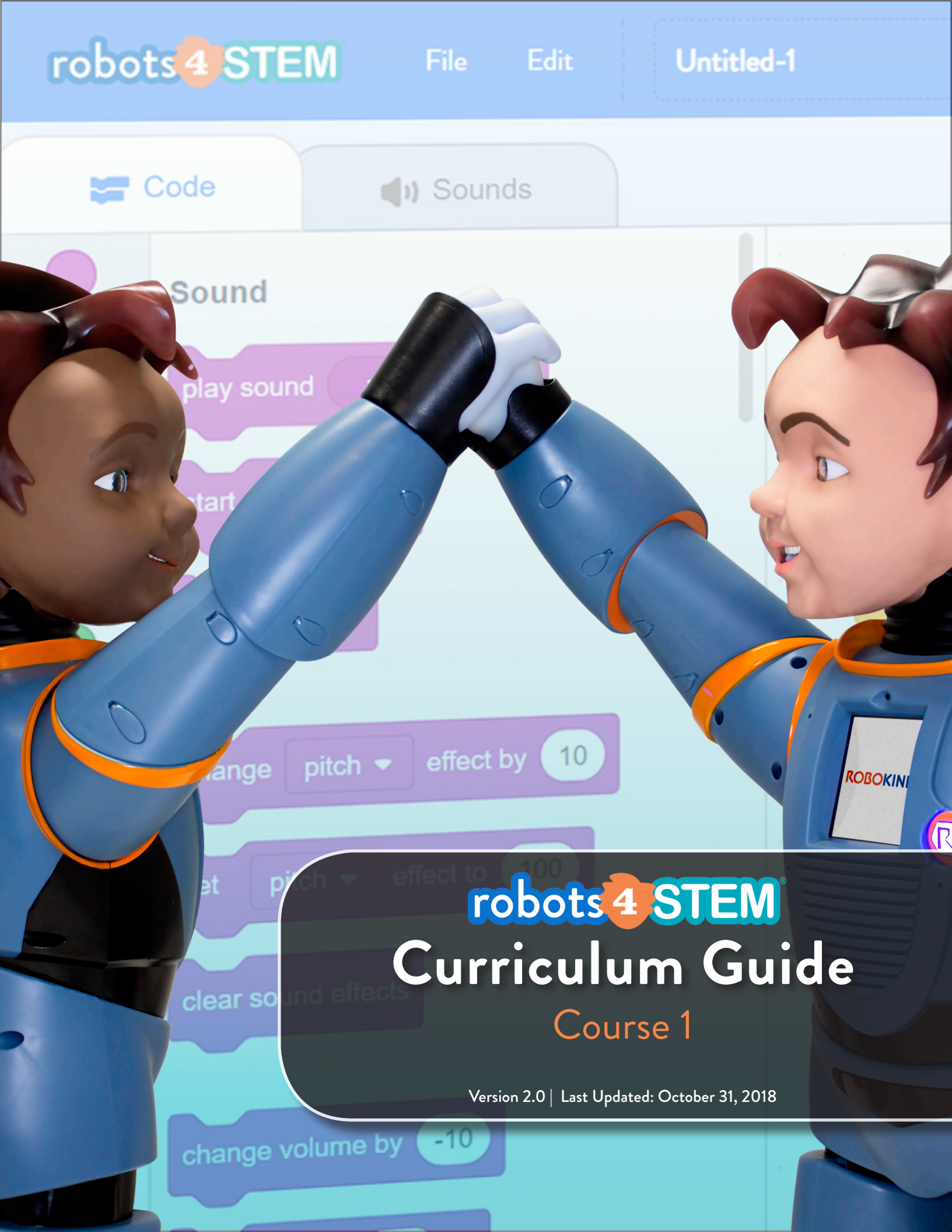


robots 4 STEM® Curriculum Guide

Course 1

Version 2.0 | Last Updated: October 31, 2018

Contents

Introduction to robots4STEM®

Welcome to **Course 1** of robots4STEM®! You and your students are about to embark on an engaging, educational journey to learn the basics of coding!

With hundreds of programming languages to choose from, knowing where to begin can be a daunting task. Therefore, to accelerate learning, robots4STEM® uses "visual block programming" that simplifies code into singular commands so that students can begin thinking like a programmer and creating functional programs before concerning themselves with syntax or which language to use.

You as the facilitator will be using this Curriculum Guide in conjunction with the lesson materials located on the Learning Management System (LMS) to steer your students to success; all while employing a real robot to provide students the opportunity to experience how they could apply what they are learning to the real world.

Neither you nor your students are required to have any experience with code (or robots for that matter) to be successful. This course is designed in such a way that **no prior knowledge is necessary**.

Within this Curriculum Guide, you will find:

- Mission [lesson] plans
- Notes about activities outside of the lessons
- Additional activities
- An appendix containing additional information and printable materials

If you need help at any point throughout the course, please do not hesitate to contact our support team.

Facilitating Student Data Preparation

Aim: Data is an important part of navigating the world of STEM and especially programming. Part of helping learners to start practicing their computational thinking is getting them comfortable with collecting, categorizing, and quantifying data.

Key Question: What is "data"?

- There are many types of data; however, we will be focused on numerical data.

Requirements:

Students will create their data representation in a manner that allows them to apply and demonstrate the following elements:

- Adding data to spreadsheet software or application
- Creating a simple graph from that data
- Saving files
- Retrieving (Opening) files
- Deciding the best way to represent their data such as:
  - ↳ Bar graphs
  - ↳ Pie graphs
  - ↳ Line graphs

Reflection:

Students will reflect on their use of data and the way that it is represented in relation to programming their code to run automatically (without facilitator or user intervention) with the goal of answering the following questions:

- What data was required?
- How did they arrive at the first value that I used to address the goal of the program?
- Did they have to adjust from the first value based on errors or feedback from the program?
- How did they arrive at the adjusted value?
- Did they record their values and changes in values?
- If yes, how did they record those attempts?
- Is there a better way for them to record those attempts?
- If no, why didn't they record the attempt?
- If no, how did not recording their data affect their programming?

Facilitating Student Presentations

Aim: Presentation and demonstration are important skills that students will need in many aspects of life but are especially important skills for working in the STEM industry.

Demonstrations will be structured to answer the following questions:

- What was the goal of the project?
- What steps did I take to address the goal of the project?
- How did I arrive at those steps?
- Were there any challenges in the programming and demonstration of the project?
- How did I address any challenges that arose in preparing?

Requirements:

Students will create their demonstrations using a presentation software/application that allows them to apply and demonstrate the following elements:

- Creating slides
- Editing the parts of a slide
    - ↳ Text
    - ↳ Images
    - ↳ Videos
    - ↳ Sounds
    - ↳ Formatting
- Using transitions
- Creating bulleted lists
- Using animations
- Editing text
- Add speaker's notes

Facilitating Student Reflection

Aim: Student reflection is the written conclusion to the iterative process that is building a program to meet a goal. Good programming should be able to be read and/or rebuilt by other programmers based on the information used and documentation collected. Students should answer several questions in their reflection document to reach the goal of programs that could be built by one of their peers.

- What was the goal that I was trying to meet with my program?
- What were the smaller goals that I recognized in the bigger goal?
  - ↳ Example: "I need all of the plants to be watered and harvested correctly" (Big problem)
  - ↳ "All of the plants require different amounts of water" (Small problem)
  - ↳ "Some of the plants need to be harvested at different times" (Small problem)
  - ↳ "Some plants need different amounts of water at different parts of life cycle" (Small problem)
- How did I solve the smaller problems? Students may refer reviewer (facilitator or another student) to their flowcharts.
- Which blocks did I apply to match the solution that I made for the smaller problems?
- Did I solve the bigger problem with the solutions to the smaller problems?
- If solving the smaller problems did not solve the bigger problem, why not?
- What blocks did I have to use in addition to my programming for the smaller problems to solve the bigger problem?

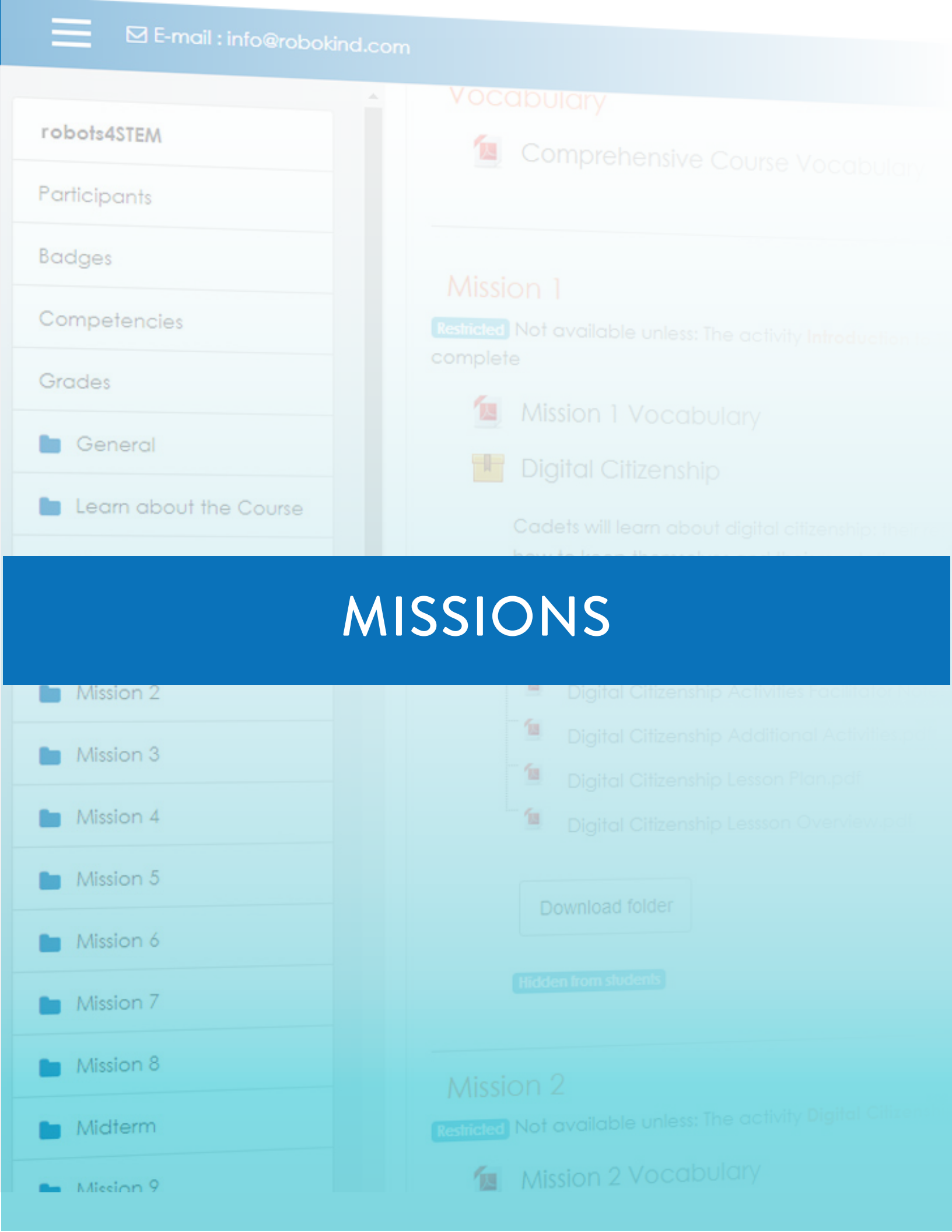
Requirements:

Students should be creating their reflection document in a manner that allows them to apply and demonstrate the following elements:

- Headers and Footers
- Citations for any research that they did for their projects or presentations
- Text boxes
- Bullets or Numbering

Students should demonstrate the ability to:

- Edit text
- Save a file
- Open a file


E-mail : info@robokind.com
robots4STEM
Participants
Badges
Competencies
Grades
General
Learn about the Course
Mission 2
Mission 3
Mission 4
Mission 5
Mission 6
Mission 7
Mission 8
Midterm
Vocabulary
Comprehensive Course Vocabulary
Mission 1
Restricted Not available unless: The activity Introduction to
complete
Mission 1 Vocabulary
Digital Citizenship
Cadets will learn about digital citizenship: the
Digital Citizenship Activities Facilitator
Digital Citizenship Additional Activities.pdf
Digital Citizenship Lesson Plan.pdf
Digital Citizenship Lessson Overview.pdf
Download folder
Hidden from students
Mission 2
Restricted Not available unless: The activity Digital Citizen
Mission 2 Vocabulary

MISSIONS

robots4STEM®

| MISSION AT A GLANCE |

MISSION 1: Introduction to Digital Citizenship

[1 hour, 30 minutes]

Mission Overview	Today's students are inundated with digital content, so it is imperative that students understand both the benefits and the dangers of having an online presence. In this mission, students will explore ways to stay safe on the Internet as well as being a positive influence in the online community.	
Mission Objectives	• Explore the concept of digital citizenship, including responsibilities in the online space and how to keep themselves and their reputations safe while online • Complete Reputation building exercises, devise strong passwords, and develop strategies to combat cyberbullying	
Core Concepts Subconcepts. Crosscutting Concepts	**Networks and the Internet** **Cybersecurity.** Privacy and Security	
Connections within Framework	3-5. Impacts of Computing. Safety, Law, and Ethics	
Materials	Online content Roles and Responsibilities Overview of Skit or Song assignment Rubric for Skit or Song Reflection Tool Access to Online Content	
Introduction to robots4STEM®	• Overview of Program including final outcomes • Introduce the Online Content – model access • Discuss Expectations for independent work	15 minutes
Online Work	• Complete mission • Complete quiz	15 minutes
Group Work	• Plan a song or skit about Cyberbullying (Groups of 3-4) • Go over Roles and Responsibilities for Presentations • Review Rubric for Song or Skit	30 minutes
Group Presentations and Debrief	• Each group presents 5-minute skit or song • After all groups present – take 5 minutes for each group to complete reflection • Each group shares one quadrant of their reflection with the class	30 minutes

MISSION 1

Introduction to Digital Citizenship
MISSION PLAN

Subject	Grade	Mission Length
Learn about responsibility and safety when online.	3rd-5th	60 minutes

OVERVIEW

In this mission, students learn foundational concepts of Digital Citizenship

	Facilitator Guide	Notes
Objectives	In this mission, students will learn and demonstrate the following concepts: • Recognizing cyberbullying • Judging a strong password • Comparing rights and responsibilities in relation to digital citizenship • Recognize respectful communication online • Recognizing fake or malicious websites	
Important Vocabulary	1. Digital Citizenship 2. Reputation 3. Cyberbully 4. Right 5. Responsibility	1. Digital Citizenship- Being a member of an online society, having certain rights and responsibilities. 2. Reputation- The opinion that other people have about you. 3. Cyberbully- Someone who uses electronic communication to threaten or intimidate others. 4. Right- The freedoms available for being a member of a community. 5. Responsibility- Actions we should do for the good of all members of a community.

	Facilitator Guide	Notes
Activity	"Password Protectors" online activity	1. The computer will generate a password. 2. The levers will all be placed in a "neutral" position. 3. Students will drag the levers either "up" or "down" to "yes" or "no" 4. Once all levers have been placed in either the "up" or "down" position the "submit" lever will move to a ready position. 5. Students will then drag the "submit" lever down to submit the answers. 6. If correct, students will be given ten points, if the student is incorrect the reasons why they were incorrect will be displayed in red in the top right corner of the game. 7. If students are interested allow them to maintain a leaderboard of the scores that they receive in the game.
Timing Breakdown	1. The mission "Digital Citizenship" (20 minutes) 2. "Password Protectors" online activity (40 minutes)	1. The online mission "Digital Citizenship" will take approximately 20 minutes to navigate and complete. 2. "Password Protectors" on the Internet. (40 minutes)
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. We recommend that facilitators assess and approve the use of Google's free game "Be Internet Awesome" to supplement the program's digital citizenship mission. Facilitators can also take this opportunity to discuss online interactions that students may have had in the past asking the following questions: • Have students chatted online before? • Have they seen someone say something mean to another person online? • What, if anything did they say to that person to make them stop? And why? • Have THEY said anything mean to someone online before? Even a friend as a joke?	If facilitators choose to engage in the topic of prior experience with online chatting remember to keep an open mind. This is a learning opportunity for students to recognize appropriate online communication

	Facilitator Guide	Notes
Summary	In this mission, students will learn and demonstrate the following concepts: • Recognizing cyberbullying • Judging a strong password • Comparing rights and responsibilities in relation to digital citizenship • Recognize respectful communication online • Recognizing fake or malicious websites	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

- Mission breakdown

Look and listen Fors

- Learners may have problems at first recognizing that some of the levers are not in the "yes" position but are in a neutral position. If you hear that learners are frustrated because an answer will not submit, direct them to make sure that all of the other levers are in the locked position in either yes or no.

Blocks Required

- No blocks are required for this mission.

MISSION 1

Introduction To Digital Citizenship
Activities Facilitator Notes

1 LEVEL Password Protectors Game

The Password Protectors game gives learners practice in recognizing passwords that meet the established criteria for safe passwords.

1. The computer will generate a password.
2. The levers will all be placed in a “neutral” position.
3. Students will drag the levers either “up” or “down” to “yes” or “no”
4. Once all levers have been placed in either the “up” or “down” position the “submit” lever will move to a ready position.
5. Students will then drag the “submit” lever down to submit the answers.
6. If correct, students will be given ten points, if the student is incorrect the reasons why they were incorrect will be displayed in red in the top right corner of the game.
7. If students are interested allow them to maintain a leaderboard of the scores that they receive in the game.

1 LEVEL Password Builder Activity

The Password Builder activity gives additional practice for the steps to creating a secure password:

1. Pick two 5 letter words and put them together Change 1-2 letters to uppercase
2. Change at least one letter to a number or symbol (It’s even better if you can use both
3. A number and a symbol!)

Reinforce to the students that this is practice, and they do not have to use the passwords they are making now. Also remind them that as they practice, they need to make changes they think they would be able to remember. A password like #rT2!$sQ may be secure, but it is highly unlikely they will remember such a string.

Challenge

The Password Builder challenge allows the students to try to make a 12-character password using the same rules as above.

Cyberbully Song or Skit Performance

Break students into groups to allow them to create a song or a short skit that reinforces how to deal with cyberbullies:

STOP

Don't write anything to the bully. You may want to tell them that what they said wasn't nice or true, or you may be tempted to say mean things back to them. DON'T.

TELL

Tell an adult when you see a cyberbully. Show them the messages you've seen or been sent. Let them help you.

BLOCK

Once adults have seen the messages, block the cyberbully from having contact with you. If they are on your friends list, remove them. If they are texting you or a family member, ask for help in how to block their number.

MISSION 1

Introduction to Digital Citizenship
Additional Activities

1 LEVEL Interland Internet Game

Visit the website **https://beinternetawesome.withgoogle.com/interland** to play a game where you can practice the rules for being a good digital citizen.

1 LEVEL Password Builder Activity

Practice building two different passwords using our Password Building process:

1. Pick two 5 letter words and put them together
2. Change 1-2 letters to uppercase
3. Change at least one letter to a number or symbol (It's even better if you can use both a number and a symbol!)

It is important to make changes that you think you can still remember!

If you cannot think of good 5 letter words to use, here are some options:

kitty	bunny	fancy	paint	sorry	shoes	dance	faint	words	glass	chair	floor	dress
arrow	fizzy	apple	juice	crazy	movie	zebra	click	water	grass	south	squid	clock

Symbols

!	@	#	$	%	^	&	*	+	<
>	;	:	/	[	]	{	}	~	|

Password 1:

Step 1:_____

Step 2:_____

Step 3:_____

Password 2:

Step 1:_____

Step 2:_____

Step 3:_____

Challenge

Can you create an even more secure password?

Follow the Password Building process to make a 12-character password by using two 6 letter words put together.

Step 1:_____

Step 2:_____

Step 3:_____

Cyberbully Song or Skit Performance

With a group, create a song or skit to help your class remember the three steps to dealingwith cyberbullies:

STOP

Don't write anything to the bully. You may want to tell them that what they said wasn't nice or true, or you may be tempted to say mean things back to them. DON'T.

TELL

Tell an adult when you see a cyberbully. Show them the messages you've seen or been sent. Let them help you.

BLOCK

Once adults have seen the messages, block the cyberbully from having contact with you. If they are on your friends list, remove them. If they are texting you or a family member, ask for help in how to block their number.

robots4STEM®

| MISSION AT A GLANCE |

MISSION 2: Introduction to Algorithms and Sequences

[1 hours, 30 minutes]

Mission Overview	Algorithms are critical to the understanding of computer programming. In this mission, students will learn that algorithms are the step by step instructions to perform a task. They also explore the concept that the sequence (or order) of those steps matter.	
Mission Objectives	• Demonstrate algorithmic thinking by formulating a set of instructions for a problem • Create a sequence by placing events and actions in a specific order	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Algorithms. Abstraction Impacts of Computing Social Interactions. Human – Computer Interaction Social Interactions. System Relationships	
Connections within Framework	3–5. Data and Analysis. Collection K-2. Computing Systems. Troubleshooting	
Materials	Paper and pencil – students write steps Board and Drawing Utensil - facilitator to test Supplies for "How to Get Dressed" Roles and Responsibilities Reflection Tool Access to Online Content	
Activity (revisions may stop at two if successful)	• Introduce the "Draw a Square Activity" (Each student does independently) • Initial Design – students write out directions • Facilitator Tests first volunteer • Second Design – students revise directions • Facilitator Tests second volunteer • Third Design – students revise directions • Facilitator tests third volunteer • Facilitator facilitates reflection of activity with the group	45 minutes
Online Work	• Complete Mission • Complete quiz	15 minutes

Group Activity	• Introduce "How to Get Dressed" (Groups of 3-4) • Students Design • Students present (2 minutes each) • Each group debriefs using reflection and presents 1 reflection point	30 minutes

MISSION 2

Introduction to Algorithms and Sequences
MISSION PLAN

Subject	Grade	Mission length
Learn about breaking tasks down into steps and why the order of those steps matter.	3rd-5th	60 minutes

OVERVIEW

In this mission, students learn foundational concepts of algorithmic thinking and sequencing.

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following: • Listing the steps needed to complete a provided task; and • Recognizing the importance of the order of steps to completing the task. • Demonstrate an ordered list of steps to complete a daily task.	
Important vocabulary	1. Algorithm 2. Sequence	1. Algorithm- a group of instructions designed to perform a specific task 2. Sequence- the following of one thing after another in a specific order

	Facilitator Guide	Notes
Activity	1. The mission "Algorithms and Sequences" 2. Activity 1: How to Draw a Square 3. Activity 2: How to Get Dressed	1. The online mission "Algorithms and Sequences" will take approximately 25 minutes to navigate and complete. 2. How to Draw a Square • Students will be divided into groups of 2-4. • The facilitator should give each group a box of crayons and paper with the directive to write instructions to draw a square. • The instructions should be explicit and as thorough as possible. • After the allotted time gather each group's instructions and follow them to draw a square on the board or on a sheet of paper. • Take all instructions should be taken extremely literally. 3. How to Get Dressed Activity • The facilitator should give the students the directive to write down instructions to someone who has never dressed themselves before. • Begin with instructions that students will use a three-step algorithm. • Once they have completed the three- step algorithm, ask them to extend to five steps and finally to seven steps. • The extension question should be focused towards the following: "What if a robot was the one doing the dressing?"

	Facilitator Guide	Notes
Timing Breakdown	1. The mission "Algorithms and Sequences" 2. Activity 1: How to Draw a Square 3. Activity 2: How to Get Dressed	1. The mission "Algorithms and Sequences" will take approximately 25 minutes to navigate and complete. 2. Activity 1: How to Draw a Square • Handing out supplies (5 minutes). • Exercise (5-7 minutes) • Debrief (5 minutes) 3. Activity 2: How to Get Dressed • Exercise (10 minutes): Steps 3-5-7 • Debrief (10 minutes)
Facilitator's Role	Facilitators should expect to be highly involved in coordinating the activities provided with this mission. Facilitators will act as a medium for the instructions that the students give throughout these activities. Facilitators should take them as literally as possible to help students understand the level of detail needed to complete a task.	
Summary	In this mission, students learned and applied the foundations for algorithmic thinking that includes: • Listing the steps needed to complete a provided task; and • Recognizing the importance of the order of steps in completing the task.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the algorithms and sequences mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

- Mission breakdown
- Pen/Pencil/Crayons
- Paper
- Computer

Look and Listen Fors

- The activities that accompany this mission require that students find every detail and be explicit in their instruction. Try your best to follow their instructions literally without filling in steps.

Blocks Required

- No blocks are required for this mission.

MISSION 2

Introduction to Algorithms and Sequences
Additional Activities

GROUP

Draw a Square

Get into a group with 2 or 3 other students. The mission for your group is to make an algorithm to teach someone how to draw a square. The person has never drawn a square before, so what exact steps would they need to follow? Assume the person has a box of crayons and blank paper to use.

Take 5-7 minutes to write down your steps as a group. Now, let's test your algorithm! Give your algorithm to another group and have them follow your steps exactly as you wrote them. Was the other group able to successfully draw a square?

You will also receive an algorithm from another group. Follow their steps exactly as they wrote them to see what you end up drawing.

Discuss in your group: was it harder than you thought to give exact instructions? What went wrong when the other group tried to follow your steps?

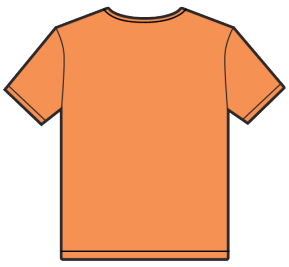
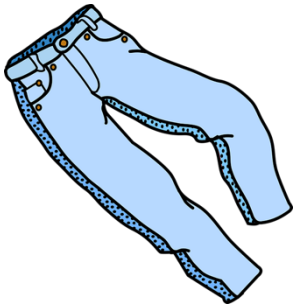
1 LEVEL

Getting Dressed Sequencing

For this activity, write your answers down on a sheet of paper.

Imagine you are teaching a younger sibling how to get dressed in the morning. They have 4 pieces of clothing to put on:



Can you provide a 4-step algorithm for the order for them to put on their clothing?

1. _____

2. _____

3. _____

4. _____

Your little sibling looks at you and says, "No! I don't want to do it that way!" Can you provide two more algorithms for them for how they can get dressed?

Alternative 1:

1. _____

2. _____

3. _____

4. _____

Alternative 2:

1. _____

2. _____

3. _____

4. _____

Your sibling is not following your directions. You notice they are putting on their shoes and then trying to put on their pants. "That won't work!" you tell them. While there are several good ways to put on the 4 pieces of clothing, you discover one rule: the shoes cannot be put on until the pants are on. Can you think of another sequencing rule for the order of clothing?

Sometimes in programming – and in life! – we must have a very strict set of instructions where the order of every instruction matters. Other times, we have a choice of the order we do instructions, as long as all of those instructions happen before a certain event.

Challenge

There are 8 algorithms that will help your sibling get dressed correctly. You have already discovered 3. Can you find the other 5?

Mission 2

Introduction to Algorithms and Sequences
Activities - Facilitator Notes

GROUP

Draw a Square

Break students into small groups of 3-4. Each group starts with a closed box of crayons, a pencil, and blank paper. Ask them (as a group) to write the steps of a specific algorithm for someone to draw a square with a crayon. The level of detail should assume that these instructions are for someone who has never drawn a square before.

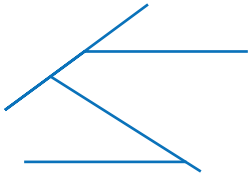
Students should work together to have only one algorithm by the entire group. Give them 5-7 minutes to do this.

Then, have each group pass their written algorithm to another group. Each group should try to follow the algorithm they were just given step by step and see if the result is a square, or various lines.

Some common inexact instructions you may see are:

- Get a crayon (from where?)
- Open the box of crayons (someone will open it from the bottom and crayons spill out)
- Draw 4 lines on the paper (Like this?)

- Draw 4 lines on the paper that touch (Like this?)



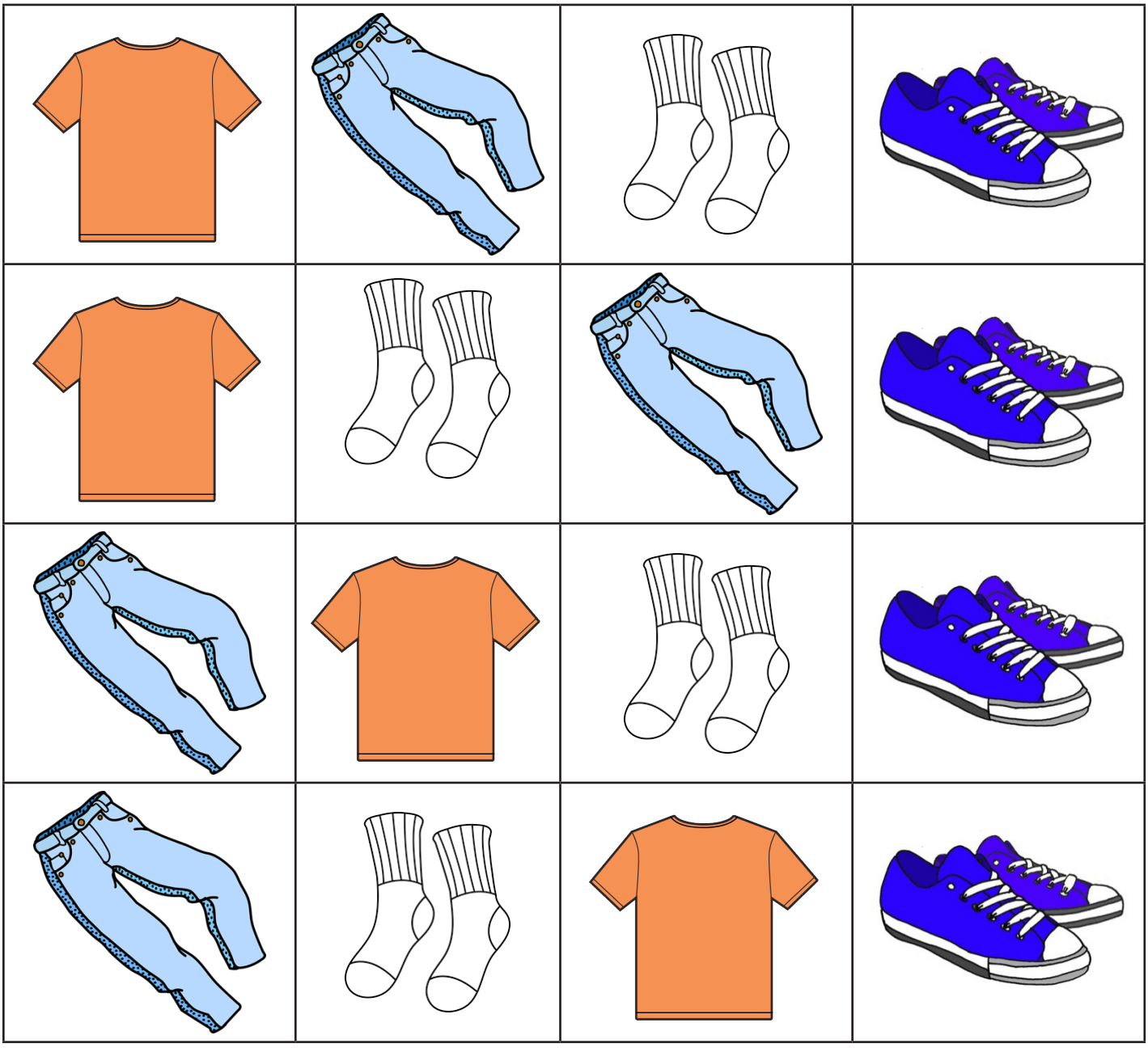
If you have time after the groups have a chance to execute their given algorithm, ask the class to raise their hand if they think they received an excellent one. Then, you can try to execute the steps in front of the class, being as literal as you can. One scenario the students will likely not remember to cover is the white crayon. If the algorithm has you choose any color crayon, choose white. The class will groan because now the lines will not show up on the paper when you draw. It is a fun demonstration to get them thinking about the many possible user scenarios.

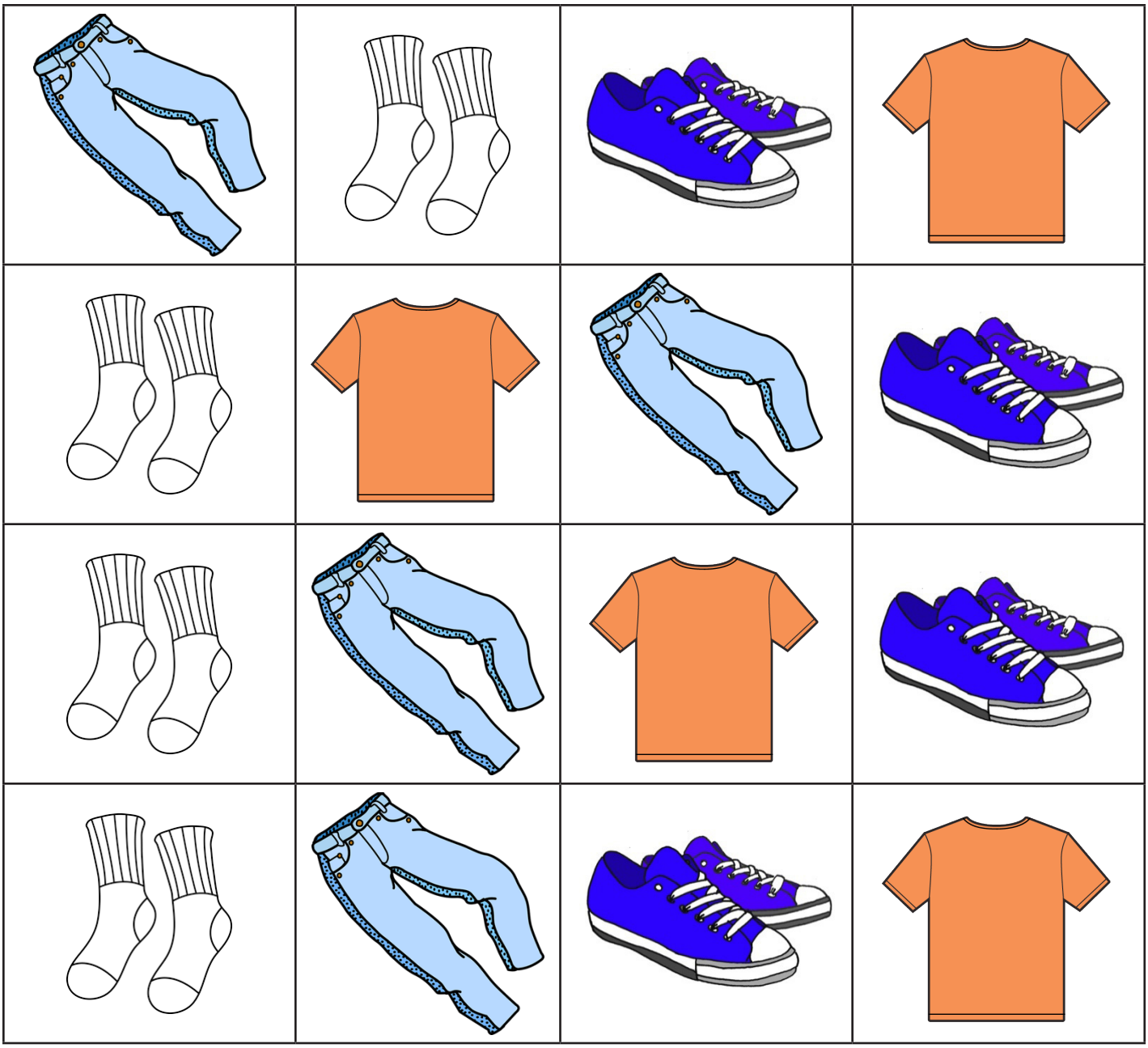
1 LEVEL Getting Dressed Sequencing - Answers

The two Getting Dressed rules are:

1. Pants must go on before Shoes
2. Socks must go on before Shoes

The 8 options for the Getting Dressed Sequencing algorithm are:





robots4STEM®

| MISSION AT A GLANCE |

MISSION 3: Introduction to Designing Computer Programs

[45 Minutes]

Mission Overview	Design is a critical step in developing programs — it is the creation of a plan for the program, much as a blueprint is the plan for a house. Implementing a program without design leads to significant time in troubleshooting and reworking the program. In this mission, students will learn the four steps in the programming process, the importance of design, and how to use flowcharts as a visual design model for algorithms.	
Mission Objectives	• Sequence the four steps of the programming process and understand why design comes before building a computer program • Recognize the differences between an algorithm and a computer program • Map an algorithm to a flowchart • Arrange flowchart pieces into the correct sequence	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Algorithms. Abstraction Program Development. Human – Computer Interaction Program Development. System Relationships	
Connections within Framework	3–5. Data and Analysis. Collection K-2. Computing Systems. Troubleshooting	
Materials	Paper and pencil – students design flow charts Board and Drawing Utensil - facilitator to illustrate flow chart Roles and Responsibilities Reflection Tool Access to Online Content	
Online Work	Introduce Mission • Complete Mission • Complete Quiz	25 minutes

Flow Chart Activity	Introduce Flow Charts • Facilitator demonstrate steps from "How to Get Dressed" ↳ Facilitator draws the framework • Students Choose a topic to make a flow chart independently and submit product at the end of class • Students share flow charts (Volunteers) • Each student completes reflection and submits • Facilitator facilitates debrief as a class with reflection tool	20 minutes

Mission 3

Introduction to Designing Computer Programs
Mission Plan

Subject	Grade	Mission length
Learn how flowcharts can help in designing computer programs.	3rd-5th	45 minutes

Overview

In this mission, students learn the foundational concepts necessary to develop a visual map of algorithms and sequences.

	Facilitator Guide	Notes
Objectives	In this lesson, students learn and demonstrate the following concepts: • Identifying a step by step algorithm of a daily task; and • Organizing these steps in a way that will complete the provided task; and • Constructing a visual map of the steps that the student has identified.	

	Facilitator Guide	Notes
Important Vocabulary	1. Computer Program 2. Connector 3. Design 4. Flowchart 5. Process 6. Terminator	1. Computer Program - A sequence of instructions that a computer can understand and execute. 2. Connector- The arrows between the symbols in a flowchart that show which path to follow. 3. Design- To prepare the sketch or plans for a work to be executed. 4. Flowchart- A diagram that is a pictorial representation of an algorithm. 5. Process- A rectangular flowchart symbol representing an action statement in the algorithm. 6. Terminator – An oblong oval flowchart symbol that shows where the algorithm execution begins or ends.
Activity	I. The mission "Learning How to Design Computer Programs" II. Activity 1: How to Get Dressed (Flowchart) III. Activity 2: Jett Sprite Tells a Joke IV. Activity 3: Jett Sprite Gets a Present V. Activity 4: Jett Sprite Helps a Friend	Things that learners need to submit: • 4 Flowcharts

	Facilitator Guide	Notes
Timing Breakdown	I. The mission "Learning How to Design Computer Programs" II. Assignment 1: How to Get Dressed III. Activity 2: Jett Sprite Tells a Joke IV. Activity 3: Jett Sprite Gets a Present V. Activity 4: Jett Sprite Helps a Friend 1 Flowchart	I. The mission "Learning How to Design Computer Programs" will take 25 minutes to navigate and complete. II. Assignment 1: How to Get Dressed (beginner): (10 minutes) • Hand out supplies (5 minutes) • Give students materials to draw a flowchart for "getting dressed". • Remind students about the previous exercise of writing out the steps. III. Assignment 2: Jett Sprite Tells a Joke (20 Minutes) • Create a flowchart for the "Jett Sprite Tells a Joke" activity. Jett will tell a joke to another sprite. • At least one additional costume for Jett. • Change the stage background. • Learn how to save project assignment(s). IV. Assignment 3: Jett Sprite Gets a Present (intermediate) (25 minutes) • Create a flowchart for the "Jett Sprite gives a present" activity. (10 minutes) • 1 additional sprite (present –main costume). • What's inside? (Second costume) - an emote costume for Jett. • New background if desired. • Learn how to add additional sprites. • Learn how to make new projects. V. Assignment 4: Jett Sprite Helps a Friend 1 (advanced) (35 minutes) • Create a flowchart for the "Jett Sprite helps a friend" activity. (10 minutes) • Need Jett plus at least 1 more sprite (Jett's friend) · • Both Jett and friend must have multiple costumes. • Required background change. • Additional sprites may be needed, depending upon the student's flowchart.

	Facilitator Guide	Notes
Timing Breakdown (cont.)		• Both Jett and friend must have multiple costumes. • Required background change. • Additional sprites may be needed, depending upon the student's flowchart.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners have four activities that are mandatory for progression. These are online activities that do not require facilitator intervention. Activities 2-4 sets the framework for the "Workspace 2" and "Jett Avatar" missions, students will need to be confident importing and exporting programs by the end of this mission. In addition, learners should provide a flowchart which will be required to check the program against.	There will be a digital copy of the flowchart images in addition to the hard copy that is given to facilitators **(see page 132)**. This is an opportunity for facilitators to use a word processing program or presentation program for students to edit the document rather than/ in addition to drawing with pen(cil) and paper.
Summary	In this mission, students learned and practiced demonstrating the following: • Interpreting which task needs to be accomplished; • Organizing these steps in a way that will complete the provided task; and • Designing a visual map of the steps that the student has identified.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

- Mission breakdown
- Pen/Pencil
- Paper
- Computer
- Flowchart pieces (starting on page 132)

Look and Listen Fors

- This is the first time that the program is asking students to complete this task and there are higher level concepts, be patient and encourage students to be as explicit as possible with their flowcharts. It may also be helpful to print the basic flowchart shapes and the daily tasks flowchart as an example for students.
- Students will need to submit a hard copy or digital copy of their flowchart to compare their finished product to.

MISSION 3

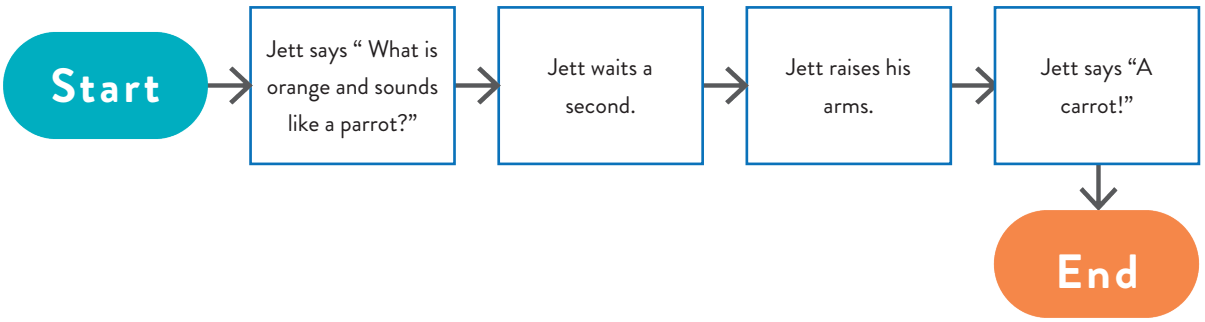
Introduction to Designing Computer Programs
ADDITIONAL ACTIVITIES

1 LEVEL Jett Tells a Joke Flowchart

For this activity, we will draw a flowchart for a program that would have Jett tell a joke. A joke is a short question followed by a funny answer. An example might be:

Question: What is orange and sounds like a parrot?
Answer: A carrot!

A flowchart for a program that tells that joke might look like this:



On a clean sheet of paper, draw your own flowchart for a new joke.

If you do not know a good joke, here are some suggestions:

Q: What is the tallest building in the world?
A: The library, because it has so many stories!

Q: What do you call a sleeping bull?
A: A bulldozer!

Q: Why can't a nose be 12 inches long?
A: Because then it would be a foot!

Q: How did the farmer round up his 27 sheep?
A: 30

Q: What did the 0 say to the 8?
A: Nice belt!

Jett Gets a Present Scene Setup

For this activity, draw a flowchart for a program where Jett gets a present. You get to decide what is in the present and how Jett feels about what is inside. Add something funny to the story.

Jett Helps a Friend Scene Setup

For this activity, draw a flowchart for a program where Jett helps a friend. You get to decide who Jett's friend is and what problem Jett solves. This story requires both the Jett and the friend to talk.

MISSION 3

Introduction to Designing Computer Programs
Activities Facilitator Notes

1 LEVEL Jett Tells a Joke Flowchart

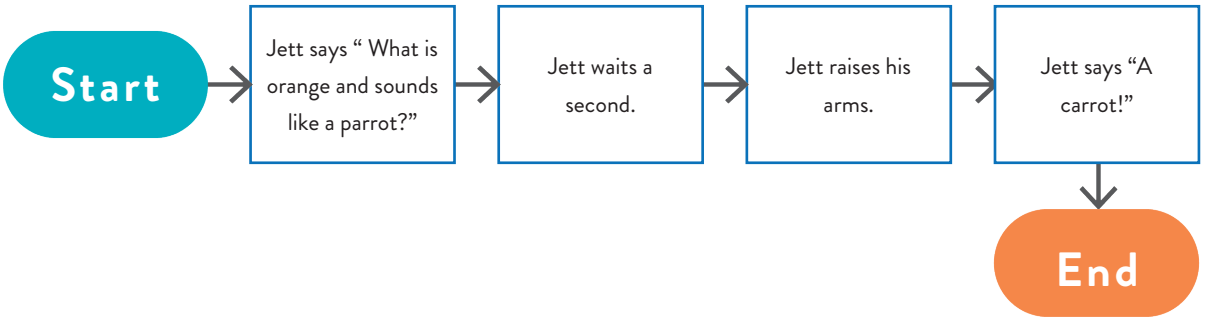
For this activity, students will draw a flowchart for a program that would have Jett tell a joke. A joke is a short question followed by a funny answer. The example given to the students was:

Question: What is orange and sounds like a parrot?
Answer: A carrot!

Students should draw their own flowchart on a blank sheet of paper.

Ensure that the students have a Start terminator, at least 3 Process symbols, and an End terminator, all with connectors. They may not have any animations for Jett, but they must at least have the joke question, wait for response, and then punchline.

A flowchart for a program that tells that joke might look like the following:

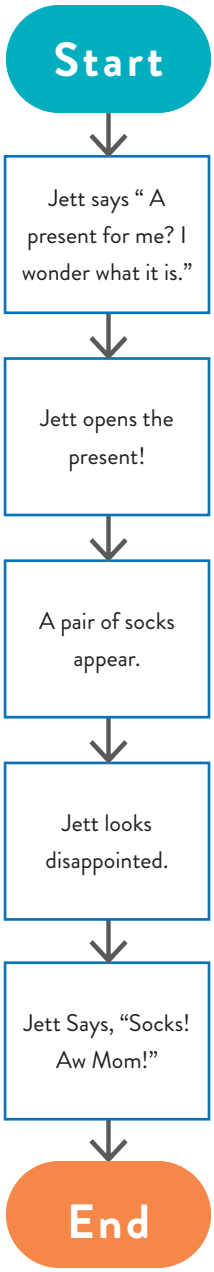


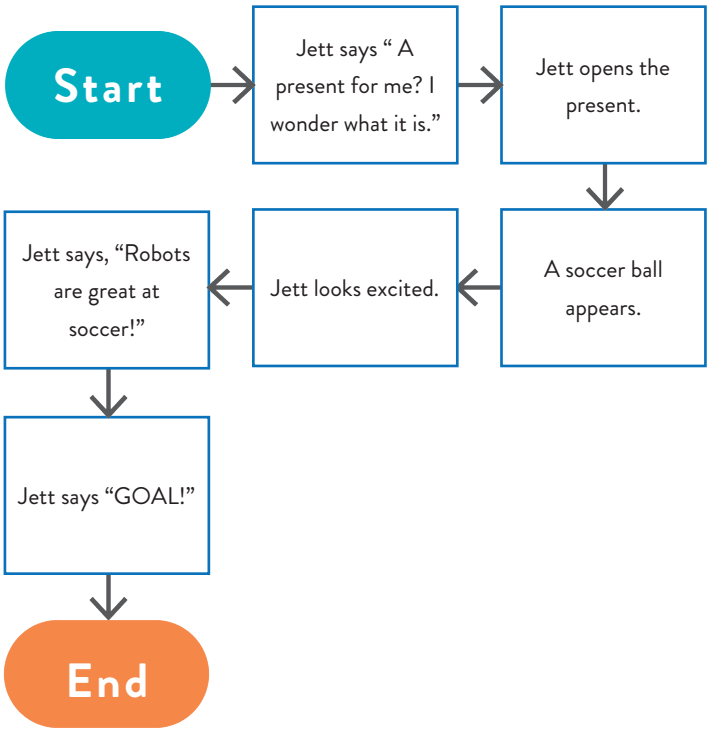
Jett Gets a Present Flowchart

For this activity, students draw a flowchart for a program where Jett gets a present. They get to decide what is in the present and how Jett feels about what is inside. They were required to add something funny to the story.

This is an open-ended assignment, therefore there are no set requirements for flowchart size. You should check their flowcharts to ensure that they do not place two distinct actions inside of one process block.

Here are two examples of flowcharts for this storyline:





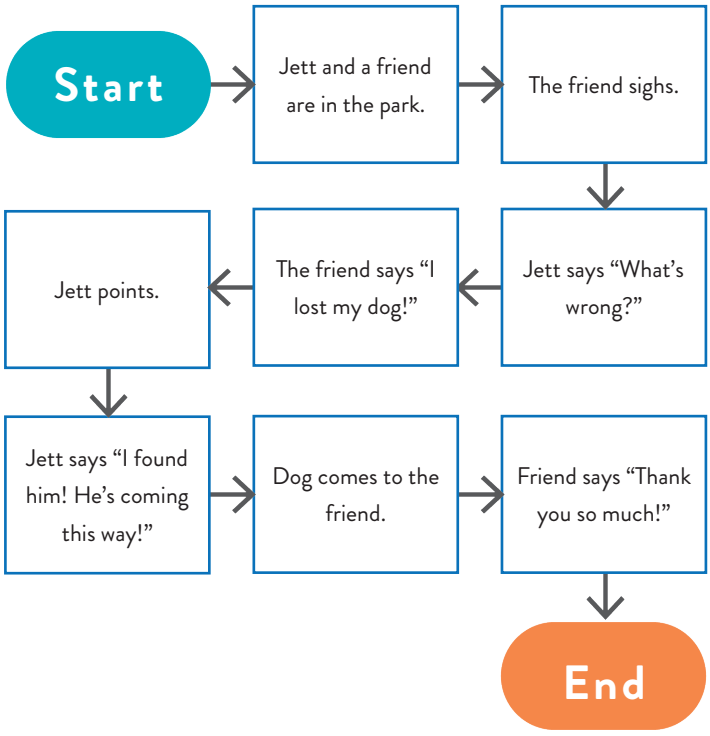
Jett Helps a Friend Flowchart

For this activity, students draw a flowchart for a program where Jett helps a friend. They get to decide who Jett's friend is and what problem Jett solves. This story requires both the Jett and the friend to talk.

This flowchart's difficulty is in correctly sequencing the action occurring between the two (or more) characters.

Note: When we reach the missions for programming, this story would require parallel scripts (a separate script for each character).

Therefore, the flowchart for correctly coding the story would require a separate flow for each character and would not be serial, but the students would not know or understand that concept yet. The example provided is a serial version, much like a student would write.



robots4STEM®

| MISSION AT A GLANCE |

MISSION 4: Introduction to the Workspace (Part 1)

[1 hours, 30 minutes]

Mission Overview	The robots4STEM® Workspace is the blocks based programming environment where students will develop programs throughout the rest of the course. In this mission, students are introduced to the interface of the Workspace and learn the basic fundamentals of scene setup: sprites, costumes, and backgrounds.	
Mission Objectives	• Become familiar with the interface of the Workspace environment • Demonstrate scene setup knowledge through the use of sprites, costumes, and stages	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Variables. Abstraction Program Development. System Relationships	
Connections within Framework	3–5. Data and Analysis. Storage K-2. Computing Systems. Troubleshooting	
Materials	Reflection Tool Access to Online Content Blank Workspace Diagram Writing Utensil for students	
Online Work	• Facilitator introduces mission and how to submit work from the workspace • Students complete mission	15 minutes
Vocabulary Activity	• Facilitator introduces the Workspace Layout • Students individually label their workspace diagram • Facilitator debriefs labels with students	10 minutes
Individual Workspace Activity	• Student accesses the workspace and completes the activity • Student submits the workspace activity to the facilitator • Facilitator asks for volunteers to share their workspace project • Each student completes a reflection for their activity and submit • Class debriefs with facilitator using reflection tool	20 minutes

Follow-up Workspace Activity	• Students Complete the next 2 activities independently ↳ Tell a Joke ↳ Gets a Present ↳ Help a Friend • Students submit projects • Volunteers present projects • Each student completes reflection tool • Facilitator debriefs with students using the reflection tool	45 minutes

MISSION 4

Introduction To The Workspace (Part 1)
MISSION PLAN

Subject	Grade	Mission Length
Learn to navigate the Workspace, customizing the avatar, and using block programming.	3rd-5th	90 minutes

OVERVIEW

In this mission, students learn the foundational concepts of using "The Workspace", RoboKind's visual block programming space and basic programming of the robot's avatar.

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following: • Starting a new project file in the Workspace. • Saving a project as an .sb3 file. • Customizing the avatar. • Demonstrating ability to use blocks to build a small program.	

	Facilitator Guide	Notes
Important Vocabulary	1. Avatar 2. Block 3. Blocks Area 4. Jett Actions Palette 5. Palette 6. Script 7. Scripting Area	1. Avatar- A computer image or program that represents something in real life. 2. Block- One specific instruction located within a palette in The Workspace. Blocks fit together like puzzle pieces to create computer programs. 3. Blocks Area- The section in The Workspace that lists the blocks available to create programs. 4. Jett Actions Palette- A collection in The Workspace of all blocks related to actions the avatar and robot can perform. 5. Palette- A collection in The Workspace of similar blocks. Each palette has a color, and all blocks inside of that palette share the palette color. 6. Script- Another term for a computer program. 7. Scripting Area- The section in The Workspace where programs are built.
Activity	I. The mission "Introduction to The Workspace (Part 1)"	Things that learners need to submit: • A flowchart • .sb3 File
Timing Breakdown	I. The mission "Introduction to The Workspace (Part 1)" II. Activity 1: Personality is Key	I. The mission "Introduction to The Workspace" will take 25 minutes to navigate and complete. II. Activity 1 (beginner) (15 minutes): • After customizing the avatar to the look that the student likes only half of the work is done! • Students should begin deciding what the personality of their avatar will be. Students should be provided with the basic personality chart (provided in the curriculum guide) • Students should provide at minimum a program that includes one speech block and one emote block.

	Facilitator Guide	Notes
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners have one activity that is mandatory for progression. This is an online activity that does not require facilitator intervention. For the learner to progress to the next mission, the facilitator must approve that a student's created script runs as intended and meets the requirements as outlined in the mission plan. In addition, learners should provide a flowchart which will be required to check the program against.	Facilitators should also decide and communicate to students whether they would like a hard copy of required flowcharts, digital copies or both.
Summary	In this mission, students learned and demonstrated the following: • Starting a new project file in the Workspace. • Saving a project as an .sb3 file. • Customizing the avatar. • Demonstrating ability to use blocks to build a small program.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission. Basic competency for this mission also includes saving a file to import in a later mission.

Requirements

- Computer

Resources

- Mission breakdown
- Pen/Pencil
- Paper
- Computer

Look and Listen Fors

- Students may have challenges with using the Workspace and missions at the same time. It may be helpful for facilitators to instruct students to open up a separate window for the Workspace.

Blocks Required

- No blocks are required for this mission as learners are manually changing the costumes.

MISSION 4

Introduction to The Workspace (Part 1)
Activities Facilitator Notes

Jett Tells a Joke Scene Setup

Set up the scene for a program where Jett would tell a joke to the user.

The requirements for this activity are:

- Add at least one additional costume to the Jett sprite.
  Think about how he might pose differently while he is telling his joke. Jett costume names start with the word Jett. Some examples are: Jett arms raised, Jett point left, or Jett wave right.
- Change the stage background.
  Where should Jett be while he is telling his joke?

When you are finished testing your program, export your project. Name it *YourName_JettJokeSetup.sb3* (replace YourName with your actual name) and then submit it to your facilitator.

Jett Gets a Present Scene Setup

Set up the scene for a story program where Jett gets a present. You get to decide what is in the present and how Jett feels about it.

The requirements for this activity are:

- Add an emotion costume to the Jett sprite for how Jett feels about the present.
  Some example costume names are: Jett Happy, Jett Sad, Jett Angry, or Jett Surprised.
- Add an additional sprite for the present.
  For the sprite to appear as a wrapped present and not another copy of Jett, you must delete the Jett costume from the second sprite and add a costume for the wrapped present. Example costumes could be Gift a or Gift b.
- Add a costume to the second (gift) sprite for what the present contains. What will be in the present?
  Explore the many costume options and add your choice as a costume for the second sprite.

When you are finished, export your project. Name it *YourName_JettPresentSetup.sb3* (replace YourName with your actual name) and then submit it to your facilitator.

Jett Helps a Friend Scene Setup

Note: If you did not complete a flowchart for this program in the Additional Activities of Mission 3, you are strongly encouraged to build a flowchart before you begin this activity.

Set up the scene for a story program where Jett helps a friend. You get to decide who Jett's friend is and what problem Jett solves.

The requirements for this activity are:

- Add a sprite for Jett's friend.
  Additional sprites may be needed depending upon your story.
- Both Jett and his friend must have at least one additional costume.
  Carefully consider what the sprites will be doing in the story. More than one costume for each sprite may be necessary.
- Change the stage background.
  Where will this story take place?

When you are finished testing your program, export your project. Name it Y*ourName_JettFriendSetup.sb3* (replace YourName with your actual name) and then submit it to your facilitator.

robots4STEM®

| MISSION AT A GLANCE |

MISSION 5: Introduction to the Workspace (Part 2)

[2 Hours]

Mission Overview	The robots4STEM® Workspace is the blocks-based programming environment where students will develop programs throughout the rest of the course. In this mission, students are introduced to additional portions of the interface of the Workspace and learn the purpose for several blocks: When green flag clicked, wait, say for n secs, next costume, and switch to costume. Students also learn the concept of parallelism in code – the practice of running more than one script at the same time.	
Mission Objectives	• Recognize block and script areas of the Workspace • Learn and utilize newly introduced programming blocks • Understand how the shape of a block determines where in a script it can be used • Exhibit a basic level of understanding of the concept of parallelism in a program • Demonstrate logical thinking and sequencing to develop a working computer program	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Variables. Abstraction Program Development. Human-Computer Interaction Program Development. Systems Relationships	
Connections within Framework	3–5. Data and Analysis. Storage K-2. Computing Systems. Troubleshooting	
Materials	Reflection Tool Access to Online Content Workspace Diagram from last mission for each student Paper and Pencil for student to submit flowchart	
Online Work	• Facilitator introduces mission and reviews how to submit work from the workspace • Students complete mission	15 minutes
Vocabulary Activity	• Students individually add labels their workspace • Facilitator debriefs labels with students	10 minutes

Individual Workspace Activity	• Facilitator introduces new expectations for all online submissions ↳ Flowchart ↳ .sb3 file ↳ Reflection tool • Facilitator discusses how to import work • Students access the workspaces and completes the activity • Student submits the workspace activity to the facilitator • Facilitator asks for volunteers to share their workspace project • Each student completes a reflection for their activity and submit • Facilitator models debrief with examples of struggle and failure	20 minutes
Follow-up Workspace Activity	• Students Complete the next 3 activities independently ↳ Tell a Joke ↳ Gets a Present ↳ Helps a Friend • Students submit projects • Volunteers present projects (1 for each activity) • Each student completes reflection tool • Facilitator debriefs with students using the reflection tool • Students reflect independently after the model	75 minutes

MISSION 5

Introduction To The Workspace (Part 2)
MISSION PLAN

Subject	Grade	Mission length
Learn to use blocks to program the avatar to talk and animate.	3rd-5th	120 minutes

OVERVIEW

In this mission, students learn the foundational concepts of visual block programming and using different "blocks" for specific purposes.

	Facilitator Guide	Notes
Objectives	In this mission, students will learn and demonstrate the following concepts: • Opening a previously saved project in the Workspace. • Displaying a flowchart depicting a simple programming task. • Constructing a basic script to address a stated task. • Creating a program to complete a simple task. • Saving changes made to a program. • Programming basic tasks such as move and turn with a provided medium.	

	Facilitator Guide	Notes
Important vocabulary	1. Hat Block 2. Parallel	1. Hat Block- Blocks in The Workspace that can only be placed at the top of a script and specify the event that must occur for the script to be launched. 2. Parallel- Side by side. In programming, use to note two programs running at the same time.
Activity	1. Introduction to Using the Workspace 2 Mission 2. Vocabulary Activity 3. Activity 1: Jett Avatar tells a Joke 4. Activity 2: Jett Avatar Does a Dance	Learners must submit at least: • Flowcharts (3) • .sb3 Files (3) • Reflection Documents (3)

	Facilitator Guide	Notes
Timing Breakdown	I. I. The mission "Introduction To The Workspace (Part 2)" (15 minutes) II. Activity 1: Jett Avatar Tells a Joke (20 minutes) III. Activity 2: Jett Avatar Does a Dance (25 minutes)	I. The Mission "Introduction to Using the Workspace 2" (15 minutes) II. Activity II: Jett Avatar Tells a Joke (20 minutes) • Students will open the personality is key file that they saved in the previous class. • Students will then program Jett to tell a joke of their choosing. • Students will then save a copy of their updated file following the convention: *StudentName_JettTellsAJoke.sb3* **Requirements:** • Jett must say the joke question, wait for a small amount of time, then tell the joke punchline. • Jett must use at least two actions. • Program must use these blocks in any order that allows programs to function properly: • Say for (n) seconds • Wait • When Green Flag Clicked **Complete assignment by fulfilling requirements.** • Learn how to open project • Learn how to save project with new name III. Assignment 2: Jett Avatar Does a Dance (25 Minutes) (basic) • Students will start a new project. • Students will program Jett to do a series of movements to make Jett move without using the "dance" blocks.

	Facilitator Guide	Notes
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners have one activity that is mandatory for progression. This is an online activity that does not require facilitator intervention. For the student to progress to the next mission, the facilitator must approve that a student's created script runs as intended and meets the requirements as outlined in the mission plan. In addition, learners should provide a flowchart which will be required to check the program against.	Learners are required to do two activities one which requires them to import the .sb3 file that they submitted in the last session. Learners may have imported files and folders before but may need guidance on how to do so within the LMS and The Workspace. Facilitators should be available to answer any of those questions. In addition, facilitators should accept, and grade submitted files for learners to progress against the flowchart that students have submitted. Remind learners that they must add to the "Personality is Key" program with their summary and conclusion of the day's lesson.
Summary		In this mission, students learned and demonstrated how to: • Opening a previously saved project in The Workspace. • Displaying a flowchart depicting a simple programming task. • Constructing a basic script to address a stated task. • Creating a program to complete a simple task. • Saving changes made to a program. • Programming basic tasks such as move and turn with a provided medium.

	Facilitator Guide	Notes
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission. Basic competency for this mission requires opening a previously saved file, completing the project, saving the file, then uploading the file to the LMS.

Requirements

- Computer
- Workspace Cut-out
- Flowchart

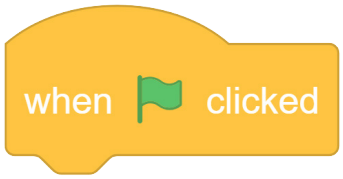
Resources

- Mission breakdown
- Pen/Pencil
- Paper
- Flowchart pieces

Look and listen Fors

Blocks Required

Events



Jett Actions

Any block(s) from this palette may be used.

MISSION 5

Introduction To The Workspace (Part 2)
Activities Facilitator Notes

Students are strongly encouraged to create flowcharts or algorithms for all programs. Activities 1, 3, and 4 are coding the scenarios presented in Missions 3 and 4, so they may already have flowcharts completed. Activity 2 is a new scenario but is an extension of Activity 1.

Mission Assignments

1 LEVEL Jett Speaks

This assignment requires the students to use the **When Green Flag Clicked** and **Say for n secs** blocks to produce speech bubbles for the sprite. They may have the sprite say anything they wish but must not leave the default text of "Hello!" in the **Say for n secs** block.

This is the first time that students will build a working program in the Workspace. Common student assistance needs may include:

- Remembering to click the green flag above the stage to start the program
- Remembering to use the **Say for n secs** block instead of the **Say** block (both produce speech bubbles, but the **Say** block keeps the speech bubble on the screen until the sprite next speaks, which can lead to odd looks in later programs)
- Exporting the project

An example program named *Mission5_JettSpeaks.sb3* is provided.

2 LEVEL Double Talk

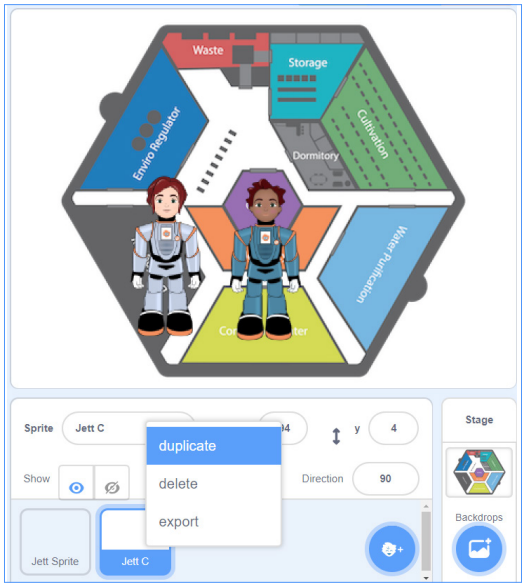
This assignment requires the students to demonstrate parallelism by demonstrating a conversation between two sprites. Students may recreate the conversation shown in the mission, or they may choose to create their own. The **When Green Flag Clicked**, **Say for n secs**, and **Wait** blocks will be used for this program. Common student assistance needs may include:

Adding a second sprite.

- To add a new sprite, left-click the "Choose a Sprite" icon. Then, select a Sprite.

Choose a Sprite button.

- To duplicate an existing Sprite, right-click on the sprite in the Sprite Corral and choose "duplicate".



Ensuring that the scripts for each sprite are in their respective script areas.

- Students may put both scripts into one sprite, which will not work. Each sprite will have one script in their own script area tab. Show the students how to switch between sprites by clicking each one in the Sprite Corral.

Conversation timing issues.

- The waits or pauses between speech bubbles is important in mimicking conversation. Ensure the students have **Wait** blocks at the appropriate places for each sprite, and that the seconds are long enough to wait for the speech of the other sprite plus a one second pause.

An example program named ***Mission5_JettSpeaks.sb3*** is provided .

Additional Activities

Jett Tells a Joke Script

In this activity, students create the program where Jett would tell them (the user) a joke.

The requirements for this activity are:

- Jett must say the joke question, wait for a small amount of time, then tell the joke punchline.
- Jett must switch costumes at least once. Use either of the costume blocks we covered to change his costume.
- Program must use the blocks
- Say for n seconds
- Wait
- When Green Flag Clicked
- Either Switch to Costume OR Next Costume

Students were asked to export the project to a file named TheirName_JettJoke.

An example program named ***Mission5_JettJoke.sb3*** is provided.

Jett Joke with 2 Sprites Script

In this activity, students create a program where Jett tells his same joke from Activity 1, but he tells it to another sprite on the stage. The flow looks like this:

Jett: tells joke question

Other sprite: "I don't know! What?"

Jett: tells joke punchline

The requirements for this activity are:

- Add a second sprite for Jett to tell his joke to
- Use parallelism. The code for both Jett and the second sprite will start with the When Green Flag Clicked block
- No costume changes are needed for the second sprite
- The same blocks from Activity 1 will be necessary for this activity

Students were asked to export the project to a file named TheirName_JettJoke2Sprites.

An example program named ***Mission5_JettJoke2Sprites.sb3*** is provided.

Jett Gets a Present Script

In this activity, students create the program for a story where Jett gets a present. They get to decide what is in the present and how Jett feels about it.

The requirements for this activity are:

- Jett must change costumes at least 2 times
- Use parallelism. The code for both Jett and the gift will start with the When Green Flag Clicked block
- The costume for the gift should change to be what is inside the gift after Jett talks about receiving the present
- Ensure that one costume change for Jett is one of his emote costumes (happy, sad, excited)
- A background change may be desired but is not required.

Students were asked to export the project to a file named TheirName_JettPresent

An example program named ***Mission5_JettPresent.sb3*** is provided.

Jett Helps a Friend Script

In this activity, students create the program for a story where Jett helps a friend. They decide who the friend is and what problem Jett needs to solve.

The requirements for this activity are:

- Need Jett plus at least 1 additional sprite (Jett's friend)
- Each sprite must speak at least twice (like a conversation)
- Both Jett and friend must have multiple costumes
- Required background change
- Additional sprites may be needed depending upon the story, and those additional sprites will also run code parallel to the main two sprites

Students were asked to export the project to a file named TheirName_JettFriend.

An example program named ***Mission5_JettFriend.sb3*** is provided.

robots4STEM

| MISSION AT A GLANCE |

MISSION 6: Introduction To Working In The Simulation Room

[3 hours, 25 minutes]

Mission Overview	From the beginning of the course, students have been eager to get the opportunity to work with the real Jett Robot. In this mission, students will learn about the Jett Avatar, who is the digital representation of the Robot. They will be introduced to the Workspace blocks that control the Avatar. All programs written for the Avatar will also run on the Robot. Students will explore the many blocks available for Avatar activities and will create sequence-based programs that connect the Workspace to the Avatar for testing.	
Mission Objectives	• Understand the differences and similarities between the Avatar and the Robot • Experiment with the capabilities of the Jett Avatar • Develop a sequence of steps to create and test programmatic solutions suitable for both the Avatar and the Robot	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Variables. Abstraction Program Development. Human-Computer Interaction Program Development. Systems Relationships Impacts of Computing Social Interactions. Human-Computer Interaction Social Interactions. Systems Relationships	
Connections within Framework	3–5. Data and Analysis. Storage K-2. Computing Systems. Troubleshooting	
Materials	Reflection Tool Access to Online Content Roles and Responsibilities for Group Work Rubric for Avatar Project	
Introduction	• Facilitator introduces mission • Students open all the tools • Facilitator facilitates discussion of the concept of an avatar	10 minutes
Online Work	• Students Complete Missions • Students Complete Quiz	20 minutes

Online Activity	• Complete 1st Project • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	25 minutes
Online Bonus Activity	Bonus Project	20 minutes
Group Project (3-4 per group)	• Design Project • Test Project • Plan presentation of Project • Reflect on Project as a Group • Submit ↳ Flowchart Plan ↳ .sb3 File ↳ Group Reflection	60 minutes
Group Project Presentations	• Students Present ↳ Tell a Joke ↳ Give a Present • Students submit projects • Volunteers present projects • Each student completes reflection tool • Facilitator debriefs with students using the reflection tool	45 minutes

MISSION 6

Introduction To Working In the Simulation Room
MISSION PLAN

Subject	Grade	Mission length
Learn to program the avatar to talk, dance, and move around a virtual environment.	3rd-5th	30 minutes

OVERVIEW

In this mission, students learn the foundational concepts necessary to program the avatar, robot and digital space which they can customize.

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following concepts: • Recognizing the difference(s) between an Avatar and Robot. • Demonstrating fluid transition in programming multiple mediums. • Creating a visual representation of a program addressing a basic task. • Placing objects in a digital space. • Demonstrating how to program a script for the Avatar/Robot.	
Important Vocabulary	1. Jett Actions Palette	2. Jett Actions Palette- A collection in the Workspace of all blocks related to actions the avatar and robot can perform.

	Facilitator Guide	Notes
Activity	I. "Introduction to Working with the Avatar" Mission II. Activity 1: Take a Hike, Jett	Things that learners need to submit: • Flowcharts (4) : Camera Angles, Showing Coordinates, Entering the Simulation Room. • .sb3 files (4) • Reflection Statements (4)
Timing Breakdown	I. The mission "Introduction to Working with Jett's Avatar" (30 minutes) II. Activity 1: Take a Hike, Jett	I. The mission "Introduction to Working with Jett's Avatar" will take approximately 30 minutes. II. Activity 1 is an ongoing activity throughout the lesson. Facilitators should let students know that wherever they are required to program they are expected to write and submit a flowchart. Students will be instructed to pause and practice. At the end of the "Pause and practice" activity students will submit their sb3 file to the facilitator.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners have three activities that are mandatory for progression. There are several documents that are necessary for learners to submit. For the individual projects learners should turn in the documents that they are accustomed to turning in (e.g. a flowchart, reflection document etc.) for each personal project. For the group project they should expect to turn in their reflection document for their groups roles and a flowchart that should match the rest of the group's.	• Students will submit a flowchart which outlines the intent of the program, the facilitator will ensure that the program performs the task that the student outlines. • Remind students that when they are instructed to "Pause and Practice" they must save their program with a new name each time. • Additionally, students will be expected to contribute to their course long project and add it to the program that was started in the mission "Personality is Key".

	Facilitator Guide	Notes
Summary	In this mission, students learned and applied the foundational concepts necessary to: • Recognizing the difference(s) between an Avatar and a Robot • Demonstrating fluid transition in programming multiple mediums. • Creating a visual representation of a program addressing a basic task. • Placing objects in a digital space. • Demonstrating how to program a script for the Avatar/Robot.	
Competency Achieved	Competency Achieved: Basic	Basic Competency is achieved using this curriculum by completing the mission, the three activities and THE assessment(s) included within the mission.

Requirements

- Computer
- Avatar

Resources

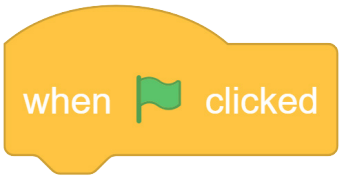
- Mission breakdown
- Pen/Pencil
- Paper
- Rubric
- Roles & Responsibilities Breakdown
- Group Reflection Template
- Individual Reflection Template

Look and listen Fors

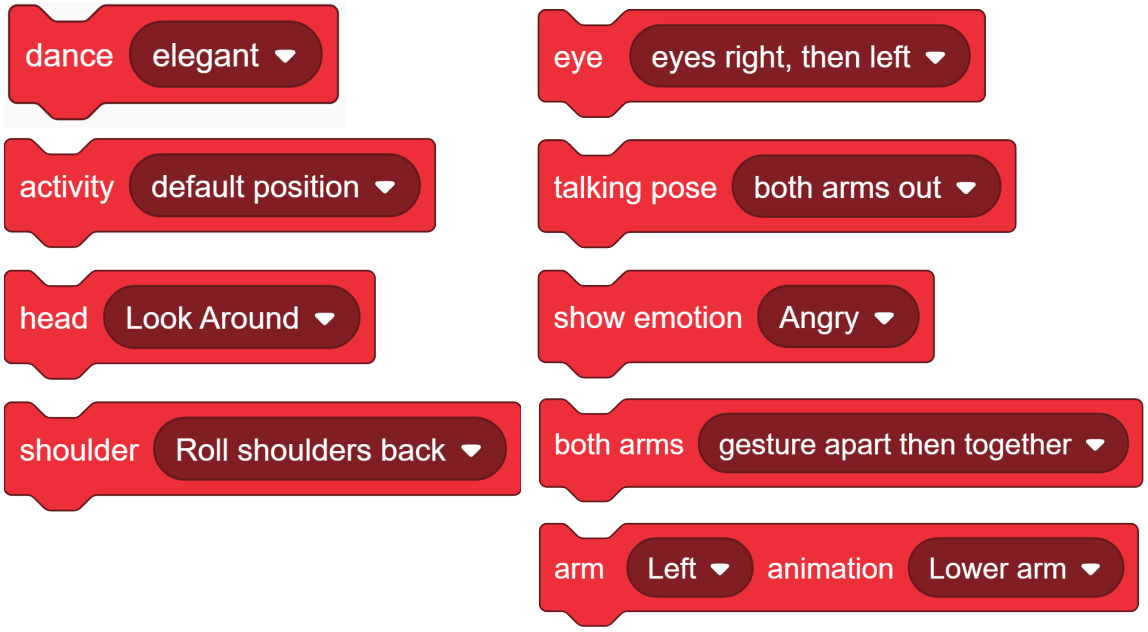
- Make sure that learners know to use "split screen" to access the avatar. If they do not and simply change tabs the avatar will pause animations until they navigate back to the Avatar Tab.
- Learners may have trouble navigating and reconciling the number of tabs at this point, make sure that you are available to help them organize since this is the first time that they will have this many windows or tabs open with this program.

Blocks Required

Events



Jett Actions



Mission 6

Introduction to Working with Avatars
Activities Facilitator Notes

Mission Assignments

1 Level Explore Jett Actions

This assignment is open exploration time of the Jett Actions palette. Allow 5-10 minutes for students to try as many blocks and options as they can. There is no requirement to submit a program.

1 Level Four Action Program

This assignment requires the students to create a program that uses a minimum of four separate Jett Actions blocks and must also include the connect to block that is necessary for the Workspace code to interface with the Avatar. While any 4 Jett Action blocks would meet the basic requirement, you may find students who are eager to build a meaningful project. It will be at your discretion to have the students export this program for grading.

Note: All programs written in the Workspace for the Avatar will also work on the live robot. If you have the robot available to you in the classroom, this may be an opportune time to have your students interact with the robot through the code they are writing.
Have them submit their code to you, and then you can connect to the robot to run their code submissions. It should result in a highly engaging mission!

Additional Activities

Avatar Dance Moves

Students were asked to make the Avatar do a dance without using a dance block. Experiment with different action blocks to create a program that shows off a whole new set of moves! The requirements for this activity are:

- Use the Connect to block to connect the Workspace to the Avatar
- Use at least 3 activity blocks to create a new set of dance moves
- Do not use the Dance activity block

Students were asked to export the project to a file named TheirName_AvatarDanceMoves.

An example program named *Mission6_AvatarDanceMoves.sb3* is provided.

Avatar Party

The Avatar is hosting a party (with invisible guests)! Students are to create a program that shows how the Avatar would start the party. Here are some questions to consider for designing and creating your program:

- What would he say? How does he welcome his guests?
- What would his emotions be? Is he happy? Nervous? Excited?
- Would he dance?
- What other actions might he do?

Students were asked to export the project to a file named TheirName_AvatarParty.

An example program named *Mission6_AvatarParty.sb3* is provided.

Avatar Storyteller

Avatars can tell great stories! Students are to create a program for the Avatar to tell a short story of their choice. To be an animated storyteller, their Avatar program should have:

- Speaking
- Emotions
- Head movements
- Arm movements

Students were asked to export the project to a file named TheirName_AvatarStoryteller.

An example program named *Mission6_AvatarStoryteller.sb3* is provided.

MISSION 6

Introduction to Working with Avatars
Additional Activities

1 LEVEL Avatar Dance Moves

Can you make the Avatar do a dance without using a dance block? Experiment with different action blocks to create a program that shows off a whole new set of moves!

The requirements for this activity are:

- Use the "Connect to" block to connect the Workspace to your Avatar
- Use at least three activity blocks to create a new set of dance moves
- Do not use the "Dance" activity block

When you are finished testing your program, export your project. Name it *YourName_AvatarDanceMoves.sb3* (replace YourName with your actual name) and then submit it to your facilitator.

2 LEVEL Avatar Party: Jett Saves the Day!

The Avatar is hosting a party (with invisible guests)! Create a program that shows how the Avatar would start the party. Here are some questions to consider for designing and creating your program:

- What would he say? How does he welcome his guests?
- What would his emotions be? Is he happy? Nervous? Excited?
- Would he dance?
- What other actions might he do?

Avatar Storyteller: Jett Breaks the Ice!

Avatars can tell great stories! Create a program for your avatar to tell a short story of your choice. To be an animated storyteller, your Avatar program should have:

- Speaking
- Emotions
- Head movements
- Arm movements

When you are finished testing your program, export your project. Name it *YourName_AvatarStoryteller.sb3* (replace YourName with your actual name) and then submit it to your facilitator.

robots4STEM

| MISSION AT A GLANCE |

MISSION 7: Introduction to Key-Press Events

[1 Hour, 30 Minutes]

Mission Overview	WithIn this mission, the learner will create programs that demonstrate event-driven programming as a way to make programs more interactive and interesting. Students learn the basics of writing interactive programs.	
Mission Objectives	• Define an event in programming terminology • Students learn how to use the blocks to trigger events • Use key-press events to cause the execution of actions using sprite and avatar • Demonstrate knowledge of multiple events in one program • Differentiate motion blocks from blocks of other palettes	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Variables. Abstraction Control. Abstraction Control. Communication and Coordination	
Connections within Framework	3-5. Data and Analysis. Collection 3-5. Data and Analysis. Storage K-2. Computing Systems. Devices	
Materials	• Reflection Tool • Access to Online Content • Flow Chart	
Introduction	• Facilitator introduces mission • Students open all the tools	5 minutes
Online Work	• Student complete missions • Student completes quiz	10 minutes
Individual Workspace Activity	Programming Events 1. Student programs an event to change the costumes of sprites. 2. Student programs events to change the size of a sprite 3. Student submits flowchart, script and reflection for each program	20 minutes
Debrief Workspace Activity	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work.	15 minutes

Online Work	1. Student complete mission 2. Student completes quiz	10 minutes
Individual Workspace Activity	1. Student programs an event to prompt an avatar action. 2. Student submits flowchart, script, and reflection for each program	20 minutes
Conclusion	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	15 minutes

MISSION 7

Introduction to Key-Press Events
MISSION PLAN

Subject	Grade	Mission length
Learn to make interactive programs by using the keyboard to activate blocks/events.	3rd-5th	80 minutes

OVERVIEW

In this mission, students learn the foundational concepts of making interactive programs

	Facilitator Guide	Notes
Objectives	In this lesson students will learn and demonstrate the following concepts: • Creating interactive programs using the appropriate blocks within The Workspace; and • Applying the correct blocks to the appropriate medium (avatar or robot).	

	Facilitator Guide	Notes
Important vocabulary	1. Interactive 2. Event-driven Programming 3. Event 4. Key-press Event 5. Block 6. Hat Block 7. Avatar 8. Input	1. **Interactive -** Programs that performs actions based upon the input of the person(s) using the program. 2. **Event-driven Programming -** When the flow of a program is determined by events from the user, rather than a specified sequential order. 3. **Event -** Any action performed that causes a reaction in a computer program. 4. **Key-press Event -** When the user presses a key on the keyboard to make an action happen in the program. 5. **Block -** One specific instruction located within a palette in The Workspace. Blocks fit together like puzzle pieces to create computer programs. 6. **Hat Block -** Blocks in The Workspace that can only be placed at the top of a script and specify the event that must occur for the script to be launched. 7. **Avatar** - A computer image or program that represents something in real life. 8. **Input** - Any information or data that is sent to a computer or device for processing.
Activity	I. Programming Events Mission II. Activity 1: Avatar Controls	Learners must submit: • Flowchart (1) • At least 1 Reflection Paragraph • .sb3 File (1)
Timing breakdown	I. The mission "Introduction to Programming Events" (10 minutes) II. Activity 1: Avatar Controls (45 minutes) III. Demonstration (10 Minutes)	I. The lesson "Introduction to Programming Events" (10 Minutes) II. Activity 1: Learners should apply key-press events to their avatar partner (45 minutes) ↳ Learners should program the avatar to use three separate key-press events to interact with the avatar. ↳ Learners should use 45 minutes to make their programming as interesting and interactive as they can.

	Facilitator Guide	Notes
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners have two activities that are mandatory for progression. These are online activities that do not require facilitator intervention. In order for the student to progress to the next mission, the facilitator must approve that a student's created scripts run as intended and meets the requirements as outlined in the mission plan. In addition, learners should provide a flowchart (for each activity) that outlines how their program should run.	Facilitators should expect that learners may have trouble at first in programming multiple key-press events; as well as in the second activity learners may have to use the "green flag clicked" and "when _____ key is pressed" concurrently.
Summary	In this lesson students learned and demonstrated the following concepts: • Creating interactive programs using the appropriate blocks within The Workspace. • Applying the correct blocks to the appropriate digital medium (avatar or robot).	
Competency Achieved	Competency Achieved: Basic/Intermediate	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission. Intermediate Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission. In addition, the avatar extension activity teaches students to differentiate application of key-press events for the sprite and avatar.

Requirements

- Computer
- The Avatar

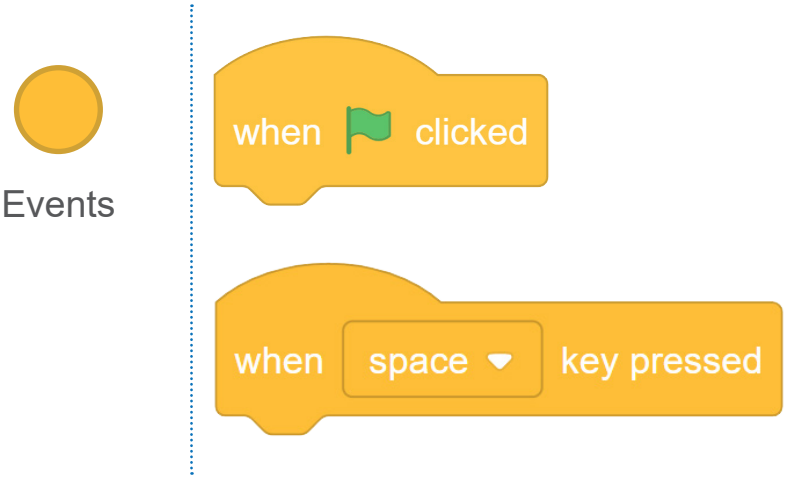
Resources

- Mission breakdown
- Pen/Pencil
- Paper

Look and Listen Fors

- Make sure that learners know to use "split screen" to access the avatar. If they do not and simply change tabs the avatar will pause animations until they navigate back to the Avatar Tab
- The event does not have to be used to start the program but they must use key-press events to create an interactive program. As an example, the additional activity in Mission 7 is to program the Jett Avatar to move by using the arrow keys).
- Learners may not need the "when green flag clicked" block but they may use it to start and connect their programs.

Blocks Required



MISSION 7

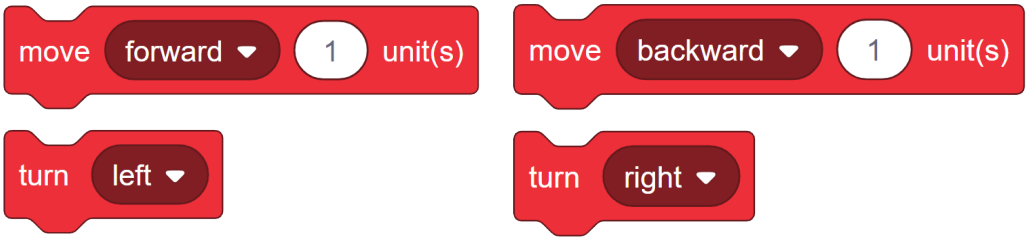
Introduction to Key-Press Events
Additional Activities

1 LEVEL Avatar Dance Moves

There will be times when students will need to write a script that controls the Avatar "directly". This activity is about allowing them to explore using key-press events to perform this function. Students should feel encouraged to get creative and explore the functions of both key-press events and the avatar. This activity also introduces and expands on the "turn" and "move" blocks in relation to the avatar (they were only used in reference to the Sprite before.)

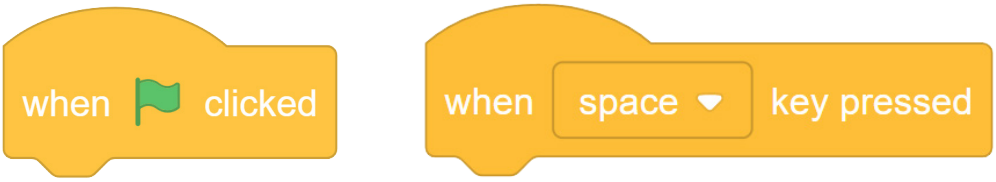
Blocks That MUST Be Used for This Activity

Jett Actions

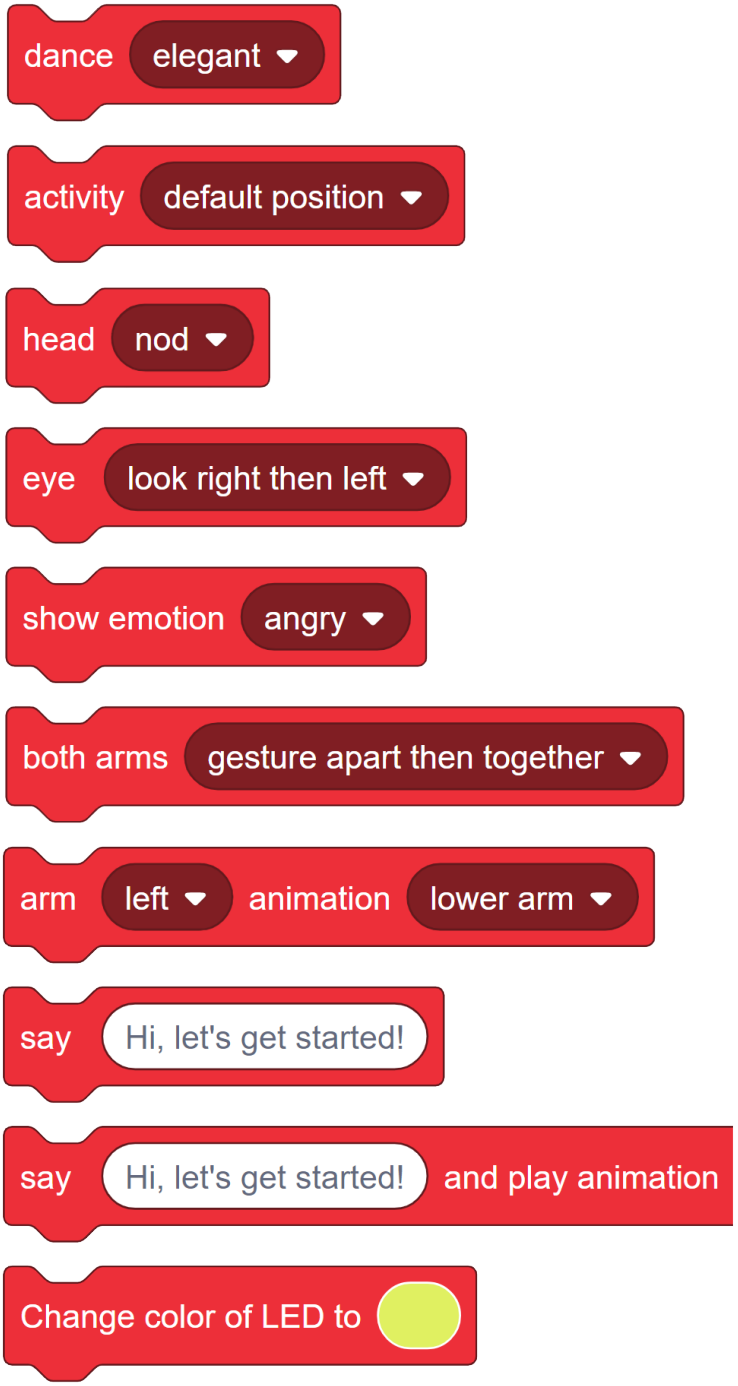


Blocks That MAY Be Used For This Activity

Events



Jett Actions



robots4STEM

| MISSION AT A GLANCE |

MISSION 8: Introduction To Variables

[6 Hours, 45 Minutes]

Mission Overview	WithIn this mission, students apply event-driven programing to create a program for sprite movement.	
Mission Objectives	• Differentiate motion blocks from blocks of other palettes • Determine x position on the x-axis after executing Change x by blocks. Determine y position on the y-axis after executing Change y by blocks. Determine stage location of a sprite after the execution of multiple Change by blocks. • Create programs that combine key-press events with motion blocks • Create an event-driven program that uses multiple key-press events. • Create an event-driven program that combines key-press events and blocks from the Motion palette. • Create a more complex event-driven program that contains multiple sprites each with multiple key-press events. Utilize the If on edge, bounce block to keep sprites from exiting the stage area. • Demonstrate understanding of event-driven programming topics and the stage placement of a sprite through x and y position manipulation	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Variables. Abstraction Control. Abstraction Control. Communication and Coordination Modularity. System Relationships Modularity. Abstraction	
Connections within Framework	3-5. Data and Analysis. Collection 3-5. Data and Analysis. Storage K-2. Computing Systems. Devices	
Materials	Reflection Tool Access to Online Content Roles and Responsibilities for Group Work Rubric for Avatar Project	
Introduction	Facilitator introduces missions and concept of project	5 minutes
Online Work	• Students Complete Missions • Students Complete Quiz	25 minutes

Online Activity	• Complete 1st Project • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	60 minutes
Group Debrief and Reflection	• Group discusses presentation and project • Group revises their rubric based upon feedback and discussion • Group completes the reflection tool • Group submits their reflection of the project	15 minutes

Mission 8

Introduction to Variables
Mission Plan

Subject	Grade	Mission length
Learn about how and when variables are used.	3rd-5th	80 minutes

Overview

In this lesson, students learn the foundational concepts of data types and variables.

	Facilitator Guide	Notes
Objectives	In this lesson students will learn and demonstrate the following concepts: • Recognizing the different data types provided in programming. • Creating or using digital objects such as visualizations, models or simulations.	

	Facilitator Guide	Notes
Important Vocabulary	1. Comparison 2. Interactive 3. Sensor 4. Program 5. Variable 6. Data 7. Data Type 8. Number 9. Text/String 10. Boolean	1. Comparison- Examining two things to note sameness or differences. 2. Interactive- Programs that perform actions based upon the input of the person using the program. 3. Sensor- A physical device that measures or detects a physical property and responds to it in some way. 4. Program- A set of ordered operations for a computer to perform. 5. Variable- A named container in a program that holds a single data item. 6. Data- A fact, such as a numeric value, measurement, result, or word. The term data can be used to mean either one fact or many facts. 7. Data Type- A particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it. 8. Number-A quantity value that can be used in mathematical statements. 9. Text/String- A sequence of characters in a program. 10. Boolean- A special data type that only holds two possible values: True or False.
Activity	I. The mission "Introduction to Variables" II. Activity 1: VariaBubbles III. Activity 2: Cosmic Cleanup	Learners must submit: • Flowchart (1) • Reflection Paragraph (1) • .sb3 File (1)

	Facilitator Guide	Notes
Timing Breakdown	I. The mission "Introduction to Variables" II. Activity 1: VariaBubbles (15 minutes) III. Activity 2: Cosmic Cleanup (15 minutes)	I. The mission "Introduction to Variables" (15 Minutes) II. Activity 1: Variabubbles (15 minutes) ↳ Students will open the game "VariaBubbles" ↳ The game gives students the ability to demonstrate knowledge of different data types that variables fit into. ↳ Students will try to achieve the highest score that they can by matching correct datatypes with the variables that appear in the bubbles. III. Activity 2: Cosmic Cleanup (15 minutes) ↳ Students will open the game Cosmic Cleanup. ↳ The game gives students the opportunity to recognize the appropriate operators to use. ↳ Students will recognize that the ship will use either "<", ">" or "=" and a numerical value. ↳ Students will then find numbers that match the operator shown on the ship. Five matches and they will grow to the next level. ↳ Once they reach the tenth level, students will win the game.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the lesson. Learners have one activity that is mandatory for progression. This is an online activity that does not require facilitator intervention. Facilitators may also find that they must direct their students to additional optional lessons to do the more advanced lesson provided.	Once the games have been introduced, they should be used as a warm up in every lesson. At the beginning of each class, facilitators should write a leaderboard with the game's scores. This will set the foundation for later lessons that talk about what to do with variables and using lists. If there is an update of the leaderboard, require a screenshot. There is no requirement that students add to their "Personality is Key" script as the learned concepts are to add to the understanding of previous concepts. There will be more detail provided on variables in Course 2 of the robots4STEM® curriculum.

	Facilitator Guide	Notes
Summary	In this mission, students will learn and demonstrate the following concepts: • Recognizing the different data types provided in programming; and • Creating or using digital objects such as visualizations, models or simulations.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

- Mission breakdown
- Pen/Pencil
- Paper

Look and listen Fors

- Students having trouble with naming variables.

Blocks Required

Events

robots4STEM®

| MISSION AT A GLANCE |

MISSION 9: Mid-Course Promotion Event Project

Overview

Learners will apply and demonstrate competency in the following areas:

- Digital citizenship;
- Algorithmic thinking using algorithms and sequences;
- Designing a computer program;
- Using programming concepts in a digital environment;
- Programming motion along an XY plane;

Objective

Ensigns and their avatar will give a short presentation on how robot, computers and computer science can or do help humanity. Both the avatar and the student should be talking and taking turns. They will have to use the wait block and all of the other skills that they have learned in the program.

Students will begin with the help of their facilitator by researching a topic of their choice.

Ensigns must provide a brief outline of their presentation (possibly using the algorithm/flowchart format).

Next, students will create an algorithm and flowchart for the avatar's portion of the presentation.
Ensigns will then program the avatar's portion of the presentation using R4C guidelines

R4C Program Guidelines

- Must be activated using a key-press event.
- Must use at least one wait block.
- Must use at least five actions for the avatar.
- Must answer all three questions.

Phases of planning (Time breakdown)

Introductory/ Prep Lesson for Promotion Event Project (60 Minutes)
Building the Program/.sb3 (60 Minutes)
Rehearsal (60 Minutes)
Demonstration (120 Minutes)
Documentation (60 Minutes)

Requirements

Students will be required to provide the following documentation

- 1x Flowchart or Algorithm Detailing Process to achieve the given task before working with a partner
- Research List
- Reflection Paragraph/Documentation
- 1x Flowchart or Algorithm detailing the process to achieve the given task after working with a partner
- Group Reflection Paragraph/Documentation
- 1x Powerpoint (or similar) presentation explaining what the student-programmed and why they programmed it the way that they did
- 1x .sb3 file

Provided Resources Necessary to Grade the Final Promotion Event

Facilitators have been provided the following resources to assess student competency on the final project:

- Reflection template found on **page 127** of the Appendix.
- Group Reflection template found on **page 128** of the Appendix.
- Group roles and responsibilities on **page 129** of the Appendix.
- Rubric found on **page 131** of the Appendix.

robots4STEM®

| MISSION AT A GLANCE |

MISSION 10: Introduction to Conditionals

[2 hours 15 minutes]

Mission Overview	Students play a game using conditional logic, and then develop their own descriptive steps using conditions to create a new game as a team.	
Mission Objectives	• Demonstrate the ability to create conditional Statements • Define condition, action, and conditional Statement • Play the Conditionals Game • Engage in team-based algorithm writing and testing using conditional statements • Demonstrate theoretical and applied knowledge of conditional statements	
Core Concepts Subconcepts. Crosscutting Concepts	**Algorithms and Programming** **Control.** Abstraction **Modularity.** System Relationships. **Modularity.** Abstraction **Program Development.** Human-Computer Interaction **Program Development.** System Relationships **Impacts of Computing** **Social Interactions.** System Relationships **Social Interactions.** Human-Computer Interaction	
Connections within Framework	K-2. Networks and the Internet. Network Communication and Organization K-2. Computing Systems. Troubleshooting K-2. Computing Systems. Devices	
Materials	• Online access to Mission • Conditional worksheet • Conditionals Game Instruction Sheet • Roles and Responsibilities for Group Work • Rubric for Game Creation (To be Created)	
Introduction	Facilitator introduces class and student begin with online work	5 minutes
Online Work	Student complete missions Student completes quiz	20 minutes

Individual Workspace Activity	• Complete 1st Project • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	30 minutes
Group Project Conditionals Game (3-4 per group)	• Design Project • Test Project • Plan presentation of Project • Submit ↳ Flowchart Plan ↳ .sb3 File ↳ Rubric for the project	60 minutes
Debrief and Close	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	20 minutes

MISSION 10

Introduction to Conditionals (Part 1)
MISSION PLAN

Subject	Grade	Mission length
Learn how to activate blocks/events if certain conditions are met.	3rd-5th	95 minutes

OVERVIEW

In this mission, students will learn the foundational concepts of using conditional statements in programming

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following concepts: • Applying conditional thinking to perform more complex tasks. • Constructing programs using basic automation. • Applying basic mathematical logic programming.	
Important Vocabulary	1. Algorithm 2. Action 3. Condition 4. Conditional Statement	1. Algorithm- A group of instructions designed to perform a specific task. 2. Action- An instruction that is executed in a computer program. 3. Condition- A statement that evaluates to true or false. 4. Conditional Statement- A programming construct that performs different actions depending on whether the tested condition is true or false.

	Facilitator Guide	Notes
Activity	I. Introduction to Programming Conditional Statements Mission II. Activity 1: Race Jett Omega III. Demonstration	Students must submit: • Flowchart (1) • At least 1 Reflection Paragraph • .sb3 File (1)
Timing Breakdown	I. The mission "Introduction To Conditionals (Part 1)" (25 Minutes) II. Activity 1: Navigate the Simulation Room (25 minutes) III. Activity 2: Cosmic Cleanup (15 minutes) IV. Ongoing Activity: Personality is Key (20 minutes) V. Demonstration (10 minutes)	I. The Mission "Introduction To Conditionals (Part 1)" (25 Minutes) II. Activity 1: Navigate the Simulation Room (25 minutes) • Students will start in The Simulation Room at position (5,1) facing up. • Students are allowed to use one "key press event" to move around The Simulation Room. • Students must use the conditional thinking blocks (e.g. "if...then" and similar) and the sensing block. • Facilitators should be able to run the program without adapting it at all and the process should be clearly outlined in the flowchart/reflection document. III. Activity 2: Cosmic Cleanup (15 minutes) • Students will open the "Cosmic Cleanup" game in the LMS. • The game allows students to demonstrate their understanding of "greater than", "less than" and "equal to" in relation to simple numerical values. • The player's ship will display an operator that students will have to comply with. • If a student finds five integers that comply with the displayed operator they will grow in size and advance a level. • After reaching level 10 the game will end successfully. • If a student uses the incorrect integer for the displayed operator they will drop down a level. • Once students reach level "0" it is game over for them.

	Facilitator Guide	Notes
Timing Breakdown (Continued)		IV. Ongoing Activity: Personality is Key (20 minutes) • Now that students have learned about the basics of conditional thinking they can move onto the next stage of programming in their "Personality is Key" program. • Students should place obstacles or automate how their avatar will interact with obstacles that they have already placed. • Remind students that their content must include a description of what they learned and how they applied it into their ongoing program.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from students throughout the mission. Students must complete one activity in order to progress to the next mission. This is an online mission and does not need direct facilitator intervention, but facilitators should be available to support students.	
Summary	In this mission, students learned and practiced: • Applying conditional thinking to perform more complex tasks. • Constructing programs using basic automation. • Applying basic mathematical logic in programming.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

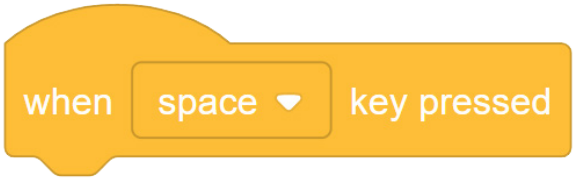
- Mission breakdown
- Pen/Pencil
- Paper

Look and Listen Fors

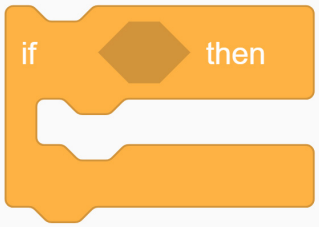
- Students may get frustrated with the programming aspect of this activity. Depending on their measure of success students may get frustrated with their progress with this activity. Several topics are being used in conjunction for this mission such as mathematics, visual block programming, grammar, unit measurement and graphing.

Blocks Required

Events



Control



Operators

Variables

Jett Actions

Jett Simulation

Any block(s) from these palettes may be used.

Jett Sensors

robots4STEM

| MISSION AT A GLANCE |

MISSION 11: Introduction to Programming Loops

[3 Hours, 35 minutes]

Mission Overview	Students develop algorithms and create programs using the concept of looping
Mission Objectives	• Recognize the issues with having multiple copies of instructions in an algorithm • Alter an existing instruction to change a loop Condition • Define Definite/Counting Loop • Exhibit use of the Repeat block to loop Instructions • Trace existing code to determine correct Outputs • Define Infinite Loop • Exhibit use of the Forever block to loop instructions • Define Nested Loop • Create a program utilizing a basic definite loop • Create a program utilizing a basic infinite loop • Combine Events, Motion, and Loops to develop an interactive program • Demonstrate the use of nested loops in a program • Demonstrate the use of nested loops in a program • Demonstrate basic knowledge of loops through definitions and code tracing
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Control. Abstraction Modularity. System Relationships. Modularity. Abstraction Program Development. Human-Computer Interaction Program Development. System Relationships
Connections within Framework	K-2. Networks and the Internet. Network Communication and Organization K-2. Computing Systems. Troubleshooting K-2. Computing Systems. Devices
Materials	• Online access to Mission • Flow Chart • Reflection

Introduction	Facilitator introduces class and student begin with online work	5 minutes
Online Work	Student complete missions Student completes quiz	20 minutes
Individual Workspace Activity	• Complete Mission • Complete 2 projects • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	150 minutes
Debrief and Close	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	25 minutes

MISSION 11

Introduction To Programming Loops
MISSION PLAN

Subject	Grade	Mission length
Learn to repeat blocks/events a set number of times or throughout the program.	3rd-5th	60 Minutes

OVERVIEW

In this mission, students learn the foundational concepts of loops to repeat tasks a finite number of times or to run until a user stops the program manually.

	Facilitator Guide	Notes
Objectives	In this mission, students will learn and demonstrate the following concepts: • Applying basic automation of programs through definite loops. • Demonstrating basic automation of programs through infinite loops.	
Important vocabulary	1. Algorithm 2. Action 3. Condition 4. Conditional Statement	1. Algorithm - A group of instructions designed to perform a specific task. 2. Action - An instruction that is executed in a computer program. 3. Condition - A statement that evaluates to true or false. 4. Conditional Statement - A programming construct that performs different actions depending on whether the tested condition is true or false.
Activity	I. Introduction to Programming Loops II. Activity 1: Pick Me III. Activity 2: It's All Automatic	Learners will be required to submit at minimum: • Flowcharts (2) • .sb3 files (2) • Reflection Documents (2)

	Facilitator Guide	Notes
Timing Breakdown	I. The mission "Introduction to Programming Loops" (30 minutes) II. Activity 1: Pick Me (10 minutes) III. Activity 2: It's All Automatic (10 minutes)	I. The mission "Introduction to Programming Loops" (30 Minutes) II. Activity 1: Pick Me (10 minutes) ↳ Jett's avatar is excited because he knows the answer to a question. ↳ Write a program that connects to the avatar. The avatar should then wave his arm in the air and say "I know! Pick me!" 3 times. ↳ Be sure to use a loop in your program! III. Activity 2: It's all Automatic: Navigate The Simulation Room (10 minutes) • Having navigated The Simulation Room using one key-press event students will now attempt to automate that process using conditional thinking and loops. • Student's programs will be expected to run by themselves without facilitator interference once it has been started. • Student's flowcharts should include what loops they used and why. • Students should use at minimum a "Repeat x times" block OR a "Repeat forever" block • Students should program the avatar to run the course forever using both conditionals and forever loops.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners must complete a minimum of two activities in order to progress to the next mission. These are online missions and do not require facilitator intervention, however , facilitators should be available to support learners.	Learners are required to do fairly simple activities to progress to the next mission, however, it may be up to the facilitator to require up to both additional activities if the facilitator feels that the learner could use more practice or if time allows.

	Facilitator Guide	Notes
Summary	This mission provided instruction and practice for students within The Workspace for the following: • Using the "repeat" block • Using the "forever" block	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

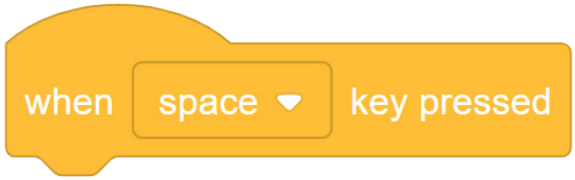
- Mission breakdown
- Pen/Pencil
- Paper

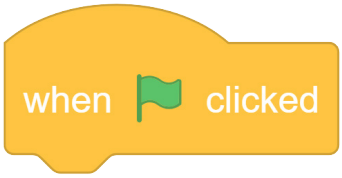
Look and listen Fors

- This is an extensive and time-consuming mission, it is important that learners understand not only the "how" but also the "why" of programming at this point. Looping is designed to repeat steps that learners have had to program multiple times. Learners will need to figure out what makes the most sense to repeat and where to make that distinction.

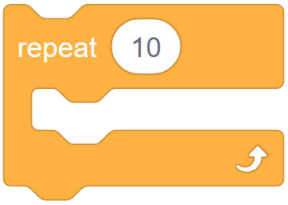
Blocks Required

Events





Control



robots4STEM

| MISSION AT A GLANCE |

MISSION 12: Conditionals Continued

[2 hours]

Mission Overview	Students break a problem into manageable parts through using conditional statements to programmatically determine the correct result from several options. Students will evaluate conditions and choose an execution path based upon the value of the evaluated condition.
Mission Objectives	• Determine the evaluated value of a comparison • Evaluate comparison statements • Determine the execution path of an if/else statement based upon the evaluated value of a condition • Choose the correct execution path of a presented if/else statement • Determine the correct action in a nested if statement based upon multiple evaluated conditions • Evaluate multiple conditions to reach the correct action within a nested if statement • Demonstrate the ability to choose an execution path in conditional statements based upon the value of an evaluated condition • Create, play, and test a custom game using nested if statements • Demonstrate the ability to choose the correct execution path through an algorithm using nested if statements
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Control. Abstraction Modularity. System Relationships. Modularity. Abstraction Program Development. Human-Computer Interaction Program Development. System Relationships Impacts of Computing Social Interactions. System Relationships Social Interactions. Human-Computer Interaction
Connections within Framework	K-2. Networks and the Internet. Network Communication and Organization K-2. Computing Systems. Troubleshooting K-2. Computing Systems. Devices

Materials	Online access to Mission Nested If Custom Game Nested If Game Rules Roles and Responsibilities for Group Work Rubric for Game Creation (To be Created) Dice or JettDieRoller.sb3 for the avatar/robot to call dice rolls to the whole class NestedIfCustomGame.pdf worksheet	
Introduction	Facilitator introduces class and student begin with online work	5 minutes
Online Work	Student complete missions Student completes quiz	20 minutes
Group Activity (continued)	• Design Game Project (continued) • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	75 minutes
Debrief and Close	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	20 minutes

MISSION 12

Conditionals Continued
MISSION PLAN

Subject	Grade	Mission length
Learn to activate blocks/events based on multiple conditions or if certain conditions are met.	3rd-5th	120 minutes

OVERVIEW

This mission teaches students to further apply the concepts of using conditional statements in programming to allow for multiple outcomes based on predetermined settings

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following concepts: • Applying more advanced concepts of automation using the appropriate blocks. • Applying more advanced concepts of computational thinking through the "nested if" concept.	
Important Vocabulary	1. Loop 2. Condition 3. Flowchart 4. Decision	1. Loop-A sequence of instructions that is repeated over and over until a specific situation occurs. 2. Condition-A statement that evaluates to true or false. 3. Flowchart- A diagram that is a pictorial representation of an algorithm. 4. Decision- A diamond shaped symbol in a flowchart that represents asking a question or testing a condition.

	Facilitator Guide	Notes
Activity	I. The mission "Conditionals Continued" II. Activity 1: Repair the Space Station Storage System III. Demonstration	Learners will be required to submit at least: • Flowchart (1) • Reflection Paragraph (1) • .sb3 File (1)
Timing Breakdown	I. The mission "Conditionals Continued" (20 Minutes) II. Activity 1: Reach For The Goal (75 Minutes) III. Demonstration (15 minutes)	I. The mission "Conditionals Continued" (20 Minutes) II. Activity 1: Reach for the Goal (75 Minutes) • Having navigated The Simulation Room using an automated program; students will now attempt to reach a generated goal using conditional thinking and loops. • Student's programs will be expected to run by themselves without facilitator interference once it has been started. • Once student's avatar reaches the goal they should celebrate and then return to the starting position to continue running the program. • This program should run without facilitator intervention and should be building on the existing programs that students have worked on. • Student's flowcharts should include what loops they used and why.
Facilitator's Role	The facilitator should expect to be available to answer basic questions from learners throughout the mission. Learners must only complete and demonstrate one activity to progress to the next mission. This is an online activity and does not require facilitator intervention, however, facilitators should be available to support learners.	Facilitators should expect that learners may make mistakes in the use of repeat blocks and functions while nesting them within one another. If they do so incorrectly the program may not run so it may be a good idea to remind them to check the order in which they use their blocks and encourage them to review the missions if it continues to be a problem. Additionally, this may be a good time to introduce the final project.

	Facilitator Guide	Notes
Summary	In this mission, students learned and demonstrated the following concepts: • Applying more advanced concepts of automation using the appropriate blocks. • Applying more advanced concepts of computational thinking through the "nested if" concept.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

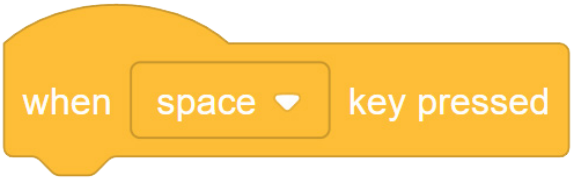
- Mission breakdown
- Pen/Pencil
- Paper

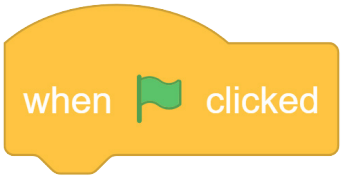
Look and Listen Fors

- This is an extensive and time-consuming mission, it is important that learners understand not only the "how" but also the "why" of programming at this point. Nesting conditionals are integral to programming and learners should evaluate where it makes sense to use them in programming. Be patient with learners and encourage them to return to the mission frequently. It may be worthwhile to have them open the mission in a separate window.

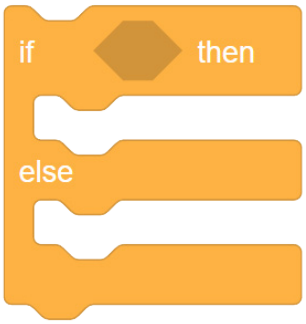
Blocks Required

Events





Control



robots4STEM

| MISSION AT A GLANCE |

MISSION 13: Programming Loops Continued - Indefinite Loops

[2 hours, 15 minutes]

Mission Overview	Students categorize problems to determine if they are best solved programmatically through the use of the concept of an indefinite loop. Students will create multiple computer programs using the concept of an indefinite loop and will be able to determine scenarios that are appropriate to be solved using this construct.	
Mission Objectives	• Students will differentiate between counting and indefinite loop scenarios. • Students will demonstrate an understanding of the concept of indefinite loop • Students will understand the usage of the repeat until block • Students will trace the execution path of a code snippet against its flowchart • Students will categorize scenarios into counting or indefinite loop options, match a flowchart to code, and demonstrate understanding of the programmatic execution of a coded indefinite loop.	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming Control. Abstraction Modularity. System Relationships. Modularity. Abstraction Program Development. Human-Computer Interaction Program Development. System Relationships	
Connections within Framework	K-2. Networks and the Internet. Network Communication and Organization K-2. Computing Systems. Troubleshooting K-2. Computing Systems. Devices	
Materials	Online access to Mission Flowchart Reflection .sb3 File	
Introduction	Facilitator introduces class and student begin with online work	5 minutes
Online Work	Student complete missions Student completes quiz	20 minutes

Individual Workspace Activity	• Complete Mission • Complete 2 projects • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	90 minutes
Debrief and Close	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	20 minutes

MISSION 13

Programming Loops Continued - Indefinite Loops
MISSION PLAN

Subject	Grade	Mission length
Learn to loop blocks/events until a certain condition is met. These loops repeat an indefinite amount of times.	3rd-5th	135 minutes

OVERVIEW

This mission teaches students to further apply the foundational concepts of loops in programming to write scripts that run until a condition is true without knowing the number of times that the loop will run.

	Facilitator Guide	Notes
Objectives	In this mission, students learn and demonstrate the following concepts: • Recognizing the appropriate blocks necessary to complete a complex task. • Creating a fully automatic script using the appropriate blocks to complete a complex task.	
Important Vocabulary	1. Loop 2. Condition 3. Flowchart 4. Decision 5. Indefinite	1. Loop-A sequence of instructions that is repeated over and over until a specific situation occurs. 2. Condition-A statement that evaluates to true or false. 3. Flowchart- A diagram that is a pictorial representation of an algorithm. 4. Decision- A diamond shaped symbol in a flowchart that represents asking a question or testing a condition. 5. Indefinite- Repeating a set of actions until a condition is true instead of a specific number of times.

	Facilitator Guide	Notes
Activity	I. The mission "Programming Loops Continued - Indefinite Loops" II. Activity 1: Overcoming Obstacles - Rushing To The Goal	• Learners will submit at least: • Flowcharts (1) • .sb3 Files (1) • Reflection Documents (1)
Timing Breakdown	I. The mission "Programming Loops Continued - Indefinite Loops" (20 Minutes) II. Activity 1: Overcoming Obstacles - Rushing To The Goal" III. Demonstration (15 minutes)	I. The mission "Programming Loops Continued - Indefinite Loops" (20 Minutes) II. Activity 1: Overcoming Obstacles: Rushing to the Goal ↳ Students will try to reach the goal on the eleventh row of the y-axis of The Simulation Room using an automated program. ↳ Students will move forward "until" they reach a value on 'y' . ↳ They will have to use conditional thinking to avoid the obstacles that are currently there. ↳ Once they reach the appropriate y-value they will have to turn right and take steps forward. ↳ At a minimum students must have the looping block "until" in their program. ↳ Facilitators should be able to run this program without any interaction on their part.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the mission. Learners must complete a minimum of two activities in order to progress to the next mission. These are online missions and do not require facilitator intervention.	

	Facilitator Guide	Notes
Summary	In this mission, students learned and demonstrated the following concepts: • Recognizing the appropriate blocks necessary to complete a complex task. • Creating a fully automatic script using the appropriate blocks to complete a complex task.	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the digital citizenship mission. Basic Competency: The student has completed the mission and all activities and quizzes necessary to progress to the next mission.

Requirements

- Computer

Resources

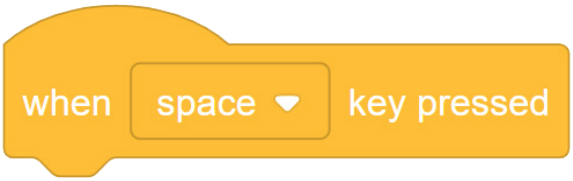
- Mission breakdown
- Pen/Pencil
- Paper

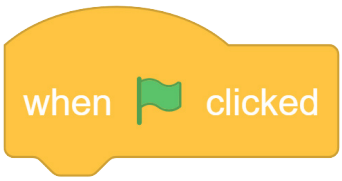
Look and Listen Fors

- This is an example of an incorrect nested loop. If your learners continually have problems it may be worthwhile to compare their script to the above.

Blocks Required

Events





Control

robots4STEM

| MISSION AT A GLANCE |

MISSION 14: Promotion Project Essentials

[1 hours, 40 minutes]

Mission Overview	Students will learn about how to create their own space with digital objects and how to use sensors and loops to create a fully automated program. This program will react to objects in the space to achieve a goal.	
Mission Objectives	• Students will use objects to create a virtual space of their own design. • Students will utilize sensors to navigate the virtual space that they create. • Students will use conditional statements to demonstrate the fundamentals of automating their program. • Students will demonstrate knowledge of appropriate loops to fully automate a program of their own creation.	
Core Concepts Subconcepts. Crosscutting Concepts	Algorithms and Programming, Computing Systems Control. Abstraction Modularity. System Relationships Modularity. Decomposition Variables. Abstraction Program Development. Troubleshooting Program Development. System Relationships	
Connections within Framework	K-2. Networks and the Internet. Network Communication and Organization K-2. Computing Systems. Troubleshooting K-2. Computing Systems. Devices	
Materials	• Online access to Mission • Flowchart • Reflection • Reflection document • .sb3 File	
Introduction	Facilitator introduces class and student begin with online work	5 minutes
Online Work	Student complete lessons Student completes project	45 minutes

Individual Workspace Activity	• Complete Mission • Complete 1 Project • Submit project ↳ Flow Chart ↳ .sb3 File ↳ Reflection	30 minutes
Debrief and Close	Facilitator debriefs with students and asks for volunteers to discuss their reflection and work using the reflection tool.	20 minutes

Mission 14

Promotion Project Essentials
Mission Plan

Subject	Grade	Mission length
Learn how to use sensors and coordinates to interact with a digital medium on a single plane grid.	3rd-5th	60 minutes

Overview

In this mission, students will learn the foundational concepts of logical operators, detect object sensors and using coordinates in a single plane grid.

	Facilitator Guide	Notes
Objectives	In this mission students will learn and demonstrate the following concepts: • Using greater than, less than and equal to appropriately. • Using coordinates to navigate a single plane grid. • Using coordinates to place digital objects in a single plane grid. • Using sensors to automate how a programmed item interacts with placed objects.	
Important Vocabulary	1. Sensor	1. A physical device that measures or detects a physical property and responds to it in some way.

	Facilitator Guide	Notes
Activity		Students will need to submit a program that contains or performs the following before moving on: 1. One, "If...then" OR "if...else block". 2. One, "Repeat until" block. 3. One, "Sensing" block. 4. One of each; "greater than", "less than", "equal to" blocks. 5. Eight objects using the "place object at" block. 6. Program your script to run and complete using only one key-press to start the program. 7. Your avatar must walk from the bottom right of the Simulation room to the top left of the Simulation Room and interact with all placed objects.
Timing Breakdown	I. The lesson "Promotion Project Essentials"	I. The lesson "Promotion Project Essentials" will take approximately 60 minutes to complete.
Facilitator's Role	Facilitators should expect to be available to answer basic questions from learners throughout the lesson. Learners may have trouble understanding the change between a single plane grid and a grid with quadrants, facilitators should expect that they may have to spend some time explaining the difference between negative and positive values based on grids. Learners will not have learned about sensors in detail at this time, therefore facilitators should expect to be able to relate a robot sensor to a person touching an object in this example. Learners may also be directed online to learn more about different kinds of sensors at this time.	Facilitators will evaluate learner understanding of concepts necessary to complete the Final Promotion Project Event.

	Facilitator Guide	Notes
Summary	In this lesson students learned and applied the following concepts: • Using coordinates to navigate a single plane grid • Using coordinates to place digital objects in a single plane grid • Using sensors to automate how a programmed item interacts with placed objects	
Competency Achieved	Competency Achieved: Basic	Levels of competency achieved range from Basic to Advanced. In this curriculum "Basic Competency" is the highest level attainable in the Promotion Project Essentials lesson. Basic Competency: The student has completed the lesson and has successfully submitted a program that meets all requirements as expressed.

Requirements

- Computer

Resources

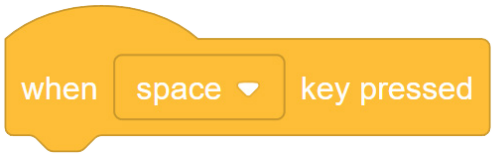
- Mission Plan

Look and Listen Fors

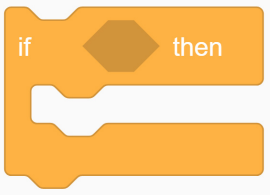
- N/A

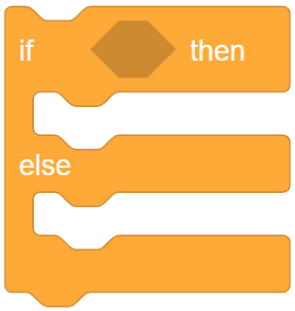
Blocks Required

Events

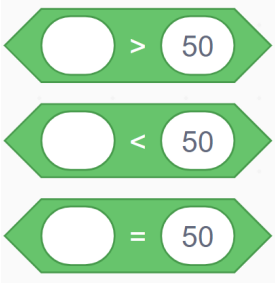


Control

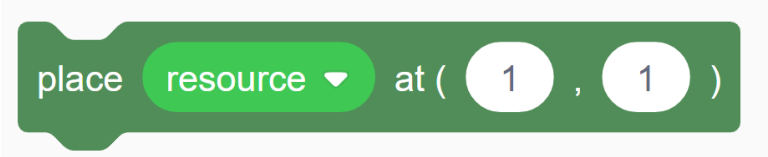




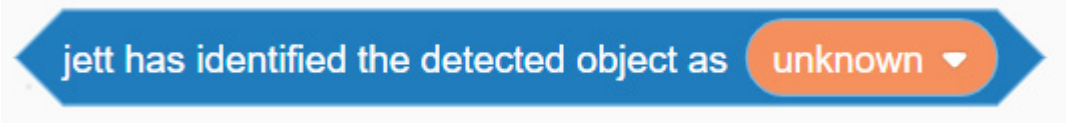
Operators



Simulation



Sensors



robots4STEM®

| MISSION AT A GLANCE |

MISSION 15: Final Promotion Project

[6 Hours, 20 minutes]

Mission Overview	Students complete a cumulative project by developing a visual program.	
Mission Objectives	• Project requires the use of the following: ↳ Connection to the avatar for some portion of the project ↳ A minimum of two if or if else blocks ↳ A minimum of one loop (any type) ↳ A minimum of two events (hat blocks) besides the green flag hat block. Students will connect the avatar for a part of the project • Students will create a written algorithm AND/OR a flowchart for the program BEFORE beginning code creation	
Core Concepts Subconcepts. Crosscutting Concepts	Assessment for Course	
Connections within Framework	Assessment for Course	
Materials	• Online access to Mission • Flowchart • Reflection • .sb3 File • Rubric	
Introduction	Facilitator introduces class and student begin with introduction project via online access	10 minutes

Individual Workspace Activity	• Research Mission • Create Flow Chart • Create .sb3 • Complete 1st project • Submit and Demo project ↳ Flow Chart ↳ .sb3 File ↳ Reflection ↳ Rubric	6 hours
Debrief and Close	Facilitator debriefs with students	10 minutes

Mission 15

Final Promotion Project
Project Plan

Overview

Learners will apply and demonstrate competency in the following areas:

- Digital citizenship
- Algorithmic thinking using algorithms and sequences
- Designing a computer program
- Using programming concepts in a digital environment
- Programming motion along an XY plane
- Creating an interactive program
- Using conditional thinking in designing programs
- Creating programs using the basics of automation

Objective

Researchers have placed a resource at the other end of the simulation room. The Ensign (student) must program the avatar to cross the simulation room from (11, 0) to a randomly generated location within the simulation room between the values of (11, 0) and (11, 2). The student must use everything that they have learned to create an automated program that can operate without user intervention once started.

Phases of planning (Time breakdown)

- Introductory/ Prep Lesson for Promotion Event Project (60 Minutes)
- Testing (60 Minutes)
- Building the Program/.sb3 (60 Minutes)
- Partner work/Group Work (60 Minutes)
- Demonstration (120 Minutes)
- Documentation (60 Minutes)

Requirements

Students will be required to provide the following documentation

- 1x Flowchart or Algorithm Detailing Process to achieve the given task before working with a partner
- Reflection Paragraph/Documentation
- 1x Flowchart or Algorithm detailing the process to achieve the given task after working with a partner
- Group Reflection Paragraph/Documentation
- 1x demonstration using presentation software explaining what the student-programmed and why they programmed it the way that they did
- 1x .sb3 file

Provided Resources Necessary to Grade the Final Promotion Event

Facilitators have been provided the following resources to assess student competency on the final project:

- Reflection template found on **page 127** of the appendix.
- Group reflection template on **page 128** of the appendix.
- Group roles and responsibilities on **page 129** of the appendix.
- Rubric found on **page 131** of the appendix

robots 4 STEM®

Curriculum Guide

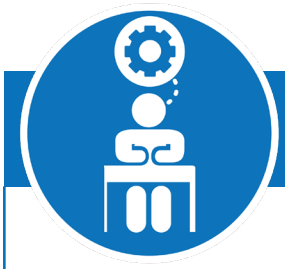
APPENDIX

In facilitating for a class using the **robots4STEM®** curriculum, you may find that you would like to use the curriculum resources for projects not included in the curriculum and that's great! The more ways students feel they can use the Robot, Avatar and The Workspace the better they will be able to "connect the dots" and feel at home using the resources that are provided. The purpose of this document and the accompanying documents is to empower facilitators to construct their own activities in class from beginning to end and grade them once they are completed.

---

1. **How to Create Your Own Activities:** Creating activities for your students does not require you to be an expert at visual programming. First and foremost you want to think of the goal.
    a. Look at the lesson plan provided in your **robots4STEM®** instance, it will give you a clear picture of the goal for any activity that is provided by RoboKind®.
    b. Write out the task that you would like your students to perform in paragraph form (it is okay if you do not know what processes a student would use to complete the task. It will follow the same format as all of our prior lessons in that they will provide a flowchart that describes how they would like to complete the process.)
    c. Require a flowchart- A flowchart is not negotiable for your students, this gives you a frame of reference on what they were trying to accomplish.
    d. Require a reflection document: A reflection document is non-negotiable. As with any part of the **robots4STEM®** curriculum, this gives you a document to compare against the flowchart and helps your student to think through any changes that they made or should have made. The reflection document should contain the following parts:
        i. What was the overall goal of the project itself?
        ii. What were the steps that the student took to complete the goals?
        iii. What blocks did the student use to complete the steps within the goal?
        iv. Was the student successful in completing the goal? If not, why?
2. **Rubrics**
    a. Individual Work
    b. Team Work

3. **How to give feedback for your projects:** All feedback given to your students should be with the intent of making sure that they know that it is okay to fail and try again. This is the attitude of a great programmer! Not having a background in programming is okay here, just try to ask questions that help them to arrive at a more precise answer, (e.g. "I see that you said that the program was supposed to do _____, however when I ran the program it did _____, why would that be?")



Individual Work Rubric

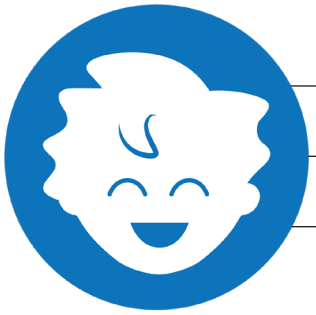
	Complete	Needs Improvement
Documentation » Flowchart » SB3 File » Reflection Document	• Student provided all documentation required to submit with the program.	• Student provided some or none of the documentation required to submit with the program.
Program	• Student's program addresses the task that they were assigned. • Student's program adheres to the flowchart that they submitted. • Student's program was provided in the appropriate format (SB3).	• Student's program addressed some or none of the task as it was assigned to the student. • Student's program did not adhere to the flowchart that they provided with the program. • Student's program was not provided in the appropriate format (SB3 file).
Flowchart	• Student's program addressed some or none of the task as it was assigned to the student. • Student's program did not adhere to the flowchart that they provided with the program. • Student's program was not provided in the appropriate format (SB3 file).	• Student's flowchart addresses some or none of the task which they were assigned. • Student's flowchart does not have a clear beginning and end. • Student's flowchart does not use the symbols appropriately within their flowchart to denote how the program should run.
Reflection Document	Student's reflection document addresses: » the task which they were assigned. » the flowchart which they provided to the teacher. » which blocks were used in each step to complete the task. » any challenges that they had in writing the program and why.	Student's reflection document does not address » the task which they were assigned in part or full. » the flowchart which they provided to the teacher. » which blocks were used in each step to complete the assigned task. » any challenges or lack of challenges that they had in writing the program.

Teamwork Rubric

	Complete	Needs Improvement
Cooperation	• Student listened carefully to what other team members had to say. • Student shared ideas and resources. • Student offered constructive feedback where appropriate.	• Student did not listen to what other team members had to stay. • Student did not share ideas and resources. • Student offered criticism that was not constructive or did not offer constructive feedback.
Problem Solving	• Student identified problems through talking with team members. • Student asked others for ideas concerning the task. • Student voted on the best solution to solve the problem and used the agreed upon solution.	• Student did not identify problems by talking with teammates. • Student did not ask for ideas concerning the task as assigned. • Student did not vote on or use an alternative means of solving the task that they were assigned. • Student did not use the agreed upon solution to solve the task as assigned.
Contribution	• Student shared ideas every time the group met. • Student completed all assignments with enough time for others to complete their assignments. • Student submitted all documentation required of them for their group.	• Student did not always have ideas when the group met. • Student did not complete all assignments with enough time for others to complete their assignments. • Student did not submit all documentation required of them for their group.

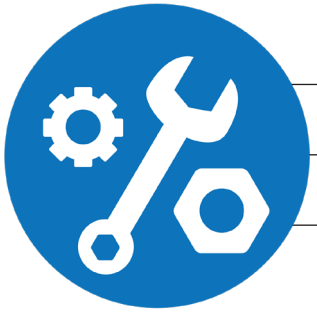
Name:

REFLECTION

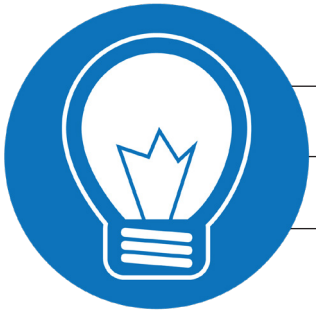


What Worked?

What Did Not Work?



How Did You Troubleshoot Challenges?



What Did You Learn?

Name:

Role Reflection

Team Leader

Name:

Did they follow their role? (yes/no)

What did they do well?

What suggestions did they make?

Did the group use those ideas? Why or why not?

Presenter

Name:

Did they follow their role? (yes/no)

What did they do well?

What suggestions did they make?

Did the group use those ideas? Why or why not?

Timekeeper

Name:

Did they follow their role? (yes/no)

What did they do well?

What suggestions did they make?

Did the group use those ideas? Why or why not?

Recorder

Name:

Did they follow their role? (yes/no)

What did they do well?

What suggestions did they make?

Did the group use those ideas? Why or why not?

ROLES AND RESPONSIBILITIES

Timekeeper

- Track time for project and activities
- Prompt the group to stay on task
- Participate in discussion
- Actively listen to ideas from team

Recorder

- Record notes for the group
- Participate in discussions
- Actively listen to ideas from team mates

Presenter

- Present information team strategy and products
- Speak for the team when asked questions
- Participate in discussions
- Actively listen to ideas from team mates

Team Leader

- Start and facilitate the conversation
- Remind team of the primary objective
- Participate in discussions
- Actively listen to ideas from team mates

PERSONALITY TYPES

The Caregiver

- Always eager to help and works to ensure that everyone has what they needs.

The Scholar

- Always ready to provide needed information. Gets straight the to point. Not especially nice, but not especially mean.

The Clown

- Always tries to lighten the mood and tell a joke no matter how bad things get.

The Ruler

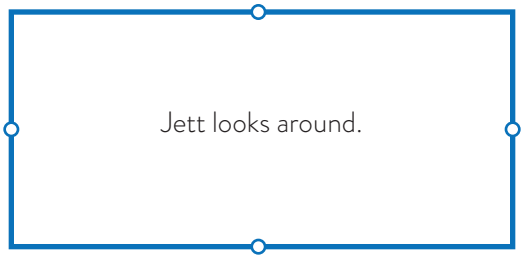
- Always wants to be in charge. Never asks questions, only gives orders.

PROJECT GRADING RUBRIC

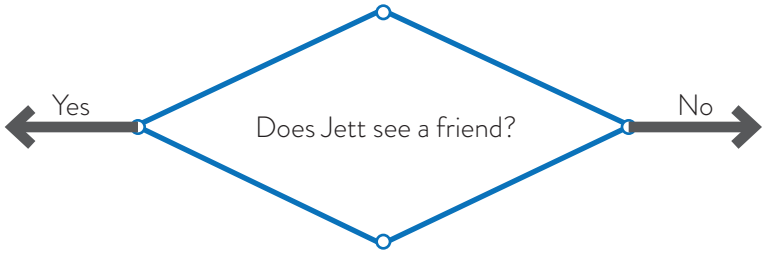
CATEGORY	4 of 4 [A]	3 of 4 [B]	2 of 4 [C]	1 of 4 [D]
Program	The program was able to complete the given task without user intervention once started.	The program was able to complete the given task with minimal user intervention once started.	The program was not able to complete the given task without user intervention.	The program was not able to complete the given task.
Use of Required Concepts	The submitted program demonstrated 5/5 of the required concepts.	The submitted program demonstrated 4/5 of the required concepts.	The submitted program demonstrated 3/5 of the required concepts.	The submitted program demonstrated less than 3 of the required concepts.
Documentation (Flowchart/SB3 File/Reflection Document)	The student submitted all requested documentation. The documentation was very well thought out, concise and logical.	The student submitted all requested documentation. The documentation was somewhat well thought out, concise and logical.	The student submitted all requested documentation. The documentation was not well thought, concise and logical.	The student did not submit all requested documentation.
Demonstration	The student was able to construct (or edit) a presentation with transitions, notes and bullet points describing their goal(s), process and challenges in programming the avatar or robot.	The student was able to construct (or edit) a presentation describing their goal(s), process and challenges in programming the avatar or robot.	The student was able to construct (or edit) a presentation but was unable to describe their goal(s), processes and/or challenges in programming the avatar or robot.	The student was not able to construct (or edit) a presentation describing their goal(s), process and challenges in programming the avatar or robot.

ABOUT FLOWCHART PIECES

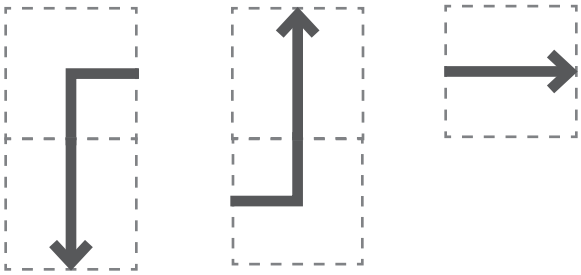
Every flowchart begins with a 'start' piece and ends with an 'end' piece to show that the flowchart is complete.



The 'process' pieces of the flowchart is where you would write a single "step" or "action". It is the most commonly used type of piece. Connecting multiple 'process' pieces together with arrows is how you create a 'sequence'.



The 'decision' pieces are used when two or more different paths can be taken. Not every flowchart will need a 'decision' piece. Typically, it will contain a question and have arrows (for example, 'yes' and 'no' which should be written next to the arrow) pointing to separate 'process' blocks.

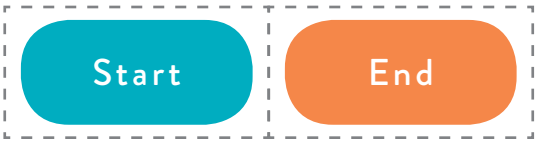


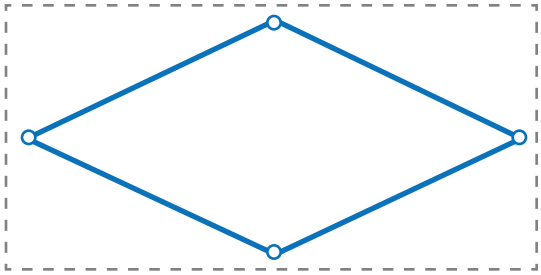
Arrows are used to point to the next step in the process - creating the 'flow' in flowcharts.

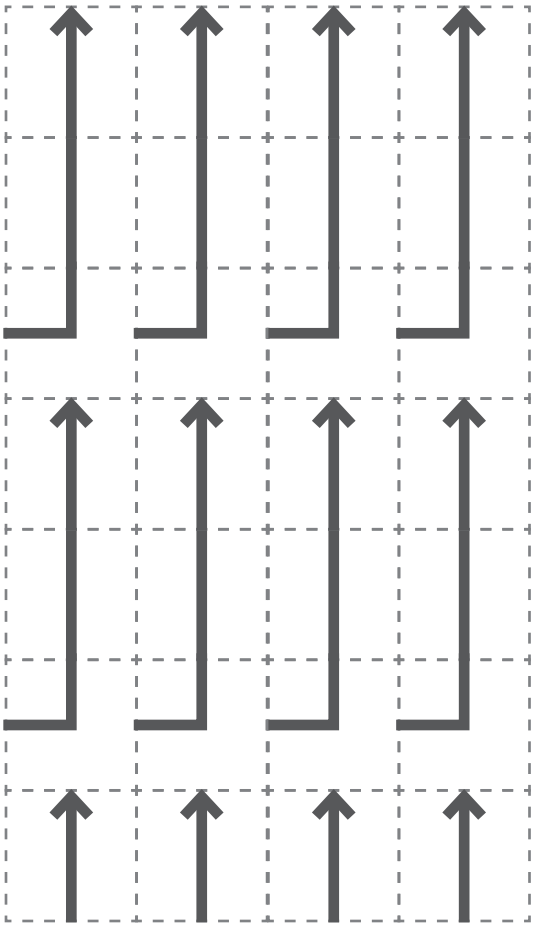
FLOWCHART KIT

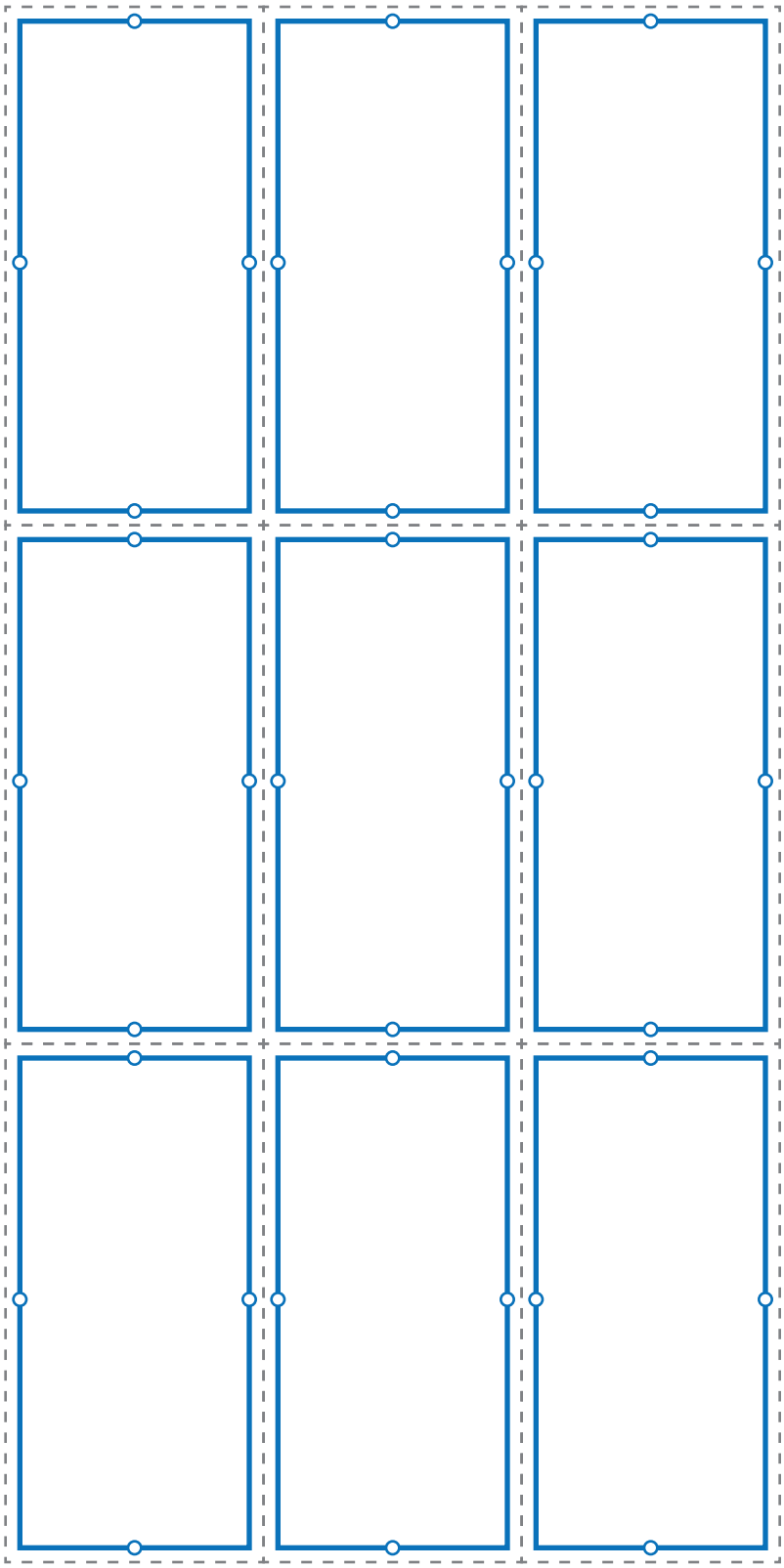
[Makes 1 flowchart.]

Please cut and paste the flowchart pieces to a sheet of paper or on the gridded worksheet provided to make your flowchart.

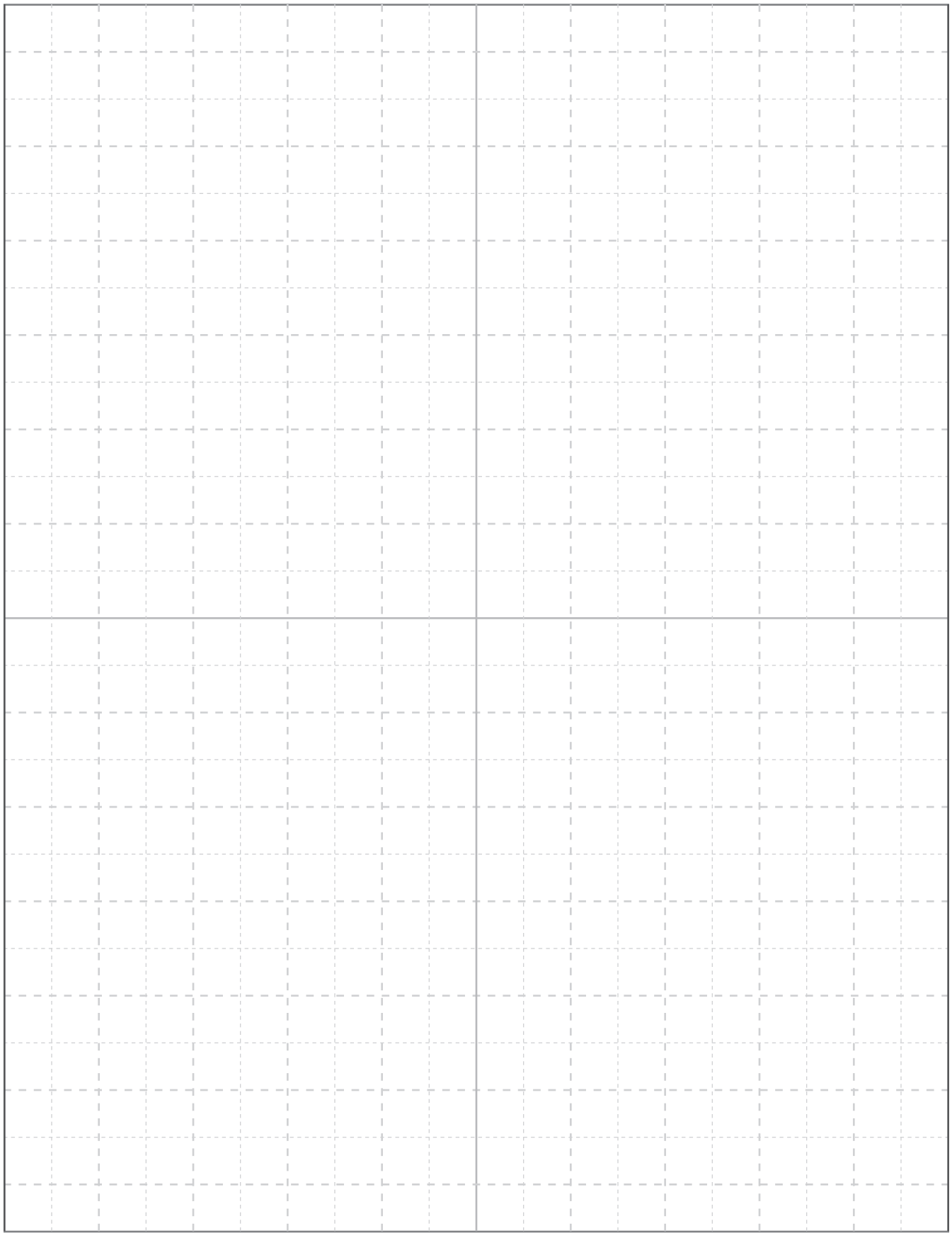








Name _____ Flowchart Title _____ Mission Number _____

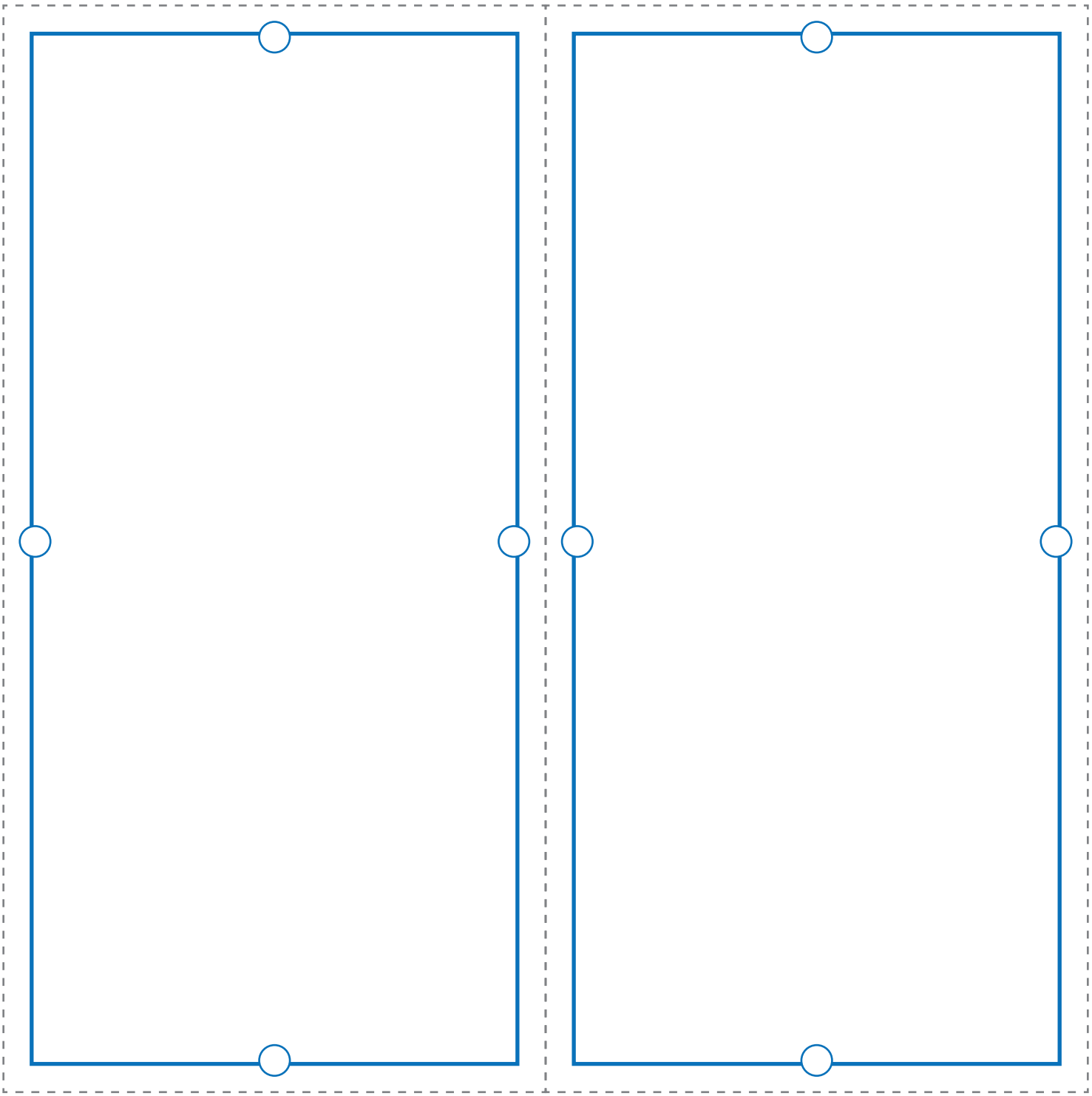


FLOWCHART PIECES (START/END)

Print, cut, and laminate these pieces in order to create reusable flowchart pieces that can be used with whiteboard markers. Back the pieces with velcro, magnets, or tape so you can display flowcharts on the wall.

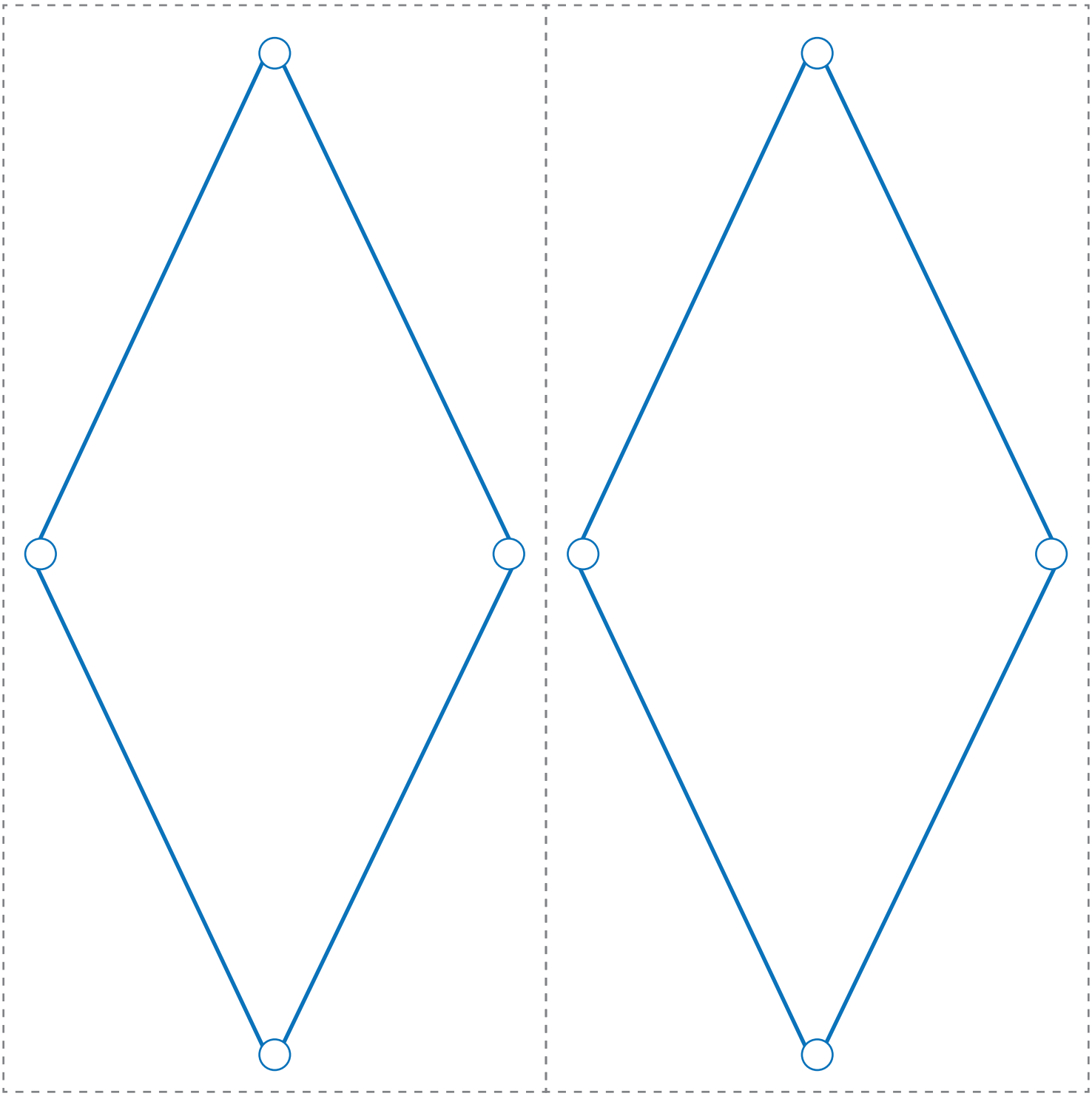
FLOWCHART PIECES (PROCESS)

Print, cut, and laminate these pieces in order to create reusable flowchart pieces that can be used with whiteboard markers. Back the pieces with velcro, magnets, or tape so you can display flowcharts on the wall.



FLOWCHART PIECES (DECISION)

Print, cut, and laminate these pieces in order to create reusable flowchart pieces that can be used with whiteboard markers. Back the pieces with velcro, magnets, or tape so you can display flowcharts on the wall.



FLOWCHART PIECES (ARROWS)

Print, cut, and laminate these pieces in order to create reusable flowchart pieces that can be used with whiteboard markers. Back the pieces with velcro, magnets, or tape so you can display flowcharts on the wall.