use EarSketch, a free online code editor for recording, editing, and playing digital audio files. They use the tool, which is available in English, French, Inuktitut, and Ojibwe, to create their own songs by remixing music by Indigenous artists. The 2024 edition featured the music of Aysanabee, Dakota Bear, Jayli Wolf, Samian, and Twin Flames, five nationally recognized Indigenous artists who incorporate themes of social justice and Indigenous endurance into their lyrics.

The contest takes place within an Indigenous learning program, developed by Anishinaabe educators, that aligns with public education curriculum requirements and includes sources for teachers to support participating students. The program's eight modules focus on core elements of anti-Indigenous racism in Canada, including residential schools and the Sixties Scoop, as well as the calls to action of the Truth and Reconciliation Commission. Participants also learn about decolonization, equity and activism, and how to be an ally of Indigenous Peoples.

"[The] Indigenous history that is being taught in schools is still not consistent across curricula nationally. Many educators and school districts are struggling to incorporate these subjects into classroom learning," says Furdyk.

Contest entries are judged by a panel of industry experts who evaluate the songs for the quality of their music and code, and the inclusion of social justice themes. The contest attracted 15,000 students in 2023; this year, more than 20,000 students were involved. The grand prize, a scholarship worth $5,000, is awarded to two students—one who is Indigenous and the other an Indigenous ally.

"This program is extremely powerful in terms of bringing important Indigenous topics to light, and exposing students to Indigenous musical role models," says Furdyk. "The students aren't just consumers, they are getting to be creators and integrate their perspectives and feelings about these topics into their music." &

CODING AND PROGRAMMING | TECHNOLOGY AND SOCIETY

**Want to bring the Your Voice is Power competition to your classroom? Check out the Teacher Checklist:** ***yourvoiceispower.ca/teacher_checklist***

# Unplugged Computer Science Activity

BY TANNER BIG CANOE

## OVERVIEW:

**Topic:** Algorithmic Searches

## LEARNING OBJECTIVES:

### Learning Goals:

- Explore concepts of time complexity, algorithms, and search optimization
- Understand sorting algorithms using a deck of cards
- Create, define, and test a sorting algorithm

### Success Criteria:

- Explore the processes of linear and binary searches (sorting algorithms)
- Create and define a unique search algorithm, complete with written rules

### Essential Questions for Activity:

- What is an algorithm?
- What is a search algorithm?
- How can search algorithms be compared?
- Can we create a new search algorithm?
- How does the most effective search algorithm change given the size of the data?

PHOTOS COURTESY OF AMPERE

## BREAKDOWN:

### Minds-On

**Time: 5 minutes**

One of the most important concepts in computer science is searching. Computers have to search through large amounts of data to find exactly what the user has requested. Computer scientists have created unique algorithms that allow users to search through data in different ways. In order to compare the efficiency of these algorithms, computer scientists have also created a way to measure the effectiveness of the algorithm called time complexity, which allows comparison of which algorithms perform best under different circumstances. In this activity, students are introduced to these concepts through a simpler version of sorting, searching, and timing using playing cards.

**Sort the class into groups of three or four learners (depending on class size). Each group will need one full deck of cards. Additionally, each learner will require a piece of paper and a pencil for part three of the activity, when they will create and define search algorithms with accompanying rules.**

### Action

**Time: 55 minutes**

Each group is given a deck of cards to test two search algorithms: linear and binary searches. Remind students that computers can only examine one piece of data at a time when executing an algorithm. To reflect this, only **one card** can be flipped over at a time.

# ACTIVITY 1
## Linear Search

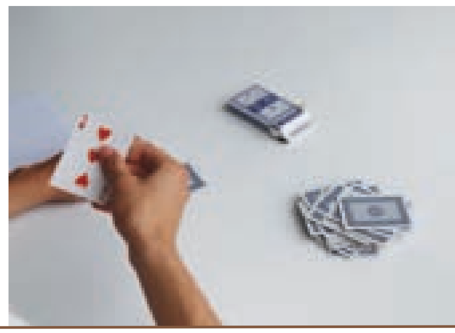
*How to Play:*

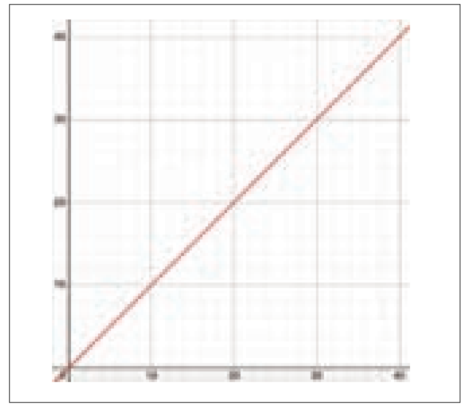
1. The teacher identifies a specific card in the deck (e.g. the three of Diamonds).
2. Players sit in a circle with the deck of cards in the middle, face down. For this section the cards can be shuffled or sorted (see Activity 2 for how to sort cards). This showcases the idea that linear searches work for both random and sorted data.
3. Players take turns drawing the top card from the deck and placing it face up next to the deck.
4. Play continues until the target card is drawn.

This game can be played as a class, with all the groups playing at the same time to see who finishes first, or having the groups play independently, timing the game on a device if they have one, and then taking note of it.

Explain that the time to conduct a linear search increases in proportion to the amount of data the computer is being asked to search through. That is, the more data there is, the more time the search will take on average.

Shuffle the deck and play the game again with the same target card. This will help students understand the average amount of time it takes to find the target card, as well as the process of searching one element at a time. This activity can be extended by having the students change the number of cards they use to showcase that having less data to search through results in less time on average to find the target card with a linear search. To demonstrate that, on average, searching through less data requires less time to locate the target card, have students remove all face cards from the deck before starting their search.





*This linear graph shows the relationship between data and time in a linear search. In this graph the x-axis is the amount of data, and the y-axis is time.*

# ACTIVITY 2
## Binary Search

The binary search operates with data that is sorted, so for this exercise, the deck of cards should be ordered before play. Cards can be sorted in a variety of ways, given the variables of value, suit, and colour. One example is to sort the cards in ascending order, by colour, and by suit—which in one iteration would sort the deck by grouping Aces, then Twos, all the way to Kings, with each value ordered by colour (Red, Black), then suit (Diamonds, Hearts, Clubs, and Spades). The deck should be sorted in a way that allows players to determine where the target card is in relation to any card pulled from the deck.

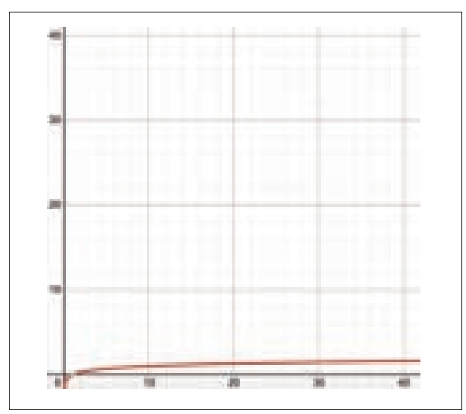
Binary searches work by assessing the amount of data to be searched then repeatedly splitting the data in half, and assessing which of the halves contains the target of the search. For example, if the target card is the Five of Hearts, split the deck in half. Now, determine which half of the deck contains the target card. In this example, the 26th card is the Seven of Hearts, so the target card is in the half of the deck that contains that Seven of Hearts. That half of the deck is now cut in two at the 13th card, and so on until the only card remaining is the target.

*How to Play:*

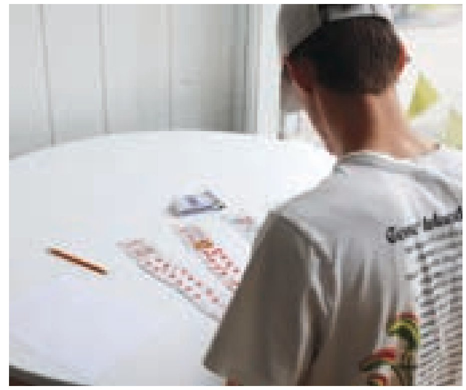
1. Sort the deck of cards according to a defined order. The example given above is great for an initial run through. (Aces through Kings, in the order of Diamonds, Hearts, Clubs, and Spades)
2. How many cards are there? Halve that number, then split the deck at that point. (If the number of cards is odd, split it at one-half-plus-one.)
3. Turn the top half of the deck face up. If the card that shows up is the target card, end the timer. If it is not, determine which half of the deck contains the target card.
4. Repeat steps 2 and 3 until the target card has been found. End the timer.

Students may find that the binary search takes longer than linear search with this amount of data. Ask students how this might change if they were given a larger set of data.

Activity 1 demonstrates that search time in a linear search increases linearly based on the amount of data. The binary search process is logarithmic rather than linear. As the amount of data increases, the search time increases but the rate at which it does this decreases over time. This can be demonstrated with a logarithmic graph.



*This logarithmic graph shows the relationship between data size and time to search. In this graph the x-axis is the amount of data, and the y-axis is time.*

## ACTIVITY 3
### Create Your Own Search and Sort

Have the students create their own way of sorting the cards and searching through to find the target card. The only constraints are that the cards must be in a pile one on top of the other to begin, and players may only view one card at a time.

***Ask students:***
Is there a way to sort the cards initially that would decrease the time spent searching?

Is there a way to search for the target card that would decrease the time spent searching? Have students brainstorm and, if they are stumped, conduct research into other existing search algorithms and apply it to the deck of cards.

Have the students compare the average time it took their algorithms and sorting methods created in Activity 3 to the previous examples of linear and binary searches.

Has the time taken to search for the card decreased on average?

An expanded version of this activity could have students write out the steps followed in the algorithm they have created and have other students try to follow it to test the method. This is a great way to demonstrate the sequential processing of a computer and that steps need to be very precise.

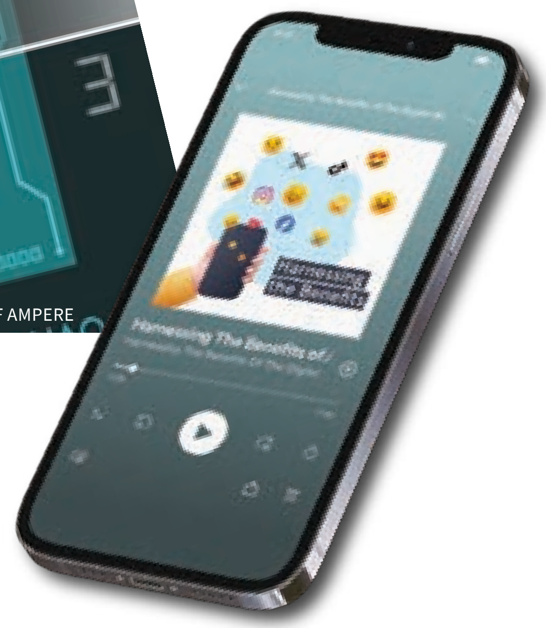
CODING AND PROGRAMMING

COMPUTING AND NETWORKS  DATA

### Details

**Materials Needed:**
- Deck of cards (1 per group)
- Paper and pencils

### Post-Lesson Reflections
**What worked?**
**What didn't?**
**Why?**

PHOTOS COURTESY OF AMPERE

# Digital Kit

### PAST ISSUES

If you missed past issues of *Root & STEM*, you can find them online at
rootandstem.ca

We have developed additional digital resources for educators to use in the classroom that connect to the concepts of computer studies—including podcasts and interactives. They can be accessed online via the links that follow.

### Root & STEM Podcast

This podcast expands on *Root & STEM* content and invites subject experts from each issue of the magazine to share their knowledge. The current series explores the theme of computer studies in relation to STEAM education. Episodes are approximately 15–30 minutes long and are available on the Ampere website or your preferred platform.
rootandstem.ca/learn/the-root-stem-podcast

### Root & STEM App

Filled with the same informative articles, podcasts, and lesson plans as its printed counterpart, the Root & STEM educational app is a free digital resource for K–12 educators and learners of all ages. The app puts the magazine's STEAM content and curriculum in the palm of your hand. Interactive elements are added regularly. Available for download on the App Store and the Google Play Store.

### Harnessing the Benefits: The Digital World Podcast

This three-episode series designed for Grade 9–12 students highlights the advantages of technology while introducing some potential dangers of the online world and how to avoid them. Each episode focuses on one popular aspect of the digital world: artificial intelligence, social media, and online learning. Each episode explains how these relatively new technologies can benefit students in and out of the classroom, how others are using the technology, and how to implement digital safety to harness the benefits of the digital world.
rootandstem.ca/learn/harnessing-the-benefits/

### All About AI

This video takes students on a journey through time to learn about the history of artificial intelligence (AI). It explores how AI came to be and profiles key scientists who first thought of creating intelligent computers, which are capable of using logic and mimicking thought processes to solve problems. Students learn what early models of AI looked like, what the intelligent machines of the past were, and how far technology and AI have come in the last two decades. The video also focuses on where we currently stand with AI-powered technology and what the future holds for humans and AI.
youtube.com/watch?v=iencUcExpRo

### How to Assemble a Computer

This video tutorial guides learners through the parts of a computer and how to build their own.
amp.ca/learn/how-to-assemble-a-computer

### Gon' Phishing

Phishing scams are a particularly insidious element of the online experience. Gon' Phishing is a game built to provide an opportunity to experience a variety of phishing attempts and become familiar with the signs of phishing emails. In the game, the user must determine if incoming emails are "Legit" or "Phish." If players mark emails incorrectly, they receive more emails, which clog up their inboxes!
amp.ca/learn/gon-phishing

LESSON 1

# Make a Medical Alert Fall Detector

**Author:** Science Buddies

**Level:** Grades 9 to 10
**Subject:** Computer Studies
**Software:** micro:bits

## CURRICULUM LINKS

This module aligns with provincial and territorial computer studies curricula for Ontario Grade 10.

### Computer Science:

**A1. Computational Thinking, Planning, and Purpose**
*Apply computational thinking concepts and practices, and use various tools and processes to plan and develop computational artifacts for a wide variety of contexts, users, and purposes*

**B1. Understanding Hardware and Software**
*Demonstrate an understanding of the functions and features of the hardware and software they encounter in everyday life*

**B2. Using Hardware and Software**
*Demonstrate an understanding of various ways to use hardware, software, and file management, and of research practices to support their use of digital technology*

**C1. Programming Concepts and Algorithms**
*Explain fundamental programming concepts and algorithms*

**C2. Writing Programs**
*Use fundamental programming concepts to write simple programs*

### Materials:

- micro:bit Go Bundle. This kit contains everything needed to get started working with a micro:bit:
  - micro:bit board
  - micro-USB cable
  - battery holder
  - AAA batteries (2)
- computer with internet access and USB port

In this project, learners program a fall detector using a pocket-sized programmable computer called a micro:bit and software called MakeCode. Devices like these are used by many people with a variety of medical conditions.

### Learning Objectives:

- Understand what medical fall and medical alert devices are used for
- Compare methods of sending signals and determine which work best over short and long distances
- Interpret graphical data from microphones, light sensors, and accelerometers and explain what might cause changes in a graph

### Vocabulary:

- **Sensor:** A device that detects or measures a physical property and records, indicates, or otherwise responds to it
- **Acceleration:** An increase in the rate or speed of an object or person
- **Accelerometer:** An instrument for measuring acceleration, typically that of an automobile, ship, aircraft, or spacecraft
- **Gravity**: The force that attracts a body towards the centre of the Earth, or towards any other physical body having mass
- **Fall detector:** A device that helps reduce the risk of long-term injury by detecting a person's abrupt change in position
- **Fall monitor:** An assistive device, commonly used by the elderly. Its primary purpose is to notify a carer or family member when a fall occurs
- **Medical alert device:** Technology that tends to the complex physiological monitoring of patients
- **Medical alarm:** An alarm system designed to signal a hazardous situation requiring urgent attention and to summon emergency medical personnel
- **Pitch angle:** In engineering, the angle between a slanted gear (called a bevel gear) and its centre line
- **Light-emitting diode (LED):** A semi-conductor device that emits light when an electric current flows through it

### Guiding Questions:

1. What sensors are built into the micro:bit?
2. What are some examples of fall detectors and medical alert devices that can be bought?
3. What features do you want to program into a micro:bit fall detector?

## INTRODUCTION

Electronic sensors called accelerometers are installed in smartphones and video game controllers. Accelerometers measure acceleration, which is a change in speed. Video game controllers rely on accelerometers to detect if they are being shaken. Accelerometers can also detect the direction of gravitational force, and this allows devices to determine which way is up and which way is down. For example, this is how a phone knows to rotate the screen when it is turned sideways.

The micro:bit has many built-in sensors, including an accelerometer.

## COMPUTER ACTIVITY

### STEP 1

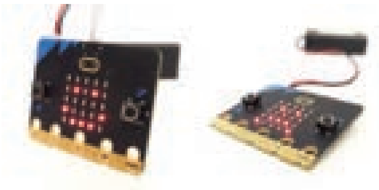
Start a new MakeCode program, then access the **Fall detector smiley face** program on the MakeCode website at **makecode.microbit.org/S92388-40846-91439-76595**.

This program makes the micro:bit display a smiley face when it is standing upright and a frowning face when it is lying flat. To get started using a micro:bit, follow the instructions here: **microbit.org/get-started/getting-started/introduction**

### STEP 2

Download the smiley face program code from the site to the micro:bit and test it:

a. Stand the micro:bit upright on its front edge as shown at the left side in the figure below. In this position, the pitch angle is 90 degrees. What does the micro:bit display?



b. Lay the micro:bit flat on its back with the LEDs facing up. In this position, the pitch angle is zero degrees. What does the micro:bit display?

c. Tilt the micro:bit forward so the LEDs are facing down (hold it up in the air to observe the LEDs). In this position, the pitch angle is 180 degrees. What does the micro:bit display?

### STEP 3

Examine the code from the MakeCode site to understand how it operates. The key to this program is the **if/else statement**. The code tells the micro:bit **"IF the pitch angle is outside a certain range, show a frowny face. ELSE, show a happy face."** The condition for the if statement will be true if the **pitch** variable is less than the **angle1** variable OR if the **pitch** variable is greater than the **angle2** variable. As a result, when the micro:bit is "vertical," it shows a smiley face. When it "falls over" (whether backwards or forwards), it shows a frowny face.

### STEP 4

Experiment by changing the **angle1** and **angle2** variables and downloading the new code. What happens if the angle values are made larger? Smaller? Consider which values might work best for a fall detector when a person is wearing the micro:bit in their pocket or clipped to their belt.

### STEP 5

Use the "Science Buddies—Engineering Design Process" resource (see link below) to design a fall detector. There is no right or wrong way to do this, but here are some suggested features that can be included:

a. Program a help button using the **on button A pressed** block under **Input** in the MakeCode menu.

b. Make the micro:bit play a sound at the same time that it displays something with the LEDs.

c. Experiment with the **on shake** block under **Input**. This block allows the micro:bit to detect a variety of motions using its accelerometer (the motions that are detected are selected using the drop-down menu). Do any of the options available make sense as elements of a fall detector?

d. Try using the **on loud sound** block. Does this block work to detect loud sounds like a person falling or calling for help?

### STEP 6

Test the fall detector

a. Use the battery pack that comes with the micro:bit so it does not need to be plugged in.

b. Wear the micro:bit or ask a volunteer to do so. It can be attached to clothing or carried in a pocket.

c. Try some (gentle!) practice falls, such as a trust fall into a classmate's arms. Does the fall detector detect the falls? If not, try changing elements of the code, like the values of the **angle1** and **angle2** variables, or other conditions that have been set.

## CONCLUSION

At the end of this lesson, students should be familiar with advanced features of the micro:bit and be able to modify code.

**Resources:**

- **Micro:bit Educational Foundation—Getting Started** (microbit.org/get-started/getting-started/introduction)
- **Microsoft MakeCode—Rotation** (makecode.microbit.org/reference/input/rotation)
- **Science Buddies—Accelerometer Technical Note** (sciencebuddies.org/science-fair-projects/references/accelerometer)
- **Science Buddies—Engineering Design Process** (sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-process-steps)

LESSON 2

# Introduction to Arduino: Setting Up the Arduino

**Author:** Science Buddies

**Level:** Grades 6 to 12
**Subject:** Computer Studies, Art/Arts
**Software:** Arduino

## CURRICULUM LINKS

This module aligns with provincial and territorial arts, and computer studies curricula for Ontario Grade 10.

### The Arts:

**A2. Elements and Principles:**
*Apply key elements and principles from various arts disciplines when creating, modifying, and presenting art works, including integrated art works/productions*

**A3. Tools, Techniques, and Technologies:**
*Use a variety of tools, techniques, and technologies to create integrated art works/ productions that communicate specific messages and demonstrate creativity*

### Computer Studies:

**A1. Computational Thinking, Planning, and Purpose**
*Apply computational thinking concepts and practices, and use various tools and processes to plan and develop computational artifacts for a wide variety of contexts, users, and purposes*

**B1. Understanding Hardware and Software**
*Demonstrate an understanding of the functions and features of the hardware and software they encounter in their everyday life*

**B2. Using Hardware and Software**
*Demonstrate an understanding of various ways to use hardware, software, and file management, and of research practices to support their own use of digital technology*

**C1. Programming Concepts and Algorithms**
*Explain fundamental programming concepts and algorithms*

**C2. Writing Programs**
*Use fundamental programming concepts to write simple programs*

### Materials:

- Arduino UNO
- Science Buddies Electronics Kit for Arduino, available from Home Science Tools®
- Computer
- USB-C to USB-A adapter (for newer computers that only have USB-C ports)

This lesson introduces students to physical computing: the process of building circuits and programming a microcontroller (in this case, anArduino Uno®) to interact with users.

### Learning Objectives:

- Connect an Arduino Uno board and upload code to it
- Edit an existing Arduino program

### Vocabulary:

- **Arduino:** An open-source electronics platform based on easy-to-use hardware and software
- **Microcontroller:** A control device that incorporates a microprocessor
- **Robotics:** The branch of technology that deals with the design, construction, operation, and application of robots

### Guiding Questions:

1. What do you know about robotics?
2. Where can robotics be useful?

## INTRODUCTION

This lesson introduces students to the basics of working with an Arduino Uno board and Arduino open-source software: writing and uploading programs, building simple circuits, and interfacing with inputs and outputs like buttons and LEDs. This sets the stage for more advanced projects, like building robots, drones, voice-controlled appliances, etc.

## COMPUTER ACTIVITY

### STEP 1
Download and install the Arduino IDE open-source software from **www.arduino.cc/en/software.**

***NB:** The Arduino IDE is maintained and updated on a regular basis.*

### STEP 2
Open Arduino. Follow any on-screen prompts to download and install updates.

### STEP 3
Plug the Arduino Uno board into the computer with a USB cable.

### STEP 4
Click the **Select Board** drop-down menu and select **Arduino Uno**.

### STEP 5
Select **File**, then **Examples**, then **01.Basics**, then **Blink** to open the built-in Blink example program.

### STEP 6
Click the **Upload** button to upload the Blink program to the Arduino Uno board.

### STEP 7
Look at the Arduino Uno board. The onboard LED should blink on for one second, then off for one second, then repeat.

### STEP 8
Read through the annotated Arduino code.

```
// the setup function runs once when you press reset or power
// the board
void setup() {
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
    }
// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
        // turn the LED on
        // (HIGH is the voltage level) delay(1000);
    delay(1000);
        // wait for a second
    digitalWrite(LED_BUILTIN, LOW);
        // turn the LED off by making
        // the voltage LOW
    delay(1000);
        // wait for a second
```

***NB:** If students have no prior programming experience, supplement the video with further explanation. See the resources section for the link to the code.*

### STEP 9
Edit the program so the LED stays on for two seconds and off for half a second. ***NB:** Explain that the delay command uses milliseconds. There are 1,000 milliseconds in 1 second.*

## CONCLUSION

At the end of this lesson, students should be familiar with the basics of the Arduino Uno board and the Arduino IDE, and be ready to explore the many other features they have to offer. This is just one introductory lesson to Arduino products—more can be found here: **www.sciencebuddies.org/teacher-resources/lesson-plans/introduction-to-arduino**

**Resources:**

- **Arduino**
  (arduino.cc)
- **Technical Specs of your Arduino Uno board**
  (arduino.cc/en/Main/Products)
- **How to Use an Arduino**
  (sciencebuddies.org/science-fair-projects/references/how-to-use-an-arduino)
- **Language Reference**
  (arduino.cc/reference/en)

LESSON 3

# Code Phi Spirals

**Author:** Ampere

**Level:** Grades 9 to 10
**Subject:** Computer Studies
**Software:** Scratch

***NB:** This lesson is intended for advanced users of Scratch*

## CURRICULUM LINKS

This module aligns with provincial computer studies curricula for Ontario Grade 10.

### Computer Studies:

**A1. Computational Thinking, Planning, and Purpose**
*Apply computational thinking concepts and practices, and use various tools and process to plan and develop computational artefacts for a wide variety of contexts, users, and purposes*

**B1. Understanding Hardware and Software**
*Demonstrate an understanding of the functions and features of the hardware and software they encounter in their everyday life*

**B2. Using Hardware and Software**
*Demonstrate an understanding of various ways to use hardware, software and file management, and of research practices to support their own use of digital technology*

**C1. Programming Concepts and Algorithms**
*Explain fundamental programming concepts and algorithms*

**C2. Writing Programs**
*Use fundamental programming concepts to write simple programs*

### Materials:

- Computer and internet
- Scratch accounts (if needed, create an account at **scratch.mit.edu**)
- Scratch Example (available at **scratch.mit.edu/projects/888242258**)

In this lesson, students dive into phi numbers and use them to code a sequence to create a phi spiral in Scratch.

### Learning Objectives:

- Understand phi numbers and golden ratios in depth
- Create and use a formula to create a phi spiral in Scratch
- Understand the code and develop a basic understanding of variable and methodical operations in Scratch

### Vocabulary:

- **Phi number/Golden ratio:** a ratio between two numbers that equals approximately 1.618
- **Variable:** a quantity that is assumed to vary or be capable of varying in value in the course of a calculation
- **Mathematical operators:** A symbol that stands for carrying out one or more mathematical operations on some function

### Guiding Questions:

1. What is the Fibonacci sequence, and where is it used?
2. What is coding? What skills might we need to carry out this code?
3. What are Phi Spirals? How do they determine symmetry in nature?

## COMPUTER ACTIVITY

In this project, students build code in Scratch to generate a phi spiral, which is a spiral that moves farther from its origin by a factor of phi ($\varphi$) with every quarter turn it makes.

### STEP 1

Log into your Scratch account. Click on **Create** to start a new project. Name the project “Phi Spiral Sequence.”

### STEP 2

**Draw seven sprites.**

- A dot labelled **Sprite1**—create it using the paint button in the **Costumes** tab
- A blank sprite labelled **Centre**
- A button labelled **Pi** with the symbol $\pi$ in it
- A button labelled **Golden Ratio**
- A button labelled **E (which stands for Euler’s Number)**
- A sprite labelled **22/7**
- A button labelled **¼ Turn**


**Create seven variables:**

- angle
- clone#
- number of colours
- step
- turn ratio
- turn ratio denominator

## STEP 3

Using the values provided below, apply base code instructions to the **Pi, Golden Ratio, E, 22/7**, and **¼ Turn** buttons. To do this:

- Select the button
- Add a **when this sprite clicked** block from the **Events** category.
- Add a **set __ to** block from the **Variables** category and select the **turn ratio** block created in Step 2.

Set the variable values in the set to blocks as follows:

**For Pi**

- Variable value: **3.141592653589793238462643383279**


**For Golden**

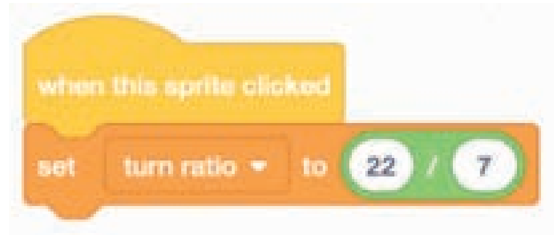
- Variable value: **1.6180339887498948482045868343655**

**For (E) Euler's Number**

- Variable value: **2.718281828459045235360287471352**

**For 22/7**

- Go to the **Operators** category. Enter a **() / () (division)** block in the **set to** block. Enter **22** in the first space and **7** in the second.




**For ¼ Turn**

- Variable value: **0.25**


## STEP 4

**4.1** Code the Sprite1 dot. Add a **when green flag clicked** block from the **Events** category, followed by a **set (my variable) to block from the Variable** category. Set my variable to **clone#** and enter **0** (zero) in the value space.

**4.2** Add a **forever** block from the **Control** category. Inside this block, add a **turn__degrees (clockwise)** block from the **Motion** category. This generates code that instructs the spiral to rotate continuously clockwise.

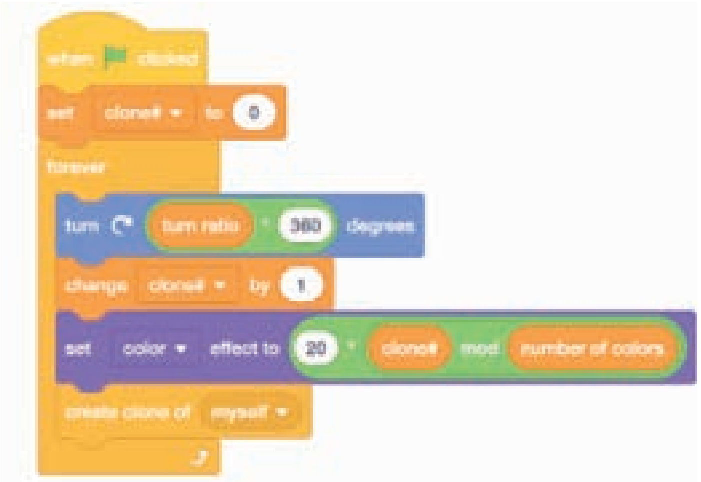
**4.3** Place a **() / ()** block from the **Operators** category inside the **turn__ degrees** block. Select **turn ratio** as the variable in the first space and enter **360** as the value in the second space.

**4.4** Add a **change __ by** block from the **Variable** category. Select **clone#** as the variable and assign **1 (one)** for the value. This dictates that one clone is created at a time. This action happens quickly in Scratch, so it can seem as though multiple clones are created. In essence, this tells the code how many clones to create in one instance.

**4.5** From the **Looks** category, place **the set color effect by** block underneath the previous block. Place another **() / ()** block in the value space. In the first value of the **() / ()** block, type the number **20**. In the second, place a **_Mod_** block from the **Operators** category. In the first **_Mod_** value space, put the **clone#** variable. In the second, place the **number of colors** variable.

This code specifies how many clones will be created and the frequency of colour changes.

**4.6** Finally, add a **create clone of myself** block from the **Control** category. Now the code will only clone the Sprite1 dot.




### STEP 5

We will now code the size and number of dots that show on the screen when the green flag is pressed for Sprite1.

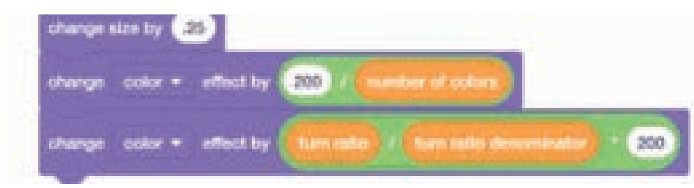
**5.1** Add a **change size by** block from the **Looks** category and enter **0.25** as the value.

**5.2** Add a **change color effect by** block from the **Looks** category. In the value space, enter a **() / ()** block from the **Operators** category. Enter **200** in the first value space and a **number of colours** variable block in the second. This tells the code to change the color of the dots as the spirals increase or decrease in size.

**5.3** Add another **change color effect by** block and enter a **() / ()** block in the value space.

**5.4** Within this operator block, in the first value, add the **() / ()** block. In the second value, type **200**. Place the **turn ratio** variable in the first division value. In the second value, place the **turn ratio denominator** variable.

**5.5** Insert another **()__()** block in the first value space, and enter **200** in the second. In the first value space of the embedded **() / ()** block, enter a **turn ratio** block from the **Variables** category. in the second, enter a **turn ratio denominator** block from the **Variables** category.




### STEP 6

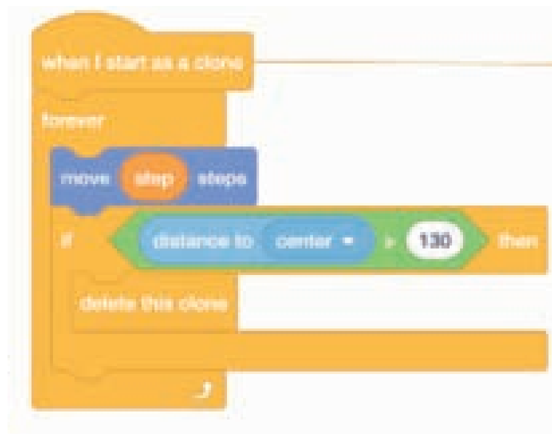
The final, separate piece of the code creates a sliding bar that speeds up or slows the rotation of the dots.

**6.1** From the **Control** category, select a **when I start as a clone** block and follow it with a **forever** block.

**6.2** Insert a **move __ steps** from the **Control** category in the forever block. Insert **step** from the **Variables** category in the value space.

**6.3** Add an **if ... then** block from the **Control** category. Insert a **__>__** block from the **Operators** category in the value space after **if**. In the value space, place a **__>__** block from the **Operators** category. In the **() / (>50)** block, insert a **distance to __** block from the **Sensing** category in the first value space and select **centre** as the value. In the second value space, enter **130**.

**6.4** In the space in the **if ... then** block, insert a **delete this clone** block from the **Variables** category.




### STEP 7

Test the code by clicking the green flag to start the spiral.

***NB:** Pressing the various buttons (E, pi, etc) changes the structure of the spiral. There is a link to the complete and correct code below in the references.*

## CONCLUSION

At the end of this lesson, students should be able to complete the phi spiral code and fix any errors they come across, as well as manipulate the code to create more or fewer dots and alter the degree at which they turn.

**Resources:**

- **Scratch Example** (scratch.mit.edu/projects/888242258)
- **Scratch Website** (scratch.mit.edu)
- **What is the Fibonacci Sequence & the Golden Ratio? Simple Explanation and Examples in Everyday Life** (youtube.com/watch?v=2tv6Ej6JVho)

LESSON 4

# Animate a Sprite with Scratch

**Author:** Canada Learning Code

**Level:** Grade 10
**Subject:** Computer Studies, Art/Arts
**Software:** Scratch

## CURRICULUM LINKS

This module aligns with provincial and territorial arts, language arts and computer studies curricula for Ontario Grade 10.

### The Arts:

**A2. Elements and Principles:**
*Apply key elements and principles from various arts disciplines when creating, modifying, and presenting art works, including integrated art works/productions*

**A3. Tools, Techniques, and Technologies:**
*Use a variety of tools, techniques, and technologies to create integrated art works/productions that communicate specific messages and demonstrate creativity*

### Computer Studies:

**A1. Computational Thinking, Planning, and Purpose**
*Apply computational thinking concepts and practices, and use various tools and processes to plan and develop computational artifacts for a wide variety of contexts, users, and purposes*

**B1. Understanding Hardware and Software**
*Demonstrate an understanding of the functions and features of the hardware and software they encounter in their everyday life*

**B2. Using Hardware and Software**
*Demonstrate an understanding of various ways to use hardware, software and file management, and of research practices to support their own use of digital technology*

**C1. Programming Concepts and Algorithms**
*Explain fundamental programming concepts and algorithms*

**C2. Writing Programs**
*Use fundamental programming concepts to write simple programs*

### Language: Oral Communication:

**Listening to Understand, Using Active Listening Skills**

1.2: Select and use appropriate active listening strategies when participating in a variety of classroom interactions

**Reflecting Skills and Strategies, Interconnected Skills**

3.2: Identify a variety of their skills in viewing, representing, reading, and writing and explain how the skills help them improve their oral communication skills

### Materials:

- Scratch: **scratch.mit.edu/**
- Class set of laptops/chargers
- Mice
- Headphones

In this activity, students create and animate a sprite in Scratch using code and algorithms. By the end of the lesson, students will have gained confidence in using Scratch and programming sprites to move.

### Learning Objectives:

- Provide learners with the opportunity to execute code in investigations
- Identify and describe the impacts of coding and of emerging technologies on everyday life
- Be able to explain the use and purpose of animations in their own lives as well as in the world of technology

### Vocabulary:

- **Algorithm:** A step-by step set of instructions that is performed to help solve a problem
- **Bug:** An error in a program that prevents the program from running as intended
- **Debugging:** The act of finding and solving problems in code
- **Loops:** Allows the same sequence of code to run multiple times, if the predefined conditions are met

### Guiding Questions:

1. Why do people use animations?
2. What problems do animations create?
3. What do we know about coding already? What processes do you think we will need to use to create this project?

## COMPUTER ACTIVITY

### STEP 1

Go to **scratch.mit.edu** to work online or download the application at **scratch.mit.edu/download** to work offline. If online, click **Create** in the top left corner.

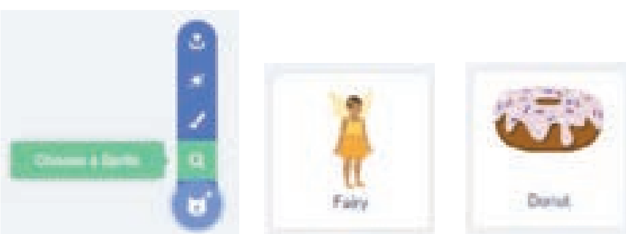
**OPTIONAL (recommended):** Create an account/sign in to save the progress of your project.

### STEP 2

Hover over the purple **Choose a Sprite** icon located at the bottom right corner of the screen, then click on the magnifying glass. Click on the sprite you'd like to use! Tip: Hover over each sprite to see the various options there are for that particular sprite. Some may have different colour options or positions. These are called **costumes**.

***Advanced option:*** *If students would like to create a sprite from Scratch, they can hover over the* ***Choose a Sprite*** *icon, then click on the paint-brush. The tools outlined in step 3 (specifically brush, fill, line, circle, rectangle) will help them create a new sprite.*

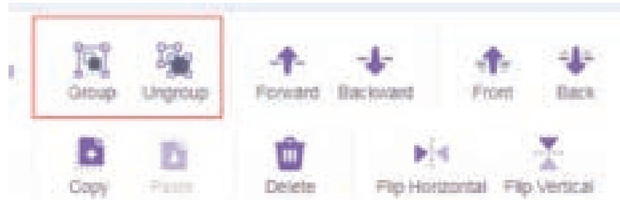
***NB:*** *Allow students 10 to 15 minutes to create their sprites before moving on to the next step.*




### STEP 3

1. Select the **Costumes** tab to access the **Paint Editor.**
2. A list of the costumes for the selected sprite will show up on the left-hand side. To duplicate a costume, right click one of the costumes and select **duplicate.**
3. Explore the various tools in the Paint Editor, which can be used to:
   - **Select, stretch, compress or rotate objects** by selecting the parts of the main sprite and adjusting the bounding box that appears when it is selected
   - **Reshape the edges of an object** by using the reshape tool located on the left-hand side
   - Draw new shapes using the **brush**, **line**, **circle**, and **rectangle tools** located on the left-hand side
   - Change colours using the **fill tool** located on the left-hand side
   - Delete elements using the **eraser tool** *(located on the left-hand side)* or the **undo** tool *(located on the top left side, beside the costume name)* to correct mistakes

   - **Group/ungroup** objects to select specific areas of the sprite, by selecting areas while holding down the **shift** button on the keyboard. Once the objects are selected, use the **group/ungroup** button, which is located at top left beside the forward/backward tool




   - Move selected objects **forwards or backwards** using the tool beside the group/ungroup tool


### STEP 4

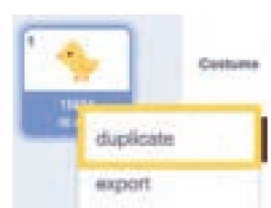
a) Depending on students' prior knowledge of Scratch, allow them to explore the program codes and animate the sprite they have chosen in a way that makes sense. (For example, if the sprite is a face, they might want to make it smile or frown; if it is a wolf, they could make it walk or howl.)

*If students are struggling or would benefit from examples, instruct them to follow the instructions below:*

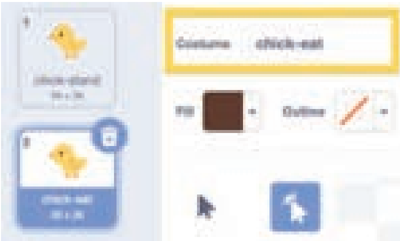
b) This portion of the lesson uses the chick sprite as an example to demonstrate how animated loops are created. Students animate the chick sprite to make it appear to bend over and open its mouth. They can refer to this example to apply the concept to their own character to animate their own sprites.

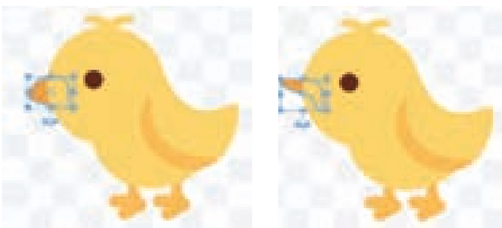
**4.1** Select the **Chick Sprite** and edit it in the **Costumes** tab

- Select the chick sprite by opening the **Choose a Sprite** icon located at the bottom right corner of the screen
- Rename the costume "chick-stand". Duplicate this costume by right clicking on its icon. Name the new costume "chick-eat"



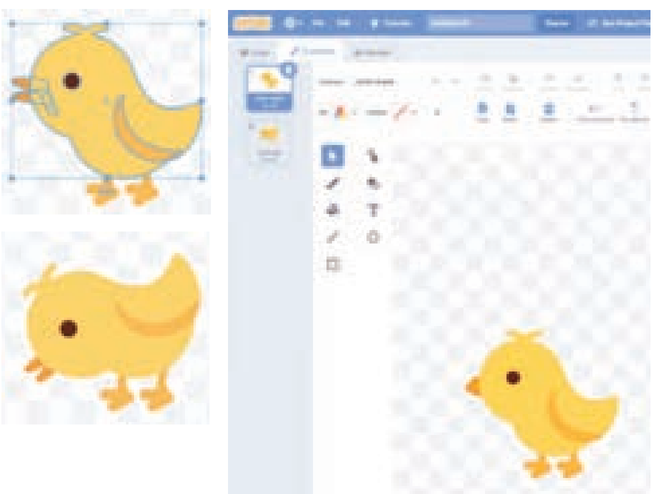
- Select the "chick–eat" sprite. Click the **reshape** tool, then click the chick's beak. Reshape it so it looks like a thin upper beak



- Select the new upper beak object and make a copy of it. Use the copy of the upper beak to create a lower beak. To do so, rotate the new object and move it backwards so it is placed behind the chick's head

- Select the chick's beak, eye, wing, and body, then group and rotate them such that the chick looks like it is bending over



**Code a way to switch between the costumes**

- Instead of toggling between sprites to make the chick look like it's eating, experiment with code to develop a way to create an animated sprite
- Insert a **forever** block from the **Events** category. Insert two **switch costume to __** blocks in the space. In the value space of the first **switch costume to** block, select **chick–stand**. In the second, select **chick–eat**. Test the code.

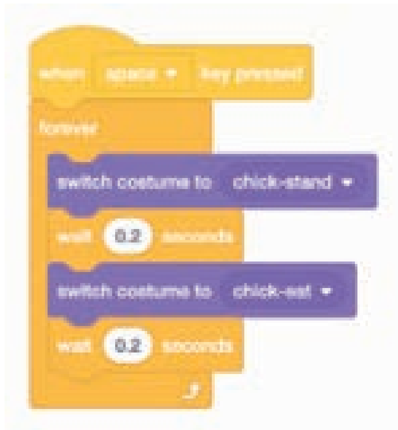
**To ensure the alternation between images is visible, slow down the animation**

- Between the two **switch costume to** blocks, add a **wait __ seconds** block from the **Control** category. Leave the value space at the default value of 1. Test the code to see how this timing changes the animation. Now, add a **wait 1 seconds** block after the second **switch costume to** block
- Adjust the timing of the animation as desired by changing the values in the **wait __ seconds** blocks

**Make the chick eat when a defined event happens**

- Add the desired event block from the **Events** category. Test out different ways to make the chick appear to eat

***NB:** Allow students time to create their code and make their sprite move. If students are struggling, urge them to chat with each other to solve any coding issues.*


## CONCLUSION

At the end of this lesson, students should be able to generate an animated sprite in Scratch. They should feel confident in their ability to animate a sprite as well as explain its purpose or why they have created it. By the end of this lesson, students should also feel comfortable with using Scratch and using basic code to create animated sprites.

**Resources:**

- **Scratch**
  (scratch.mit.edu)
- **Tutorial for Educators**
  (youtube.com/watch?v=4nfVirm1lkA)
- **Emoji: The Language of the Future | Tracey Pickett | TEDxGreenville (TEDx Talks)**
  (youtube.com/watch?v=Dzlek8nMrc8)
- **The History of Emoji (Tofugu)**
  (youtube.com/watch?v=SoZlB9pFV2M)
- **Inclusive and Accessible Emoji Practices for Our Increasingly Digital World**
  (feminuity.org/post/inclusive-and-accessible-emoji-practices-for-our-increasingly-digital-world)

# Discover Diverse Voices in STEAM Learning


Expanding on the print edition, the *Root & STEM* podcast introduces listeners to communities, projects, and activities that explore STEAM education in a digital world.

LISTEN ON Spotify | Listen on Google Podcasts | Listen on Apple Podcasts

**rootandstem.ca/learn/the-root-stem-podcast**

**Interested in following the journey of *Root & STEM*?** Follow Ampere on Facebook and Instagram to share thoughts about the podcast using the hashtag **#RootandSTEMPod.**