Seeker®

BY PITSCO EDUCATION

TEACHER'S GUIDE
UNIT 1

Unit Overview

In this Unit, students will be introduced to the Seeker robot and their role as Innovation Specialist working for a company named Adaptibots. Adaptibots is a small start-up company that aims to make a positive impact on the world through technology and robotics. The company currently operates in a garage, but the owners have dreams to expand their company with the help of the Innovation Specialist. This Unit is focused on orientation and training, similar to what a new employee would receive when starting a job. The Unit consists of two activities that should be completed in order.

In Activity 1, students are introduced to our two main characters, Sophia and Lucas, who are the founders of Adaptibots. They also learn about their position as Innovation Specialist and how they will adapt and apply Seeker to assist workers in different careers and industries. The content of Activity 1 focuses mainly on learning Seeker's hardware. They build Seeker into robot mode (humanoid form), engineer different attachments out of the accessory pieces, and load a couple different demo programs onto Seeker's controller to learn what Seeker is capable of doing.

Activity 2 centers around Seeker's software, especially its programming app and the programs that students will be writing for Seeker. After going through a tutorial of the app, students learn about planning a program by writing pseudocode and/or using a flowchart. They write their first program and then learn the differences between uploading a program using USB connection or Bluetooth. Students are also introduced to the simulator, where they can test their code in context of the real-world environment and situation tied to the Unit. Students are introduced to Seeker's debugging process as they learn to find and fix bugs as well as a troubleshooting process for dealing with hardware issues. The activity concludes with students exploring some games that can be played with Seeker.

Careers

- Chief Executive Officer
- Chief Technology Officer
- Innovation Specialist

Context and Setting

New employee training for a start-up robotics company named Adaptibots.

Computer Science and Technology Concepts

- Hardware and Software
- Innovation
- Systems and Systems Thinking
- Abstraction
- Types of Software
- Pseudocode and Flowcharts
- Simulation
- Impacts of Technology
- Seeker's Software Debugging Process
- Seeker's Hardware Troubleshooting Process
- Workplace and 21st-Century Skills

Unit Prep Instructions

- Determine how to pair students up for activities and utilize Seeker hardware. Each pair should have a Seeker set.
  - Consider giving student pairs different roles. For example, for Activity 1, have one student oversee hardware while the other student oversees software. Then, switch roles for Activity 2.
- Before starting this Unit, make sure the Seeker sets are fully disassembled and inventoried. Even if the kits have been used before, students should start as if they are brand-new sets.
- It will be important to rename your Seeker controllers for students to be able to connect and upload code using Bluetooth. See the *Seeker Getting Started Guide* for information on renaming controllers.
- The *Innovation Notebook* is a place for students to keep notes, brainstorm ideas, design adaptations for Seeker, record data from experiments, plan out programs, and answer questions. It can be distributed to students as a digital document or a paper-based workbook to be completed.
  - There are advantages and disadvantages to using the paper-based workbook versus the digital document. Besides saving paper, the digital document allows students to practice computer science skills such as managing digital files and versioning. You will need to provide instructions on where to save the digital version, as well as how to submit it for grading and review. Some tasks such as sketching, brainstorming, and making visual displays of data can be difficult with the digital document. The paper-based workbook allows students more freedom to be creative with their notebook entries. Sketching, for example, is much easier with pencil and paper. Keeping a paper-based workbook throughout the course could be a challenge for younger students who might lose their notebooks.
  - If you decide to use the paper-based *Innovation Notebook*, consider having students store their notebooks in a secure location in the classroom. This way, the notebooks never leave your classroom.
  - Even though students work in pairs, each student should complete their own *Innovation Notebook*; having students collaborate as they complete their notebooks is suggested.
- To complete the mini challenge near the end of the Activity 1, students will need to create an obstacle course. Have them utilize materials around the classroom to create the course. Seeker boxes, storage bins, existing wall, books, backpacks, and so on can all be used to create courses. Painter's tape can be used to mark out boundaries on the floor.
  - Rather than having each pair create their own course, have pairs work in larger teams to create a course.
  - Try to keep the course in a space about 6' x 6'. Bigger courses will take longer to create and longer to navigate.
- Seeker's batteries will need to be charged.
  - Consider collecting all the chargers and creating a charging station in your classroom.
  - Have the students charge Seeker's batteries at the end of the day. It generally takes a couple of hours to fully charge a dead battery.
  - Leaving a battery on a charger overnight ensures they will be ready for the next day, but it is suggested to not leave them connected to chargers over multiple days such as a weekend or holiday breaks.
  - The battery charger has an LED that shines red while charging and turns green when the battery is fully charged.

Guide to Activity 1: Seeker Hardware

Page 1 – Welcome to Adaptibots

Students are introduced to Sophia and Lucas, our two main characters. They are also introduced to Adaptibots and their role in the company as Innovation Specialist. They open their Seeker kits and begin exploring some of the hardware in the kit.

Tasks to Complete

- Investigate the parts of their Seeker kit and discuss their potential uses with a partner.

Teaching Tips

- Before starting this Activity, make sure the Seeker sets are fully disassembled and inventoried. Even if the kits have been used before, students should start as if they are brand-new sets.

Notes:

Icon Key

- Code Saved to Student Profile
- Code Testing Task
- *Innovation Notebook* Task
- Engineering Task
- Guided Programming Task
- Exploration Task
- Sequencing Task
- Debugging/ Troubleshooting Task
- Challenge Task

Page 2 – *Innovation Notebook*

Sophia discusses the importance of using technology for good and how they want to impact the world to make lives better. The students' job as Innovation Specialist will be to investigate how Seeker can be used in different careers and industries to meet needs. Students are introduced to the *Innovation Notebook* and complete a series of questions related to what needs/problems, careers, and tasks Seeker might be a good fit for.

Tasks to Complete

- Answer questions in the *Innovation Notebook*. See page 11 for the *Innovation Notebook* Answer Key for this task.

Teaching Tips

- Make sure students can access their *Innovation Notebook*. See the Unit Prep Instructions for more information on *Innovation Notebooks*.

Notes:

Page 3 – A Humanoid Robot

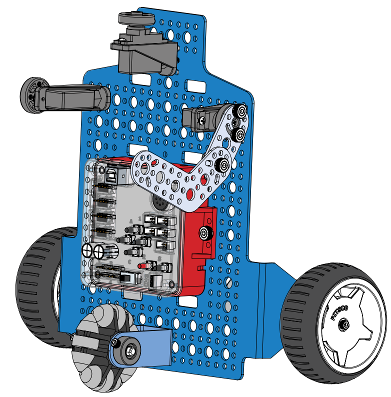
Lucas discusses tips for building using Seeker's components and parts including the two methods for fastening parts together. Students are presented with their first *Technical Guide* to construct Seeker into robot mode with a head and arms.

Tasks to Complete

- Begin constructing Seeker into its robot form by following Steps 1-5 of the *Technical Guide*.

Teaching Tips

- The *Technical Guide* is presented to students as a link to an interactive web document with animated videos and pop-up tips. This guide can also be presented to students as a digital PDF or as a printed guide. The digital PDF and printed guide will not include the animated videos. If you decided to provide a printed version of the guide, you will need to print the PDF.
- During construction, students should pay close attention to which holes are used for screws, nuts, and hexlock fasteners.
- The completed build should look such as the image shown at right.



Notes:

Page 4 – Connecting Servos

Students continue the process of building Seeker into robot mode. Lucas discusses the difference between DC motors and servo motors and then guides students through connecting the servos that control Seeker's head and arms to the controller.

Tasks to Complete

- Connect servos to the servo ports on Seeker's controller.
  - Port 1 – Head
  - Port 2 – Seeker's right arm
  - Port 3 – Not used
  - Port 4 – Seeker's left arm

Teaching Tips

- When connecting servos to the servo ports, it matters which way the wires are connected. Make sure the black wire is towards the notch (left side) in the port. If students find that their servos aren't working, this is usually the first thing to check.

Notes:

Page 5 – Centering the Servos

Before attaching Seeker's arms and head, the servos need to be centered. Lucas discusses how to get the servos to move to their center position using a program. Students walk through the process of connecting Seeker to the app via USB cable and then uploading a program. When the program is run on Seeker, the servos move to their center positions.

Tasks to Complete

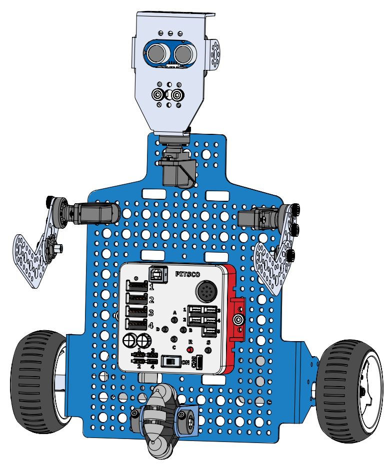
- Upload and run a simple program to center the servos.

Teaching Tips

- A Lode Code button is provided for students below the video to load the correct Blockly file.
- Students can upload programs to Seeker using the USB cable or a Bluetooth connection. For this activity, only USB connections should be used. Bluetooth connections will be discussed in Activity 2.
- When any program is run on Seeker, the servos connected to the controller will automatically rotate to their center positions before the code executes. This functionality is built into Seeker's firmware so the servos always start from a consistent center position.

Notes:

Page 6 – Attaching Seeker's Head and Arms



Using the provided *Technical Guide* (PDF), students complete the assembly of Seeker's robot mode by attaching the head and arms to the servos.

Tasks to Complete

- Complete the assembly of Seeker into robot mode.

Teaching Tips

- During construction, students should pay close attention to which holes are used for screws, nuts, and hexlock fasteners.
- The completed build should look such as the image shown at right.

Notes:

Page 7 – Hardware Exploration

Sophia and Lucas encourage the students to explore some of the hardware components in the set by engineering different attachments for Seeker. Suggestions include a pencil holder, phone holder, bumper, or even a hat. Students are also encouraged to build their own ideas. Students can check off attachments they build in their *Innovation Notebook*.

Tasks to Complete

- Explore Seeker's hardware components by engineering various attachments.
- When finished, return Seeker to its robot mode by removing all attachments.

Teaching Tips

- A checklist of attachment ideas is provided in the *Innovation Notebook*. Students can check off attachments they build, or they can build their own attachments. See page 11 for the *Innovation Notebook* Answer Key for this task.
- Give students sufficient time to explore the pieces of the set and how they can be used.
- Students should work in pairs.
- Consider using this activity as a think-pair-share type activity, where students share one attachment they created with the whole class.

Notes:

Page 8 – Controller Overview

Lucas explains the controller ports for connecting other hardware and controller features such as the speaker, buttons, and pixels. Students upload a provided demo program that helps them explore many of Seeker's capabilities. Students follow instructions provided in their *Innovation Notebook* as well as record observations about the demo program.

Tasks to Complete

- Load a demo program on to Seeker.
- Run the demo program and record observations in their *Innovation Notebook*.

Teaching Tips

- Students will need a space on the floor about 6' x 6' to test Seeker's demo program.
- Make sure students follow the "Do" instructions in their *Innovation Notebook*. See page 12 for the *Innovation Notebook* Answer Key for this task.
- When running the program, the default "waiting" mode is when the pixels flash yellow on the controller. At this point, pressing any of Seeker's black buttons will cause it to do something.
  - Button A cycles through different pixel animations.
  - Button B cycles through different sound effects.
  - Button C enters sensor mode that uses measurements from the Ultrasonic Sensor to change Seeker's arm positions and pixel colors.
  - Button D enters encoder mode. As the wheels are rotated, encoder measures change the pitch of the sound from Seeker's speaker.
  - The Start button cycles through different driving patterns.

- When pressing a button on the controller, students might need to hold the button down for about a half second to let the blink cycle of the pixels finish before the button press can be registered by the controller.
- Make sure students place Seeker on the ground in the designated test space before pressing the Start button to cause Seeker to drive in various patterns.

Notes:

Page 9 – Demo Program

After students explore the demo program on their own and record their observations, Lucas explains the different functions of the demo program in more detail, showcasing some of what Seeker can do. This includes explaining pixel animations, sounds, using sensors and encoders, and its navigational abilities.

Tasks to Complete

- Continue to explore the demo program in more detail.

Teaching Tips

- Make sure students place Seeker on the ground in the designated test space before pressing the Start button to cause Seeker to drive in various patterns.
- Have students turn Seeker off when it's not in use to save its battery.

Notes:

Page 10 – Systems and Subsystems

Sophia and Lucas discuss systems, subsystems, and using systems thinking as an approach to solve problems and meet goals. Then, students complete an activity in their *Innovation Notebook* to categorize Seeker's parts and components by system or subsystem. A system web is created as students draw lines or arrows between subsystems and components that affect each other.

Tasks to Complete

- Create a Seeker system/subsystem web in their *Innovation Notebook*.

Teaching Tips

- Encourage students to discuss each component and what system/subsystem it belongs to. Answers can vary as long as students can justify their answer.
- See page 13 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 11 – Abstraction

Lucas discusses the concept of abstraction as it relates to computer science. After giving some examples of abstraction in everyday life, an example program is given to students that demonstrates the concept. This program allows students to "enter" a path for Seeker to follow using the controller buttons. Then, after pressing the Start button, Seeker drives the stored path.

Abstraction: simplifying a task by hiding the complexity of how something works by focusing on what is important

Tasks to Complete

- Use the stored path driving demo program to input a path into Seeker and then press the Start button to execute the path.

Teaching Tips

- This activity works best if the student enters the path from behind the robot. This way, the buttons on the controller correspond to the direction Seeker will travel:
    - Button A – Backward 30 cm
    - Button B – Turn left
    - Button C – Forward 30 cm
    - Button D – Turn right
    - Start Button – Drive the stored path
- Have students use the USB cable to connect Seeker and upload the program even if they know how to use Bluetooth. Due to the size of the program, uploading using Bluetooth will take much longer.

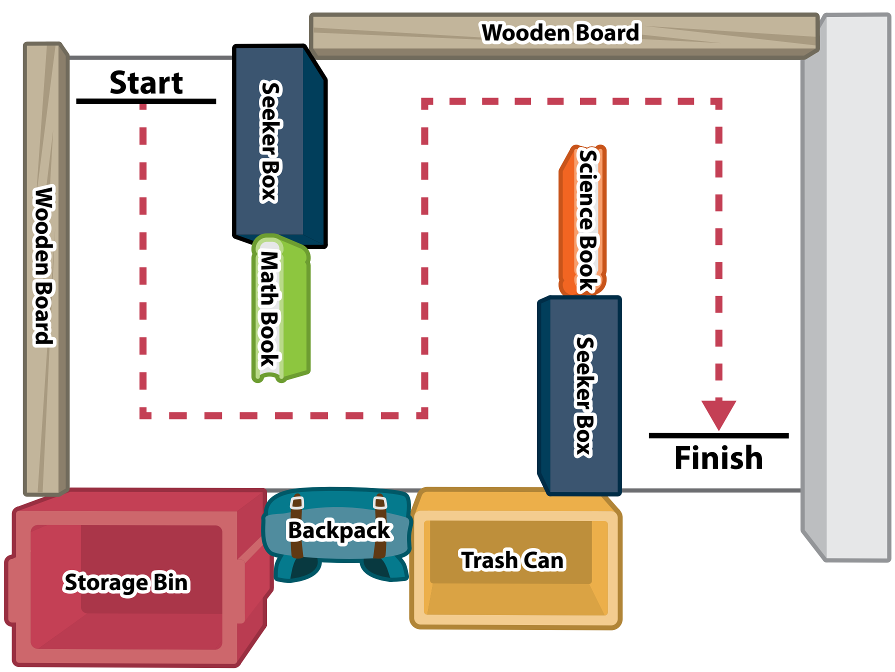
Notes:

Page 12 – Obstacle Course Mini Challenge

Lucas and Sophia give students a mini challenge to create an obstacle course for Seeker and then use the stored path driving demo program to complete the obstacle course.

Tasks to Complete

- Create an obstacle course for Seeker.
- While running the stored path driving demo, input the correct path into Seeker to complete the course.



Teaching Tips

- Rather than having each pair create their own course, have pairs work in larger teams to create a course.
- Utilize whatever materials you have in the classroom to create the course(s). Items such as boxes, books, trash cans, backpacks, and so on can all be used to create barriers. Students can also mark out a course using painter's tape on the floor.
- Try to keep the course in a space about 6' x 6'. Bigger courses will take longer to create and longer to navigate.
- Students can use the Notes section of their *Innovation Notebook* to record the button sequence for completing their course.

Notes:

Page 13 – Hardware Reflection

Sophia and Lucas wrap up the activity by having students reflect on what they have learned in the activity. In their *Innovation Notebook*, students list the three most important things they learned in the activity as well as three unanswered questions they have about Seeker. If time permits, students can share what they've learned and the questions they still have with a partner or as a class.

Tasks to Complete

- Reflect on the activity.
- Record three things they've learned in their *Innovation Notebook*.
- Record three questions they still have in their *Innovation Notebook*.

Teaching Tips

- Reflection is an important part of the learning process. Help students avoid the temptation of rushing through or skipping this important task.
- See page 15 for the *Innovation Notebook* Answer Key for this task.
- If time permits, discuss students' reflection thoughts as a whole class.

Notes:

Unit 1

Activity 1
Innovation Notebook

Answer the following questions.

How can Seeker be used to meet human needs or solve human problems?

Answers will vary depending on students' ideas and perceptions of Seeker.

What industries or careers would Seeker be a good fit for?

Answers will vary depending on students' ideas and perceptions of Seeker.

What specific tasks in these careers could Seeker do?

Answers will vary depending on students' ideas and perceptions of Seeker.

Hardware Exploration

Exploration Ideas:

- ❏ Add an attachment that holds your pen or pencil.
- ❏ Create an attachment to hold a phone. You could then use the camera from the phone and a video call to another phone to see a first-person view of what Seeker sees.
- ❏ Create a bumper on the front of Seeker that collects and pushes items around as Seeker navigates.
- ❏ Enhance Seeker's looks by adding pieces that make it look more human . . . a hat, for example.
- ❏ When finished, remove any pieces or attachments you've added and return the robot to the state it was in before starting this exploration task.

Controller Overview

With the controller overview program running, follow the instructions and answer the associated questions. **Note:** When pressing buttons, it will be helpful to press and hold the button down for about half a second for the button press to register.

Do: With the yellow lights flashing on Seeker, press Button A until it beeps. Observe what happens. With the yellow lights flashing, try pressing Button A several more times.

Question: *In general, what does pressing Button A do in the demo program?*

In general, pressing Button A toggles through and plays different LED pixel animations on Seeker's controller.

Do: With the yellow lights flashing, press and hold Button B. Observe what happens. Continue to press Button B multiple times.

Question: *In general, what does pressing Button B do in the demo program?*

In general, pressing Button B toggles through and plays different sound effects through Seeker's speaker.

Do: With the yellow lights flashing, press and hold Button C until Seeker beeps. The robot is now in sensor mode. Move your hand back and forth at different distances in front of its face and observe what happens.

Question: *What does Seeker do as your hand gets closer to its face? What about as your hand gets farther away?*

Students should notice, as their hand gets closer to Seeker, the robot's arms raise higher and higher. The farther away their hand is, the lower the arms go. Students should also notice that Seeker's pixel LEDs change color depending on the distance their hand is from its face.

Do: Press any of the four buttons in the middle to exit sensor mode. With the yellow lights flashing, press Button D until Seeker beeps. Next, turn either or both of the robot's wheels. Observe what happens.

Question: *What does Seeker do as you turn the wheels? What if you turn the wheels in the other direction? Do both wheels do the same thing?*

Students should hear the robot playing different musical notes as the wheels are turned. If the wheels turn one way, the notes get higher. If the wheels are turned the other way, the notes get lower. The left wheel changes the pitch by a half step. The right wheel changes the pitch by a whole octave.

Do: Press any of the four buttons in the middle to return to the yellow flashing lights. Unplug the USB cable from Seeker and place the robot on the ground in an open area. Press the Start button and observe what happens. Try pressing the Start button several more times.

Question: *In general, what does Seeker do when the Start button is pressed?*

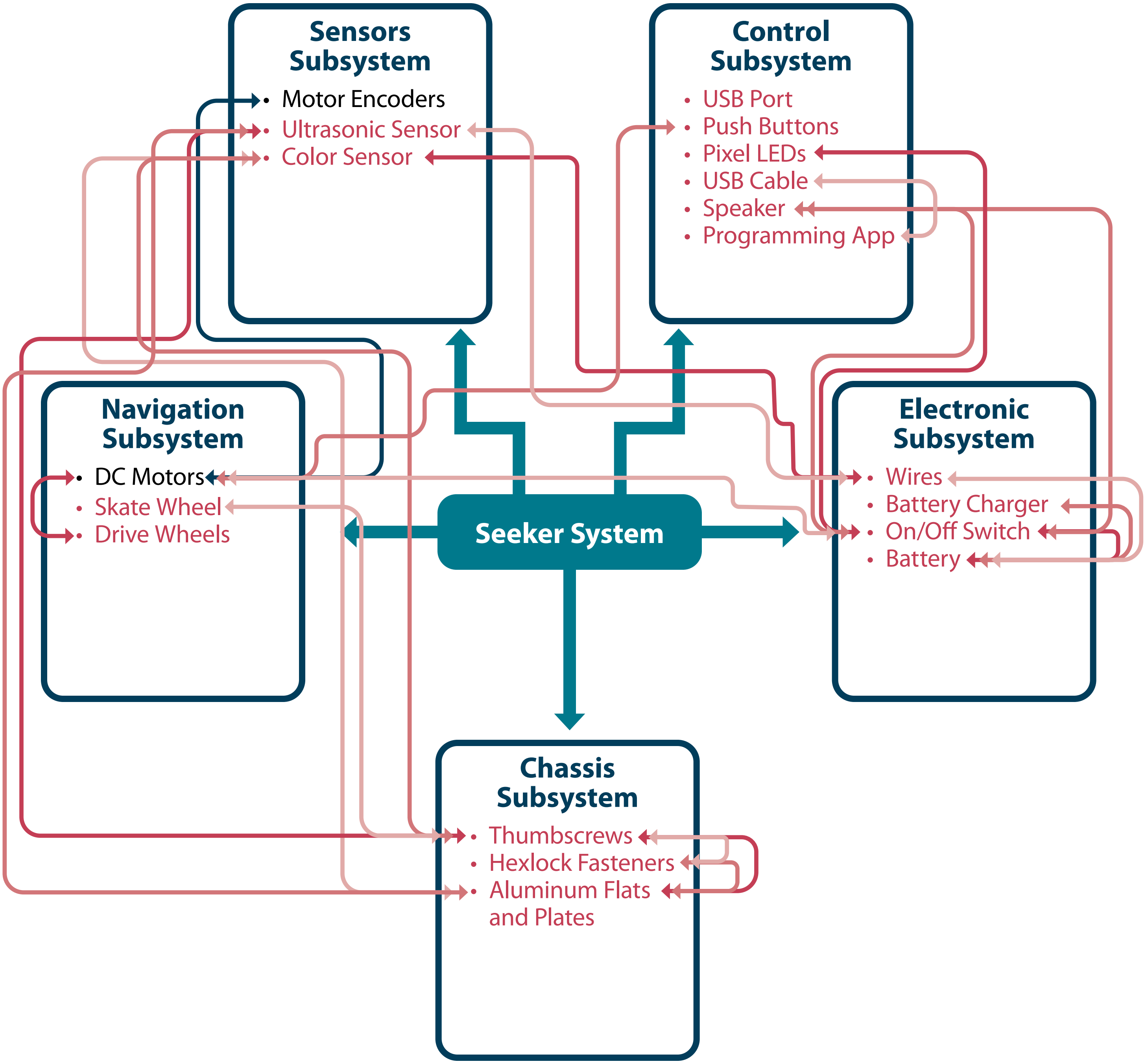
In general, the robot will drive in different patterns when the Start button is pressed. It starts with a single line and a single 360-degree turn. It then drives down and back, then in a triangle, then a square, and finally a pentagon before repeating the pattern.

Systems and Subsystems

The following lists several of Seeker's parts and components. Determine which of the robot's subsystems these parts belong to. Write each part in one of the subsystem boxes in the graphic. Motor encoders and DC motors have already been categorized for you as examples.

Seeker's Components:

- USB Port
- Ultrasonic Sensor
- Aluminum Flats and Plates
- Push Buttons
- Skate Wheel
- Wires
- Battery Charger
- Pixel LEDs
- Drive Wheels
- On/Off Switch
- Thumbscrews
- USB Cable
- Hexlock Fasteners
- Speaker
- Programming App
- Color Sensor
- Battery



Systems and subsystems are a network or web of interconnected parts. In the graphic, after you have listed the parts and components in the subsystem boxes, draw lines to connect components to show how they are connected and affect each other. For example, motor encoders and DC motors affect each other, so a line has been drawn to show this relationship. When you are finished, you should have a web of interconnected parts that make up the Seeker system.

Answers can vary because components could be categorized in different subsystems. However, students should be able to justify their thinking as to why they placed a component where they did. The connections between components can be quite complex as students realize how each component impacts others. Again, students should be able to justify their reasoning.

Hardware Reflection

Reflect on what you've learned so far. In the following space, list the three most important things you've learned so far in your training as Innovation Specialist.

Answers will vary.

While it can be tempting for students to breeze through reflection time, it's important for student to evaluate their own learning. Encourage students to spend some time thinking about what they've learned so far. If time permits, have students discuss what they've learned with a partner, in small groups, or as a class.

What are three unanswered questions you currently have about Seeker?

Answers will vary.

Encourage students to be curious and ask questions. Again, if time permits, have students share their questions with a partner, in small groups, or as a class. During discussion, avoid answering student questions. Instead, encourage them to seek the answers for themselves as they go through the other Seeker activities and Units.

Notes:

Notes:

Guide to Activity 2: Seeker Software

Page 1 – Welcome to Adaptibots

Lucas and Sophia discuss Seeker's three main types of software: Seeker's program, Seeker programming app, and Seeker's firmware. Students learn how computer or robotic software and hardware must work together to accomplish meaningful tasks. They also learn how instructions, or code, written in a programming language gets translated into binary computer code that affects millions of transistors that then control hardware parts.

Tasks to Complete

- Learn how software and hardware work together to accomplish meaningful tasks.

Teaching Tips

- Before starting this Activity, make sure the Seeker sets are fully disassembled and inventoried. Even if the kits have been used before, students should start as if they are brand-new sets.

Page 2 – *Innovation Notebook*

Sophia discusses the importance of using technology for good and how they want to impact the world to make lives better. The students' job as Innovation Specialist will be to investigate how Seeker can be used in different careers and industries to meet needs. Students are introduced to the *Innovation Notebook* and complete a series of questions related to what needs/problems, careers, and tasks Seeker might be a good fit for.

Tasks to Complete

- Answer questions in the *Innovation Notebook*. See page 11 for the *Innovation Notebook* Answer Key for this task.

Teaching Tips

- Make sure students have completed Unit 1 Activity 1 before starting this activity.
- This activity will be the first activity where students write and upload programs to Seeker.
- If students each have a device, there are a couple options for how students navigate content and write programs while being paired up. No matter which option is used, we suggest having students log into their own accounts on their own devices.
  - The Full Collaboration Option: Students fully collaborate using one computer/device to program and the other computer/device to view the content in full screen. One student logs on to each device and, although the content is only viewed on one of the devices, both students page through the content so progress through the activity is tracked for both student accounts. This option has these benefits:
    - Student pairs can view the content full screen without having to switch between viewing the content and the programming area.
    - Student pairs can utilize team programming to write their programs.
    - Student pairs navigate through the content at the same pace.

Icon Key

- Code Saved to Student Profile
- Code Testing Task
- *Innovation Notebook* Task
- Engineering Task
- Guided Programming Task
- Exploration Task
- Sequencing Task
- Debugging/ Troubleshooting Task
- Challenge Task

- The Partial Collaboration Option: Each student logs into their own computer/device and works independently to view the content and write programs. When it is time to upload the program to Seeker, students collaborate and choose one computer to upload the program. If desired, the program from the other computer/device can be uploaded and tried after the first. This option has benefits as well:
  - Each student writes their program and stays engaged rather than having one student do all the programming.
  - Students can answer quizzes and checkpoints individually.
  - Students can write their preferences into their program such as what colors or sounds to use.

Notes:

Page 2 – App Overview

Lucas gives students a quick tutorial on the features and capabilities of the programming app. Topics covered include different ways to view content, access to resources, using menus, and building programs using blocks and Python®.

Tasks to Complete

- Complete a tutorial on using the tools and features of the programming app.

Teaching Tips

- Students should be somewhat familiar with the app from completing the first activity.

Notes:

Page 3 – Scavenger Hunt

Sophia gives students the challenge of completing a scavenger hunt to further explore the features of the app. The scavenger hunt is driven by a list of questions in the *Innovation Notebook*. As students discover the answer to each question, they record their answers in their notebooks.

Tasks to Complete

- Complete a scavenger hunt in the app by recording answers to a list of questions in the *Innovation Notebook*.

Teaching Tips

- Students should collaborate in pairs to complete the scavenger hunt.
- If time permits, go through the answers to the scavenger hunt as a class to make sure students are all familiar with the tools and features of the app.
- See page 26 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 4 – Planning a Program

Lucas talks to students about the importance of planning before writing a program. Planning can include writing pseudocode (fake code) or creating a flow chart to organize thoughts and ideas. After looking at some example pseudocode and a flow chart for using Seeker as a firefighter robot, students use their *Innovation Notebook* to write their pseudocode and create their flowchart for making their favorite sandwich.

Tasks to Complete

- Write pseudocode for making their favorite sandwich in their *Innovation Notebook*.
- Create a flowchart for making their favorite sandwich in their *Innovation Notebook*.
- See page 27 for the *Innovation Notebook* Answer Key for this task.

Teaching Tips

- Writing pseudocode and organizing steps of a process in a flowchart are important concepts for creating algorithms and sequences in computer programming. Help students focus on some of the details of their sandwich making process.
- If time permits, consider using student pseudocode or a flowchart to make one of the sandwiches. Following the steps precisely usually shows that information or instructions were left out. Discuss how computers can't interpret or read between the lines if instructions are missing. This is a common problem when coding.
- Another idea is to have students swap pseudocode and read through the instructions. They can act out making the sandwich to see what steps were omitted.
- Discuss what level of detail is needed when planning a program.

Notes:

Page 5 – First Program

Lucas guides students through creating their first program in the app. It's a simple program that demonstrates the programming building process including how every program needs to start with a **program start block**. After building the program, students save the program to their profile.

Tasks to Complete

- Write the program U1A2-FirstProgram and save it to their profile.

Teaching Tips

- There are several methods students can use to submit the programs they create for grading. This includes saving programs to a shared drive, submitting via email, or using the assignment feature in the teacher management tools. Determine which method you prefer to use and instruct students on the necessary steps they need to take.
    - **Note:** It's OK to not have them submit every program they write for grading or review.

Notes:

Page 6 – Bluetooth Connection

Lucas and Sophia discuss the pros and cons of using Bluetooth versus USB when connecting and uploading code to Seeker. Bluetooth is wireless and allows for more mobility while USB is faster but requires Seeker to be connected by a cable. Students go through the Bluetooth connection process and upload the program they just wrote. When the upload is complete, they run the program.

Tasks to Complete

- Connect and upload the first program code using Bluetooth.
- Run the program.

Teaching Tips

- Make sure Seekers are renamed and labeled with their names so when students try to connect their robots, they know which one is theirs.
- When connecting via Bluetooth, it is possible for one student group to upload code to another group's robot. Whether done accidentally or intentionally, it can cause confusion and distractions. Encourage students to upload code only to their robot.
- Mac devices sometimes have a hard time recognizing the Bluetooth signal of Seeker robots. If, after trying to connect through Bluetooth and the pop-up box for selecting the correct Seeker robot just says, "Searching," press the red Reset button on Seeker. After Seeker resets, its name should appear in the connection pop-up box.

Notes:

Page 7 – Seeker Simulation

Sophia talk students through the benefits of using the simulator to test code on Seeker before uploading code to the physical robot. Some of these benefits include running quick tests on code without having to set up special scenarios in the classroom, seeing Seeker complete tasks in an environment similar to the real-world situation that Seeker is being programmed for, and following a process that a real robotics engineer would follow when developing a prototype robot. Students make changes to their first program code and test their changes in the simulator.

Tasks to Complete

- Learn how to use the simulator to test code.
- Test changes to the First Program code in the simulator.
- Discuss how the robot behaves in the simulator compared to the real world.

Teaching Tips

- Students should collaborate in pairs to complete the scavenger hunt.
- If time permits, go through the answers to the scavenger hunt as a class to make sure students are all familiar with the tools and features of the app.
- See page 28 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 8 – Impacts of Technology

Sophia discusses the impacts that technology can have on society and how the technology itself is neither good nor bad but that it is how people use technology that can have positive or negative impacts. Technology use always involves trade-offs where benefits and drawbacks must be balanced. In their *Innovation Notebook*, students brainstorm possible trade-offs of using robotic or computer technology including potential positive and negative impacts.

Tasks to Complete

- Brainstorm trade-offs of using robotic or computer technology and identify potential positive and negative impacts in their *Innovation Notebook*.

Teaching Tips

- This activity can be done in student pairs, small groups, or as a whole class.
- If students struggle coming up with ideas, rather than giving answers, ask them leading questions to help guide their thinking and discussion.
- See page 29 for the *Innovation Notebook* Answer Key for this task.

Notes:

Page 9 – Learning from Failure

Sophia and Lucas discuss trade-offs they encountered while developing Seeker and how learning from failure is part of the development process. Sophia assures students that failure is normal and innovation requires perseverance and determination to keep trying. To demonstrate failure, students are given pseudocode and a program from Sophia to use Seeker as a stoplight. However, the program has a problem and doesn't work quite right. Students upload the program to Seeker and run it to see what the problem is.

Tasks to Complete

- Upload and run the Stoplight program on Seeker to determine what the problem is.

Teaching Tips

- Although loops have not been explained, students should see from the pseudocode how the **forever loop block** repeats the actions of changing the pixel colors. Loops will be explained in more detail in other Units.

- You might need to help students identify that there is a problem with the program. Seeker's pixels never turn red.
- The problem with the Stoplight program is a common error when using loops. Seeker must be given time to complete all its actions before the loop repeats.

Notes:

Page 10 – Seeker's Debugging Process

Sophia discusses the process of troubleshooting and debugging code. A debugging process is presented that students can use to think through, identify, and fix errors in the programs they created. Students use the debugging process to fix the Stoplight program, upload the fixed code to Seeker, and test the solution.

Tasks to Complete

- Use the debugging process to fix the Stoplight program.
- Upload and test the fixed program on Seeker.

Teaching Tips

- The debugging process is given to students in the Appendix of their *Innovation Notebook*.

Notes:

Page 11 – Seeker's Troubleshooting Process

Sophia and Lucas discuss Seeker's hardware troubleshooting process. This process is presented as a series of flowcharts students can work through to help identify the problem and how to fix it. One of the most common hardware issues is a low battery. When the battery is low, a red indicator LED illuminates near the controller's USB port. When this LED turns on, it is time to charge the battery.

Tasks to Complete

- Investigate Seeker's troubleshooting process and how to determine when Seeker's battery is low.

Teaching Tips

- Seeker's troubleshooting process is given to students in the Appendix of the *Innovation Notebook*.
- Seeker's low-battery indicator light will only shine while a program is running on Seeker. When the program stops, the LED will turn off.
- The most common hardware issues with Seeker are:
    - Low battery.
    - Plugging servos in backward (black wire goes toward the notch).
    - Sensors or servos plugged into the wrong sensor or servo ports.
    - Forgetting to invert a servo or motor (if needed).

Notes:

Page 12 – Workplace Skills

Sophia talks about the importance of developing workplace skills, soft skills, or 21st-century skills. Students read through a list of these skills in their *Innovation Notebook* and complete a personal assessment to rate themselves on each skill listed. Students then identify their strengths and weaknesses. They are encouraged to utilize their strengths when working on projects for Adaptibots and to look for opportunities to grow in their weaknesses.

Tasks to Complete

- Complete a workplace skills assessment in their *Innovation Notebook*.

Teaching Tips

- Remind students to be honest with themselves as they complete the workplace skills assessment.
- Encourage students to set personal goals of growing in all skills, but especially those skills where they are not as strong.
- Students will reevaluate themselves in later activities to see how they have grown in these skill areas.

Notes:

Page 13 – Seeker Games

Lucas introduces students to three games that were developed to play using Seeker. React is a two-person game to see who has the fastest reflexes. Secret Code works like the game Simon Says, where a pattern is given and students repeat the pattern using Seeker's buttons. The pattern gets longer with each correct round. Obscurity is a game with few instructions that students have to figure out. The game is completed by pressing Seeker's buttons, turning Seeker's wheels, triggering the Ultrasonic Sensor, or a combination of these events.

Tasks to Complete

- Play games using Seeker to further explore Seeker's capabilities.

Teaching Tips

- Encourage students to save the games to their profile so they can play them whenever they have some extra time. Later, as students become more familiar with coding, they can experiment with the games by changing some of the parameters in the program to customize how the games are played. For example, the React game could become a four-player game.
- Consider having classroom tournaments with the React game or the Secret Code game. Or, keep track of top scores to make the games more fun.

Notes:

Page 14 – Wrap Up

Sophia wraps up the orientation for Seeker. Students are reminded of their goal to find new ways to help people in various jobs and careers by adapting Seeker to meet needs and solve problems.

Tasks to Complete

- Complete the activity.

Teaching Tips

- Try to keep this Unit available for students in their classroom dashboard so they can review this material at any time.

Notes:

Activity 2
Scavenger Hunt

#	Find the answer to these questions:	Answer
1	Find the **wait block**. Which category is it in?	The Control category.
2	Find the s**ervo get position block**. servo 1 get position How many servos are listed in the drop-down list?	There are four servos listed. These correspond to the four servo ports on Seeker's controller.
3	Find your grade for this class in the grade book. What is your percentage?	Most likely it will be 0% at the time students complete this activity.
4	After finding your grade, find the Home button to return you to the programming section of the app. Where is Home (Home) located?	Home is in the drop-down menu under the student's username in the top-right corner.
5	Find the **stop sound block** and put your cursor over it without clicking it. A pop-up box called a Tool Tip will appear. What does the Tool Tip say?	The Tool Tip says, "Stop any current sound from being played."
6	Try to throw a block from any category in the trash can. What has to happen for the block to go into the trash can?	While holding the block, the cursor must be over the trash can and the lid must open for the block to go into the trash.
7	Switch to Python mode and type, "# hello Seeker" next to line one. What color is the word *hello*?	After typing the # symbol, the text should turn green.
8	Find the **square root block**. What category is it in?	The **square root block** is in the Math category.
9	Change the view to split view. Drag a **Program End block** into the programming area. What is the corresponding Python code for the **Program End block**?	The Python code for the **Program End block** is **skr.End()**.
10	Find the **show animation block** from the Pixels category. What is the last animation available in the drop-down list?	The last animation available is rainbow sparkle.

Making a Toasted Peanut Butter and Jelly sandwich

Pseudocode

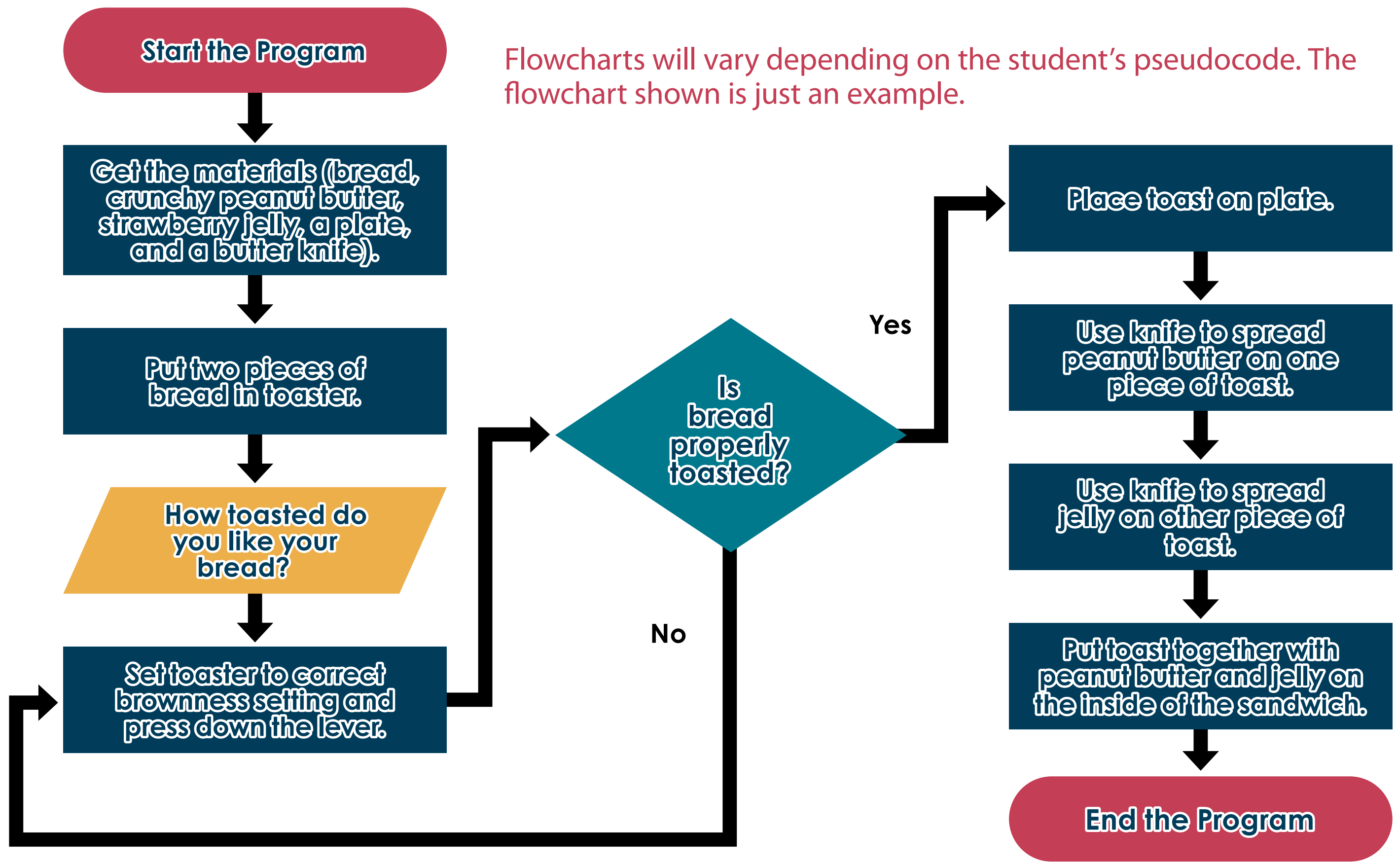
Write the steps for making your sandwich in normal English.

Answers will vary due to the nature of writing pseudocode and based on the type of sandwich being made. Pseudocode should be written in a way that makes sense to its creator. It can be simple with broad instructions or complex with many details. Shown here is an example of what students might create.

1. Get out the bread, crunchy peanut butter, strawberry jelly, a plate, and a butter knife.
2. Put two pieces of bread in the toaster.
3. Ask how toasted to make the bread.
4. Set the toaster to the correct brownness setting and press down the lever to start toasting the bread.
5. When the toaster pops up, make sure the bread is properly toasted. If not, press down the lever for a few more seconds.
6. Place the toasted bread on the plate.
7. Use the butter knife to spread layer of peanut butter on one piece of toast. When finished, clean off the knife.
8. Use the butter knife to spread a layer of jelly on the other piece of toast. When finished, put the knife in the dish washer.
9. Put the two pieces of toast together with the peanut butter and jelly sides together.

Flowchart

Create a flowchart to represent your pseudocode.



Flowchart Key

Symbol	Name	Function
	Terminator	Represents the start or end of a process.
	Process	Represents a process or a step in a process.
	Decision	Represents a decisions that needs to be made.
	Parallelogram	Represents input/output operations for data.
	Arrow (Flow Line)	Represents the flow between steps.

Impacts of Technology

Identify trade-offs related to using robotic and computer technology. List the potential positive and negative impacts related to the trade-offs.

Completion of the following chart can vary. Some possible example trade-offs along with positive and negative impacts are listed. Students may list these as well as other ideas.

Trade-off	Positive Impacts	Negative Impacts
Automation	Robots can do boring repetitive jobs more quickly and efficiently than humans.	Robots might take good paying jobs away from skilled workers.
Personal Data Collection	Technology can be tailored to an individual's interests, wants, and needs.	Collecting and using personal data can breach privacy.
Worker Safety	Robots and technology can perform dangerous jobs and limit injuries to humans.	Humans can lose their sense of danger. Jobs might not be done as well.
Environmental Issues	Robotic and computer technology can reduce carbon emissions and help provide clean energy.	Technology can create e-waste and often uses heavy metals and other dangerous chemicals that are hard to dispose of.
High Price of Technology	Pricing drives competition between companies spurring on innovation and new ideas.	Technology is not accessible to everyone, only those who can afford it.
Cybersecurity	Safeguards personal information from cyberattacks.	Cybersecurity is expensive, complex, and can lead to a false sense of security.
Communication	Computer technology can increase lines of communication through digital tools.	Reliance on digital tools and losing the personal interaction that comes through face-to-face communication.
Others		

Notes:

Notes:

Pitsco Education • P.O. Box 1708, Pittsburg, KS 66762 • 800-835-0686 • Pitsco.com

PE•0125•1125•03 79426