Lesson 1

Parts of a Robot: Learning About Robots and How They Function
Monday 9/26, Wednesday 9/28
65 - 75 minutes

Common Core, NGSS, ITEEA Standards	**<u>Common Core</u>:** **Integration of Knowledge and Ideas:** CCSS.ELA-LITERACY.RI.4.7 Interpret information presented visually, orally, or quantitatively (e.g., in charts, graphs, diagrams, time lines, animations, or interactive elements on Web pages) and explain how the information contributes to an understanding of the text in which it appears. **<u>NGSS</u>:** **Cross Cutting Concept**: Structure and Function/Systems and System Models **Science and Engineering Practices**: Constructing Explanations and Designing Solutions **<u>ITEEA</u>:** Nature of Technology 2. Students will develop an understanding of the characteristics and scope of technology 3. Students will develop an understanding of the core concepts of technology
Instructional Objective	Students will define what a robot is and explain how robots work by identifying each part, discussing each part's function, and describing how these parts work together to create a system that enables the robot's operation.

Telling Objective: (What mentor will announce to students)
"Today, you will be learning about robots and how they work. You will answer the question, 'How do robots know what to do?', and you will identify the parts of a robot and how these parts work together as a system in order for them to function."

Lesson Purpose
In a small group conversation, students will name the parts of the robot, explain how each part functions, and work with this information on their handouts. Ultimately, the students will be able to construct an answer to the Academy's question: 'How do robots know what to do?'

<u>ADVANCE PREPARATION (check off as you finish):</u>

- ☐ Review all handouts
- ☐ Arrive 15 minutes early to your school
- ☐ Sign in at the school office, find your classroom so you be ready begin on Day 1 – check in with the other mentors
- ☐ Review pre-assigned group names and roster (Katie will provide rosters)

Materials and Supplies
Mentors and Teacher will have:
Handouts to give students

a) Day 1 Assessment survey and pencils (to be provided separately on Day 1)
b) Student Notebooks with Handouts:
c) Cover page, "line of learning page," Lesson 1 handout (1-page, 2 sides) with C.O.D.E.R values and photo of Dash with robot definition and questions, a diagram of a generic robot with blank lines where students can connect the names to the parts of the robot, plus CLOZE word matching game. (handout follows this lesson plan)
d) USC STEM Consortium/VAST pencils (sharpened)

Each Lead Mentor will be provided with one designated Dash and one designated tablet today
Diagram of Robot Poster (can be projected as a Powerpoint)
Poster with C.O.D.E.R values (can be projected as a Powerpoint)
Word Wall Poster (to be kept and added to for each session)
Mentors will have a Roster with the names of the students assigned to them

Each Student will have:
Paper binder with handouts (see above)

Assessment: Day 1 Assessment survey

INTRO & ASSESSMENT: FIRST 10-20 MINUTES OF CLASS

DAY ONE:
As students enter the classroom, students will sit in their pre-assigned groups (there will be four groups, led by each of the four mentors). Mentors sit at stations (tables or circle of chairs) and greet students in each pod, get them quickly settled and ready to start the Academy.
The Lead Mentor introduces him/herself (why you study engineering), then provides this **cursory** description of RCA:
Mentor says:

- The Academy meets every week so you can learn to code the robots
- Weekly attendance at RCA is key to your success and progress
- You will demonstrate your skills by programming the robot dance to show in December
- The Academy also runs at two other schools, and in December, you will meet the students from other schools and demonstrate the robot dances you have coded – your families will be invited to this event
- This will be fun!

Then the other three mentors introduce themselves to the whole class and briefly tell why they study engineering and/or like coding and robots.

Lead mentor says:
We want to get to know you a little bit, so first everyone will fill out this survey–This is NOT a test, we just want to get to know more about you.

- Mentors distribute the surveys and pencils – students begin any time.
- Mentors take roll of the students assigned to their pod while the students take the pre-assessment survey. (Roll is taken at each week.)
- As mentors collect the surveys after students finish, make sure each survey has a legible record of the student and school name, and that all questions are answered (if

not, return the survey to the student and show them what question needs to be answered). Give the complete surveys to the Lead mentor to return to the VAST office. Do not lose these surveys!

- Mentors pass out the notebooks to each student who finishes the survey.

1. Anticipatory Set/Motivation/Hook

The anticipatory set should hook the students by engaging them in an interesting, thought-provoking task related to the lesson. An exciting anticipatory set motivates students to want to learn more.

Whole Class:
Lead Mentor reads the Telling Objective:
"Today, you will be learning about robots and how they work. You will answer the question, 'How do robots know what to do?', and you will identify the parts of a robot and how these parts work together as a system in order for them to function."
Lead Mentor presents the class with one Dash Robot and have the Dash execute a series of pre-programmed tasks (i.e., the Dash will say something like: "Hello class, My name is Dash, I am a robot." Then Dash will dance.)

2. Link to Existing Background Knowledge

This portion of the lesson helps students to access the knowledge they already have and introduces them to what they are about to learn.

a) Instruct students to write their names on the front of the notebook and on the page inside, where there is also a space to write their mentor's name.
b) The Lead Mentor will ask the question, "What is a robot" and "How do robots know what to do?", and give students two minutes to jot down their initial thoughts in their notebooks.
c) The Lead Mentor will then ask students to raise their hands to discuss how robots know what to do and/or to discuss the robots they already know about.
d) The Lead Mentor introduces the C.O.D.E.R. values on the next pages in the notebook and why it is important for computer scientists to follow these guidelines.
C = Collaborative
O = Open-minded
D = Determined
E = Enthusiastic
R = Respectful of their coding partners
e) Lead Mentor asks for examples of each type of behavior – move quickly so you can get to the Demonstration (next section):

3. Demonstration

In this section of the lesson, the mentor/teacher will provide the students with the new information they need to know to meet the objective for the lesson.

a) Mentors will work in small groups, demonstrating the Dash and explaining that some robots are modeled after humans, while others are designed from the imagination, but all robots have certain things in common: a robot is *"an **autonomous** system which exists in the physical world, can sense its environment and can act on it" (Mataric´, 2007). Autonomous is an academic word that means that the robot acts on its own without being directly controlled by a human.*

b) Students examine the robot and identify key components per the definition of a robot
   1. ***exists in the physical world,***
   2. ***can sense its environment,***
   3. ***and can act on it.***

c) Throughout, mentors will point out how the parts of Dash help it sense its environment, interact with it and move in it, while clearly naming the parts of a robot (as follows):
   - sensor
   - actuator
   - controller
   - power
   - housing (skeleton)
   - wireless receiver (this is not obvious on Dash, but explain the tablet signals the robot)

d) Using the Dash photo on the handout, the mentors review with their students the location of the key robot parts and describe each part's function. Students use the handout to take notes and answer questions on the handout. Mentors will explain and reinforce how these parts fit with the definition of a robot.

4. Check for Understanding
Generally, before guided practice, the mentor/teacher will want to do a class-wide check to make sure the students understand the information that they have just been provided.

a) Students will work in pairs on a diagram of a generic robot (page 2 of the Lesson 1 handout), linking each word on the diagram to the location of the part on the diagram. This exercise assists students in applying the knowledge they have just heard in the demo.

b) Mentors will review the students' completed diagrams with the group and ask individual students to find the robot parts that correspond to the function, e.g., "Which part makes the robot move?, etc." Student can say or point to the actuator, etc.

5. Guided Practice
Here, the teacher/mentor and student practice the application of the new material together. The teacher may model the activity as the students follow along.

a) Either as whole class or on the mentor's tablet, depending on the layout of the classroom – in 32nd St., mentors show the video on their tablet (find it in the photo app); at the other two schools, teachers will have overhead projectors. The students and mentors will watch a video with the following question in mind: "How does this robot know what to do?" https://www.youtube.com/watch?v=2STTNYNF4lk

b) In pairs, students will discuss their ideas about **how the Nao robot knows what to do.** (Nao is pronounced like the word "now.") Possible discussion questions:
   - What do the Nao robots respond to? How do they know what to respond to?
   - What kind of input do they respond to? How do they know to respond? How do they sense their environment?
   - What makes them move?
   - How do they know what to do?

c) Each mentor leads a conversation directing the students to look back again at the Dash and discuss **how the parts work together** in order for Dash to know what to do.

6. Independent Practice *This part of the lesson allows the students to apply independently what they have learned on their own.*
Students will individually complete the CLOZE exercise on page 2 of their Lesson 1 handouts.
7. Closure/Evaluation *This section of the lesson concludes the lesson for the students by requiring them to summarize or reflect on what they have learned.*
Revisiting their notebooks, students draw a "line of learning" in the front of their notebooks. This is simply a squiggly line that they draw under their initial response to the questions "What is a robot?" and "How do robots know what to do?", in order to separate their initial understanding from their understanding at this point, then they will write under the line. Mentors direct their students to respond **under the line this time** by re- the question, "How do robots know what to do?" Thank the students for their good work today – ask them to put their pencils in the pocket of their notebook. They can line up for dismissal. 32nd St. mentors – next week, there will not be an Academy session because the school is closed, so they will see you again in two weeks. Tell them to think about how robots know what to do during the next two weeks. (In week 2, the mentors at Alexander and Foshay will say the same thing because Week 3's session falls on a holiday.)
Post-Class Mentor-Teacher Group Reflection
Mentors and the teacher should always take 5 – 10 minutes after class to check in with each other about how the class went – what worked well, what needs improvement (let us know your ideas – we welcome your suggestions), which groups seem to be off to a good start, notice any behavioral challenges. NOTE*: Lead mentor arranges to turn in the assessment and roll sheets to the VAST office (DRB # 254) prior to Lesson 2.Contact kmills@usc.edu or 213-740-0237 if you have questions or need help turning in these materials.

Name: ______________________ Date: _________ Lesson #1

School (circle one): 32nd Street Alexander Foshay

Computer programmers have these CODER characteristics when they code on the tablet:

Collaborative
Open-minded
Determined
Enthusiastic
Respectful

Some robots look like humans and some are imaginative, like Dash, but they all share this definition: **a robot is "an autonomous system which exists in the physical world, can sense its environment and can act on it" (Mataric´, 2007).** Autonomous means that the robot can act on its own without being directly controlled by a human.

How does Dash sense its environment? This is called ***input****.* *Draw a line to Dash's sensors.*

How does Dash act on the physical world? This is called ***output****.*

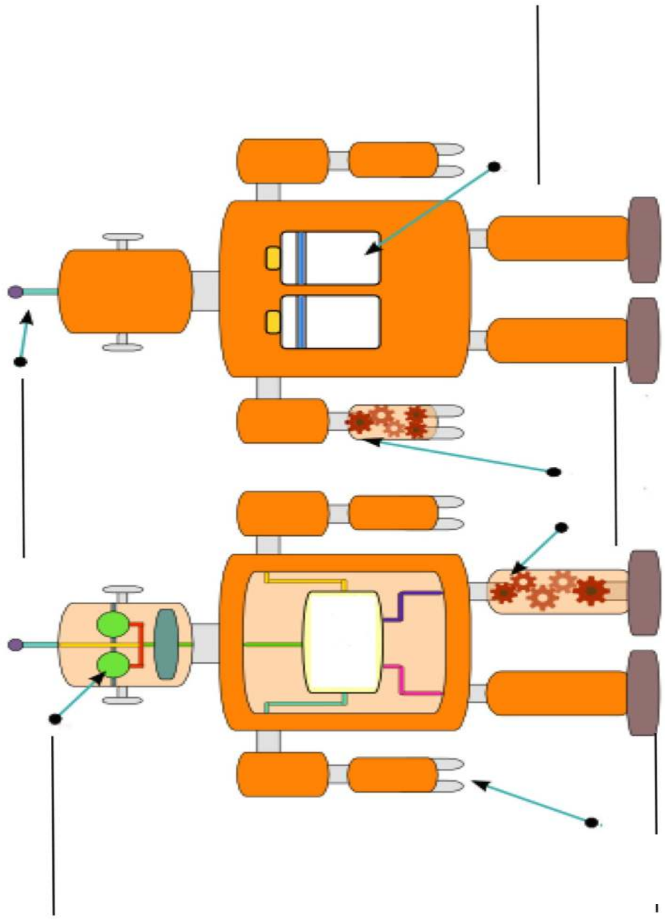
- *Draw a line to the parts that let Dash move.*
- *How does it have power to move?*
- *How does it know how to control its actions? Where is its controller? The controller in a computer is like the brain in a human.*
- *The tablet is where you program Dash. How does the tablet communicate with Dash wirelessly?*

This is the Dash robot from WonderWorks.

Name: ______________________ Date: _________ Lesson #1

School (circle one): 32nd Street Alexander Foshay

Label the parts of the robot using the Word Bank :



Word Bank

Sensor

Actuator (mechanical movement)

Power source

Wireless Receiver

Thinking Questions: Where does the input to the robot come from besides its sensors? (hint: what input does the computer programmer provide?) How does the robot create output?

Fill in the blanks with these words:

program act sensors controller sense actuator autonomous

A robot is an _______________ system. It can _______________ its environment and _______________ on it. To get input from the world, it has _______________. The part of the robot that is similar to our brain and tells it how to act is called the _______________. The _______________ is like a motor that makes the robot move. It runs a _______________, which is a set of instructions made by a computer programmer (who is also called a coder) that tell the robot what to do when it senses things.

Lesson 2
Robot Dance: Learning to Program a Sequence
Wednesday 10/5, Monday 10/10 NOTE NO SCHOOL Mon 10/3
65 - 75 minutes

Common Core, NGSS, ITEEA Standards	**Common Core:** **Integration of Knowledge and Ideas:** CCSS.ELA-LITERACY.RI.4.7 Interpret information presented visually, orally, or quantitatively (e.g., in charts, graphs, diagrams, time lines, animations, or interactive elements on Web pages) and explain how the information contributes to an understanding of the text in which it appears. **NGSS:** **Cross Cutting Concept**: Structure and Function/Systems and System Models **Science and Engineering Practices**: Constructing Explanations and Designing Solutions **ITEEA:** Nature of Technology 4. Students will develop an understanding of the characteristics and scope of technology. 5. Students will develop an understanding of the core concepts of technology.
Instructional Objective	Students will simulate a programmer-robot interaction to create, test, and revise a sequence (algorithm) of movements (i.e., a dance) to be executed by a partner (robot) that explains how humans communicate with robots through *programs* which tell the robot how to respond to input and act on the physical world.

Telling Objective: (What teacher will announce to students)
"Today, in pairs, you will take turns pretending to be either a robot or a computer programmer. Student robots will act out commands, and student programmers will code the commands. A **programmer** is a person who writes commands for a computer in a language a robot can understand."

Lesson Purpose
Students will design, test and revise a sequence (algorithm) of steps for a partner (robot) to follow. At the end of the lesson, students will construct an explanation responding to the question, ***"How do robots know what to do?"***

Materials and Supplies
Mentor and Teacher will have:
Fully charged Dash Robot and tablet – one per mentor
White board/chart paper and markers
Handout of C.O.D.E.R. characteristics
Extra handout for each student to code a rectangle and record the Twist code during activity

Assessment #2 to hand out

Each Student will have:
Their notebook and pencil

ADVANCE PREPARATION:
Each mentor will have one Dash and pre-program Lesson 2 dance (NOTE: all mentors will program the same dance below, which is the beginnings of the ripple we will work on in future lessons). Mentors can pre-program Dash before the lesson or while students complete the assessment.

> Sequence: `All robots will move forward 30cm at normal speed, spin in a complete circle to the right, move backward 30cm at normal speed, and spin in complete circle to the left. Repeat once (robot performs this sequence twice.)`

Mentors unfamiliar with the 1960's dance called the Twist should Google it or check out this Website: http://sixtiescity.net/Culture/dance.htm

Mentors will pre-assign pairs of students to start as either 1) programmer or 2) robot, then switch roles half way through the lesson.

Mentors obtain their students' notebooks prior to the start of the lesson.

Assessment #2:
Each mentor will hand out copies Assessment #2 for students to complete. LEAD MENTOR will collect and return Assessment #2 to the VAST OFFICE -- DRB #254.

Vocabulary:
sequence, algorithm, program/programmer, code/coding/coder, input, output

Pre-Lesson Assessment: 5 minutes

During the assessment, mentors take roll for their groups. Tell teacher of any absences.

a) Start in small groups. Mentors will inform students that the session today begins by seeing how much they remember from last week by completing the worksheet - this is NOT a test.
b) Distribute Week 2's Assessment.
c) Mentors will provide students no more than five minutes to complete the assessment. Keep everyone to the five-minute limit so the time is the same for everyone -- it's ok if students cannot finish, as time is part of the assessment.
d) Mentors collect assessments, checking to see that student names, school, and date are on each sheet. Lead mentor gathers all the assessments to deliver to VAST.

1. Anticipatory Set/Motivation/Hook: ~ 7 minutes
The anticipatory set should hook the students by engaging them in an interesting, thought-provoking task. An exciting anticipatory set motivates students to want to learn more.

a) Mentors continue in their pods. Have Dash perform the pre-programmed dance steps (see above).

b) Mentor will say: "Please describe each step to the Dash dance you just watched." (Mentor can demo the dance sequence again as many times as necessary.)

c) Mentor randomly records the student responses to the whiteboard. Then ask the group to reorder the responses in the sequence of the dance.

2. Link to Existing Background Knowledge: ~ 5 minutes

This portion of the lesson helps students to access the knowledge they already have and introduces them to what they are about to learn.

a) Mentors ask students how robots know what to do and review the robot parts learned during last week's lesson (and included in Assessment #2), linking parts to their functions:
   Mentors will ask students what the **controller** is like (the robot's brain) and also what the actuator does (moves the robot).

b) Mentor hands out the students' notebooks and tells students to look at the top of Lesson #2:
   "The robot parts enable the robot to perform its **output**, and the program is part of the robot's **input**. Remember that a robot is programmed, which means that a programmer writes commands or instructions that the robot follows. This sequence of instructions is called an **algorithm**."

c) Explain that people have sequences of activity in their daily lives, so students can start paying attention to the sequences or algorithms in their lives. For instance, there is an algorithm for taking a bite of cereal, brushing your teeth, or solving a math problem. Recipes are also algorithms, because to prepare food, you must follow the same steps in the same order each time. Ask for a few more examples.

d) Mentor will ask students: "Who remembers the C.O.D.E.R. characteristics?" and prompt students to give examples of how they can practice this type of behavior as they act as the "programmer" and "robot" (C.O.D.E.R. = collaborative, open-minded, determined, enthusiastic, respectful). Mentors emphasize that these values will be important to demonstrate today as they learn to program.

3. Demonstration: ~ 15 minutes

In this section of the lesson, the mentor/teacher will provide the students with the new information they need to know to meet the objective for the lesson.

a) Tell students they are going to learn about sequences by pretending to be a robot so they can see that a robot needs very specific instructions.
   a. Next, look at the bottom of the Lesson 2 handout in their notebooks to see a screenshot of Blockly code. Mentor should read it out loud and explain to the students that 50 (in the Forward 50 command) is a measure of distance. The turn command has a 90 next to it; that informs the robot to turn on a 90 degree angle. Ask the students: "Who can show me what a 90 degree angle is?"
   b. Now that they understand the numbers, have students perform the sequence. Students should substitute 3 steps for the 50 in the Blockly screenshot.
      - This sequence makes a square. Mentors clarify any confusion about the input (code on the handout) and the output (the square that students will make by acting out the sequence on the notebook handout.)
   c. Pass out the extra handout for Lesson 2 and ask students to work on the first page, filling in the blanks to write a code telling Dash to move ***fast*** in the shape of a

rectangle, turning ***right***. Remind them a rectangle has a longer top and bottom than its sides. Help students fill in the blanks.

b) Mentor tells students they will work in pairs as a pretend "programmer" and a pretend "robot" in order to learn to write an algorithm. This algorithm will be used to code the steps a human would follow to dance the Twist.
c) The mentor will model thinking out loud about how one would write a dance move in the Twist:
   "Who knows the dance, the Twist? How do you do it? (Have students model or mentor can physically model the dance.) If someone had never heard or seen this dance, how would you tell them what to do with their body?"
d) Have students narrate the sequence of steps to a simple dance move while the mentor writes the ideas down on the white board or paper.
e) Next, the mentor reminds students that the written instructions are the robot's **input** and now we will focus on the **output**. Call on one student at a time to act out a few of the steps exactly as written, being sure not to fill in any gaps in the dance instructions or add anything that is missing. Students will notice that they may need to add details to clarify the sequence about how to move the arms, hips or anything they may have missed.
f) Students and mentor will add the steps necessary to create an "algorithm" for the Twist that can repeated so that everyone can clearly see the "code" for telling someone how to do a dance step to the Twist.

4. Check for Understanding: ~ 5 minutes
Generally, before guided practice, the mentor/teacher will want to do a class-wide check to make sure the students understand the information that they have just been provided.

The mentors ask students to narrate the instructions for doing two new ***simple*** dance steps. By raising their hands, students will take turns inventing and narrating simple steps, e.g., turn head left then right. Working together, the mentor narrates instructions for the two dance steps while students record those in their new handout before they move on to working in pairs.

5. Guided Practice: ~ 10 minutes
Here, the mentor/teacher and student practice the application of the new material together. The teacher may model the activity as the students follow along.

a) The mentor places students into pre-assigned pairs (also called dyads) and designates one person in each pair as the "robot" and the other as the "programmer," reminding students they will switch roles midway in the activity. Students will have two minutes to quickly collaborate on naming one short dance step within the Twist (e.g., twist arms to the left, twist arms to the right);
b) The students will quickly go around their group, with each pair demonstrating while narrating their selected dance step (emphasize that input is as important as the output, so the pair must narrate the step, not just perform it). This should be quick and fun, but the mentor ask questions to clarify any vagueness in the narration or to correct misunderstandings about input, output, or sequence.
c) The mentor emphasizes that students will continue this work, writing down their steps they practice before they actually begin to code their robot to dance next week.

6. Independent Practice: ~ 15 minutes
This part of the lesson allows the students to apply what they have learned on their own.

a) Students work in pairs collaboratively for five minutes, with the pretend “robot” suggesting two ***simple*** dance steps while the pretend “programmer” writes down the steps on today’s extra handout.
b) Coming back together as a pod, students will take turns to watch each pair as a “programmer” reads the code out loud while the “robot” acts out the code. After each dance, students discuss ways to write the instructions for those dance steps. (This is an important process of learning computational thinking by doing it.)
c) Going back to dyads, each student will switch roles (the robot becomes the programmer and vice versa), and repeat the entire process for one complete turn, such that each student in the dyad gets practice as both the programmer and the robot.
d) Coming back together as a pod, students will take turns to watch each pair as a “programmer” reads the code out loud while the “robot” acts out the code.
e) Tell everyone to put the extra handout in the pocket of their notebooks.

7. Closure/Evaluation: ~ 10 minutes
This section of the lesson concludes the lesson for the students by requiring them to summarize or reflect on what they have learned.

a) Mentors wrap up the activity in their pods, closing with these questions:
    a. What did you notice when you were writing an “algorithm” for your “robot?”
    b. What did you notice when you were the “robot” and you were “running” an algorithm?
b) Have students answer the question, “How do robots know what to do? What is their input and output?”
c) Students complete CLOZE passage in their notebooks:
    Correct answers: A programmer is a person who writes commands for a computer in a language it understands. When writing a program, one needs to break down the sequence into very specific steps. An algorithm is a sequence of instructions, or rules to follow, for performing a task. A robot uses the input from its sensors and a program to make decisions and act. A robot’s actions are its output.
d) Students write for three minutes under the CLOZE passage in their handout:
    a. What are some examples of different algorithms that you use in your life?
e) Have student go back to the Line of Learning sheet (page 2 of their notebooks), “How do robots know what to do?” Students should read their previous responses and draw another “Line of learning” under their previous response and **add the date to the line** so they can see how their ideas evolve each week. Ask students to discuss what they would like to add to or change in their response. Have them write these ideas under today’s Line of Learning.
f) Have students put today’s handout into the front folder part of their notebook along with the pencils. Close their notebooks and pass them up to the mentor.

NOTE: The Wednesday schools will not have class next week on October 12, so remind them that the Academy skips that week but everyone will be back on October 19.

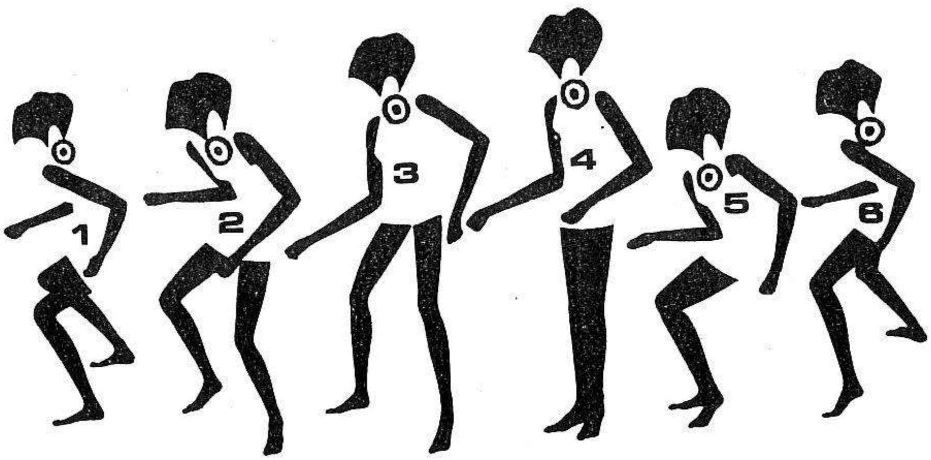
7. Additional Notes and Variations

9. After Class – Teacher and all the mentors

Mentors and teacher put away equipment and notebooks so everyone knows where these are stored. When will teacher be able to charge the tablets and Dash robots before the next class?

Briefly review Assessment #2 with the teacher and all mentors after class to see how many students remembered the robot components – Lead mentor arranges to get these assessments to the VAST office (DRB #254). Do not lose the Assessment. Also let Lead Mentor and teacher know if any students were absent so we can track this.

Mentors and the teacher should always take 5 – 10 minutes after class to check in with each other about how the class went – what worked well, what needs improvement (let us know your ideas – we welcome your suggestions), which groups seem to be off to a good start, notice any behavioral challenges. Communicate your observations to us at vast@usc.edu -- THANKS!



1. Step forward on to left foot, at the same time bending at the knees and lowering the left shoulder.

2. Step forward on to right foot, still with knees bent, but on this beat lower right shoulder.

3. Take a step back with left foot at the same time beginning to straighten.

4. Take a step back with the right foot, now straightening to upright position. Repeat this three more times.

5. Step to the side with left foot. Close right foot to left. Bend knees and then straighten again. Step to side with right foot. Close left foot to right. Once again bend knees and then straighten them. Repeat this three more times. Now repeat first step four times. Now repeat pattern once more but this time make a quarter of a turn each time on the first variation and a half turn on each of the second variations. With feet slightly apart, bend at the knees and sway from right to left then from left to right. Repeat this three more times.

6. Take a step forward with the left foot at the same time bending at the knees and lowering left shoulder. Without moving feet sway back so that weight is on the right foot. Repeat three more times. Now go back to the first variation for four more times. Then, starting from first variation, do each of the other variations doing only one of each. Keep going until fade of record.

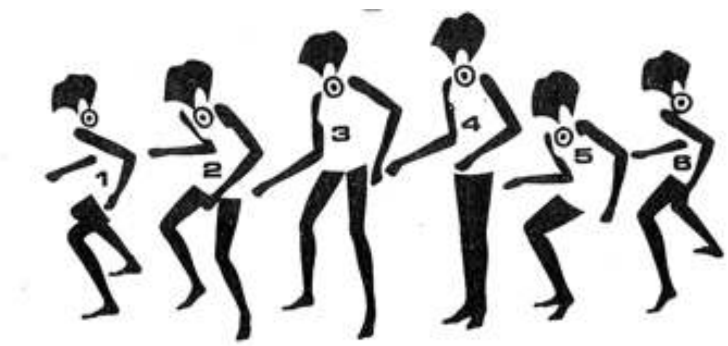
BEND IT! STEP-BY-STEP

IT had to happen! That smash hit for Dave Dee, Dozy, Beaky, Mick and Tich —"Bend It"—has been just crying out for some bright person to devise a '66 dance for it. And who better than Patrick Kerr who introduced countless dances to "Ready, Steady, Go" viewers? Above is his step-by-step idea of how it should be done. Read it, put on the record and get bending!

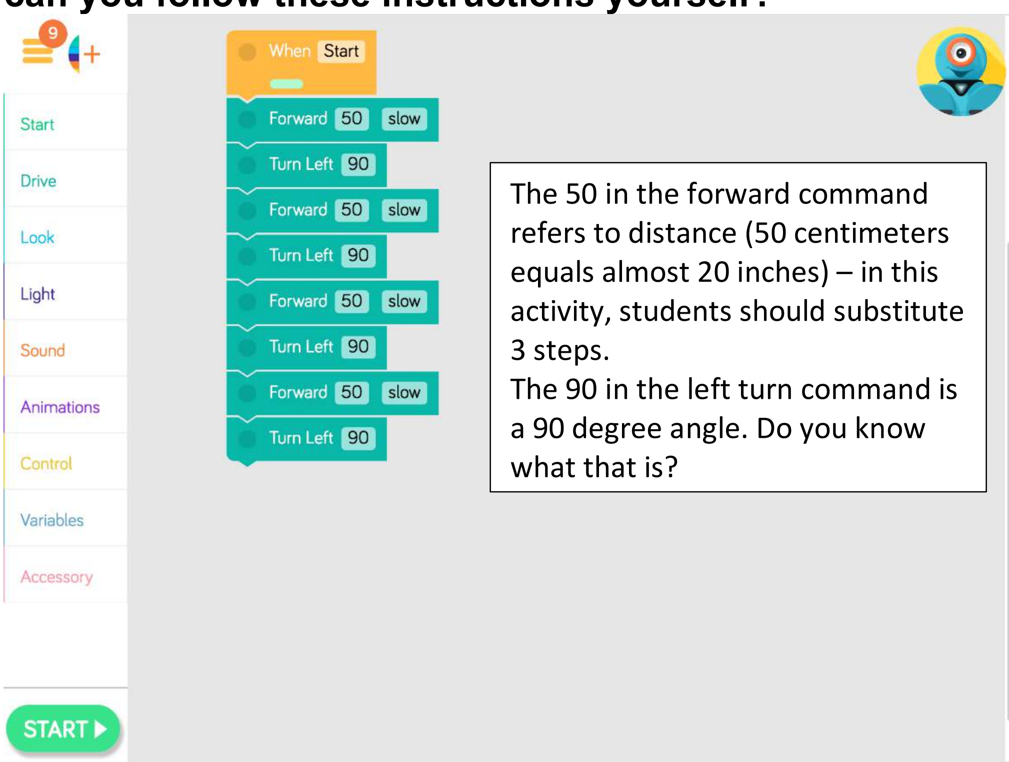
Name: ______________________ Date: _________ Lesson #2

School (circle one): 32nd Street Alexander Foshay

An algorithm is a sequence of instructions written in computer code or commands – this in the __input__ that tells the robot how to act in the world (robot's __output__). Humans also behave in sequences when they dance, brush their teeth, make a recipe. What other sequences can you think of in your daily life?



Below is a short sequence of instructions written in computer code – can you follow these instructions yourself?



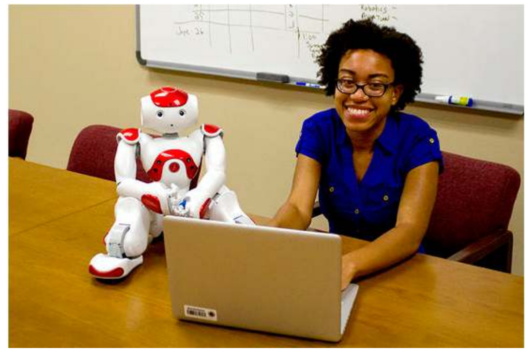
The 50 in the forward command refers to distance (50 centimeters equals almost 20 inches) – in this activity, students should substitute 3 steps.
The 90 in the left turn command is a 90 degree angle. Do you know what that is?

Name: ______________________ Date: _________ Lesson #2

School (circle one): 32nd Street Alexander Foshay

Now, students will work in pairs, trading roles as as the programmer and the robot.

source of image: https://www.linkedin.com/pulse/pair-programming-how-sell-andrea-malfer-poma



source of image: http://pc.fsu.edu/Students/Student-Profiles/Jessica-Haley-Business-major-robot-programmer-inspires-girls

Demonstrate the CODER values when working in pairs

Check off each quality you demonstrate while working in your pair:

☐ **C**ollaborative ☐ **O**pen-minded

☐ **D**etermined ☐ **E**nthusiastic ☐ **R**espectful

How did you demonstrate respectful and collaborative behavior today when working with your partner?

Name: ______________________ Date: _________ Lesson #2

School (circle one): 32nd Street Alexander Foshay

Match the words to the correct blanks:

programmer	output	algorithm
input		

A ____________ is a person who writes commands for a computer in a language it understands. An ____________ is a sequence of instructions, or coding rules for the robot to follow when performing a task. A robot uses the ____________ from its sensors and the computer program to make decisions and act. A robot's actions are its ____________.

What are some examples of different algorithms you use in your life?

Return to your sheet in the front of this binder with the "lines of learning" to update your answer to the question: "How do robots know what to do?"

Lesson 3
Beginning to Code
Monday 10/17, Wednesday 10/19
65 - 75 minutes

Common Core, NGSS, ITEEA Standards	**<u>Common Core</u>:** <u>CCSS.MATH.CONTENT.4.MD.C.5.A</u> An angle is measured with reference to a circle with its center at the common endpoint of the rays, by considering the fraction of the circular arc between the points where the two rays intersect the circle. An angle that turns through 1/360 of a circle is called a "one-degree angle," and can be used to measure angles. **<u>NGSS</u>:** **Cross Cutting Concept**: Cause and Effect/System Models **Science and Engineering Practices**: Analyzing and Interpreting Data **<u>ITEEA</u>:** 9. Students will develop and understanding of engineering design. 17. Students will develop and understanding of and be able to select and use information and communication technologies. *Q. Technological knowledge and processes are communicated using symbols, measurement, conventions, icons, graphic images, and languages that incorporate a variety of visual, auditory, and tactile stimuli.*
Instructional Objective	Students will discuss and analyze an image of pre-written code in order to identify all the variations of the "drive" function in the *Blockly* programming language. Students will be presented with the written task they will code their robot to execute, applying the steps of the Engineering Design Process to select and assemble commands to be tested. If there are any errors, the Engineering Design Process teaches students to deconstruct the code to improve the program.

Telling Objective: (What mentor/teacher will announce to students)
"Today, you will learn how to use the Blockly programming language to write your own code for Dash. Today's sequence includes turns and distance, and the segment you code today will become part of the longer dance performance for Dash you will present in December. You will learn the **Engineering Design Process** and use the C.O.D.E.R. characteristics while coding."

Lesson Purpose
The purpose of this lesson is to introduce students to the **Engineering Design Process** as a way to learn to use the Blockly language, to develop and use technical vocabulary, and to collaborate on coding a segment of their robot's final dance performance.

Materials and Supplies
<u>Mentor & Teacher will have:</u>
1 tablet and 1 Dash
Poster/handout of C.O.D.E.R. values (in mentor's lesson plan from previous week)
Poster or enlarged copy of **Engineering Design Process**

Whiteboard/chart paper
Student handouts delivered to the classroom to pass out (see handout after lesson plan)
Assessment 2 delivered to the classroom

ADVANCE PREPARATION:
Before session, ask the teacher to please charge Dashes and tablets.
Mentors to pre-assign groups (dyads or triads) and decide which pair will work with Code Sequence 1 or Code Sequence 2 in section 5: Guided Practice.
Mentors to retrieve notebooks prior to start of class.

- While the students complete the assessment, pre-program one tablet with the dance. Throw the code away before students begin coding so they don't re-use it.

```
The robot will move forward 30cm at normal speed, then spin in a
complete circle (360 degrees) to the right. Next, it will move
backward 30cm at normal speed, then spin in complete circle (360
degrees) to the left. Then, it will turn to the right by 45
degrees, go forward 30 cm at normal speed, turn to the left by a
45 degree angle and go backwards 30 cm at normal speed. Then Dash
will turn to the right by 90 degrees, then turn back by 180
degrees.*
```

*direction of the 180 turn does not matter

Each Student will have:
Notebook with pencils

Vocabulary:
function, interface, menu, drag and drop, degrees of a circle, complete circle (360 degrees), right angle (90 degrees), centimeter (cm), input, output

At some point, take roll for their groups during the session and inform the teacher and lead mentor and teacher of any absences.

Pre-Lesson Assessment: 5 minutes

a) Work today in small groups. Mentors will inform students that the session today begins by seeing how much they remember from last week by completing the worksheet - this is NOT a test.
b) Distribute Week 3's Assessment.
c) Provide students no more than five minutes to complete the assessment. Keep everyone to the five-minute limit so the time is the same for everyone -- it's ok if students cannot finish, as time is part of the assessment.
d) Mentors collect assessments, checking to see that student names, school, and date are on each sheet. Lead mentor gathers all the assessments to deliver to VAST.

1. Anticipatory Set/Motivation/Hook -- ~ 5 minutes
The anticipatory set should hook the students by engaging them in an interesting, thought-provoking task. An exciting anticipatory set motivates students to want to learn more.

Working in small groups, each mentor shares the "Telling Objective."

Have Dash demonstrate the pre-programmed dance. Tell students they will program this entire dance sequence or **algorithm** today. Repeat the dance while students call out the steps (as well as they can – don't aim for accuracy, jut encourage observation).

2. Link to Existing Background Knowledge
This portion of the lesson helps students to access the knowledge they already have and introduces them to what they are about to learn.

a) **Algorithms in daily life -- ~ 2 minutes**

Mentor asks:

"Remember, last week we discussed that an algorithm is a sequence coded for a computer, as we just watched Dash perform. Has anyone noticed a human sequence in our daily lives such as brushing teeth or using a recipe. Who can share a sequence they regularly perform in their daily lives? (e.g., getting ready for school, lining up when the school bell goes off, preparing for Halloween)

- Have group respond, and mentor helps with the examples if they need to be defined or articulated as a sequence.

b) **Coding a Robot using Blockly -- ~ 5 minutes**

Mentors hand out notebooks and Lesson 3 handout. Have all the students begin by opening their notebooks, opening the brads in the back, first adding the Lesson 2 supplement from last week to the BACK of their notebooks, then add in today's Lesson 3 handout into their notebooks. Mentors ensure the order is correct.

Mentor tells students to study the image on today's Lesson #3 handout of lines of **code**. Ask students to remember they saw something like this in Lesson #2 when they made a square and a rectangle. To learn to program, they begin by understanding how to read and write the code.

Remind students:

*"Code" is written in a "programming language," which is the **input** a human uses to communicate with a computer. Code is a **sequence** or **algorithm** of commands that tell the robot what to do. There are many different types of computer languages, and in our class, we use the **Blockly** language on the tablets to program the Dash robot.*

Ask: *What do you notice in the code at the top of the Lesson #3 handout?*

(Students responses vary). In this process, students read the lines of code, point out what they notice and say what they think it means and why, and ask questions about those things that they don't understand; all the while, they will be writing explanations and notes on the page itself.

Show: Mentor defines specific parts of the Blockly interface on handouts and on the tablets:

1. **Menu (along the left vertical)**
2. **Functions (within the menu: drive, lights, sound, etc.)**
3. **Drag and drop**
4. **Editing interface**

- Have students answer questions on page 1 of the handout, offering help with the answers (ANSWERS=any of these: forward, backward, fast, slow, turn right, turn left).

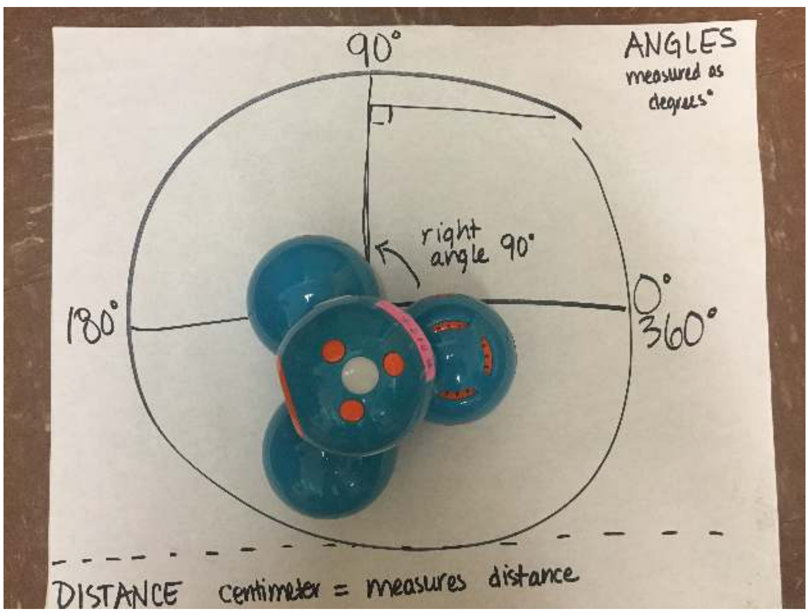
c) **Present the Coding Challenge -- ~ 10 minutes**

Mentor asks students to use today's handout and read out loud the project description:

```
The robot will move forward 30cm at normal speed, then spin in a
complete circle (360 degrees) to the right. Next, it will move
backward 30cm at normal speed, then spin in complete circle (360
degrees) to the left. Then, it will turn to the right by 45
degrees, go forward 30 cm at normal speed, turn to the left by a
45 degree angle and go backwards 30 cm at normal speed. Then Dash
will turn to the right by 90 degrees, then turn back by 180
degrees.
```

- Have students act out the words as if they were the robot, taking time so that students connect the movements with the words. Remind students their code will be the **input** but now they will pretend to be the robot and perform the **output**.

Use the handout or white paper with a circle and degrees: as shown below, put the Dash or have a student stand in the center of the white paper to demonstrate 360, 180, 90, and 45 degrees.



Ask:

Who remembers what a centimeter (cm) is?

It's a measure of distance that is smaller than an inch.

Review:

Use questions on the students' handout about distance, speed, and degrees of a circle.

Who remembers what a 90 degree angle is?

So what is a 45 degree angle?

3. Demonstration -- ~ 12 minutes

In this section of the lesson, the mentor/teacher will provide the students with the new information they need to know to meet the objective for the lesson.

Review of Engineering Design Process

Mentor tells students that programming Dash is a challenge or a problem that they need to solve by using the **Engineering Design Process**. Quickly review the steps of the chart with the pod.

- Show the larger version of the Process (on p. 8 of this lesson plan).

Introduce Task by reading aloud:

```
The robot will move forward 30cm at normal speed, then spin
in a complete circle (360 degrees) to the right. Next, it
```

`will move backward 30cm at normal speed, then spin in complete circle (360 degrees) to the left.`

The mentor will model the **Engineering Design Process** by working each step out on a poster or whiteboard, as indicated below. Students will follow along on their handouts:

Ask: ***First I need to read the criteria and ask what exactly it is that my robot needs to do. What are the first steps that I need to code for?***

Have students re-read the first sentence of the task.

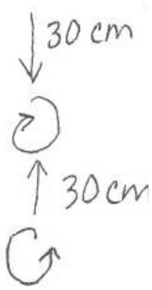
Imagine: ***Next I need to imagine these steps as separate steps or commands.***

Have students be the robot and act out the sequence. The mentor writes those steps down in a bulleted list form on paper (or whiteboard/chart paper):

```
1. move forward 30cm at normal speed
2. spin in a complete circle (360 degrees) to the right
3. move back 30cm at normal speed
4. spin in complete circle (360 degrees) to the left
```

Plan: ***After that I need to think about what commands I know and plan it out in symbols***.

The mentor will translate the bulleted list into pseudo code (i.e., just simple arrows, something like what is below).



Normal

R 360

Normal

L 360

Create: ***Now I need to use the Blockly-Dash interface to write my code.***

Mentor models how to open the program, start and name a new project and code the first step, narrating how to navigate the menu and select a function for movement (Drive).

- For the second step, the mentor asks the students for help selecting the function.

****For the spin, have the robot only spin 180˚ in order to give students a problem to solve***.

Emphasize that everyone ALWAYS has trouble getting each step perfect while coding, so the **Engineering Design Process** allows for mistakes. Finding the "bugs" or mistakes in a program is part of being a coder, which is why coders develop the C.O.D.E.R. characteristics.

Improve: ***Now I am going to run the code I wrote and see if it works.***

Run the code. If they don't notice, ask the students, *Did it spin around completely?* Look at the Blockly code and have them also look at the degrees of a circle graphic.

Ask: ***How many degrees do I need it to turn to have it spin in a complete circle?***

Make the change and run it again.

4. Check for Understanding-- ~ 8 minutes

Generally, before guided practice, the mentor/teacher will want to do a class-wide check to make sure the students understand the information that they have just been provided.

a) Tell the students that they now know the first few steps of the dance. Next they will work in pairs or triads to finish programming the dance by using the steps of the **Engineering Design Process** in their notebook/handout.
b) Next, have the students connect to their robot and start a new project on the tablet, which they name with their team name.
c) Have students program on their tablets the code for the dance steps that the mentor already modeled. (Mentors will help and also ask students to show the menu, to describe what a function is, etc.)

5. Guided Practice -- ~ 10 minutes
Here, the mentor/teacher and student practice the application of the new material together. The mentor/teacher may model the activity as the students follow along.

Tell students:
Now you and your group are going to code the next two steps. What will you need to do?

- Have students talk through each step of the **Engineering Design Process** to ensure that they understand the process. Assign one student as the task master, one as the recorder, and one as the coder; collaboratively, the team reviews the instructions and records the code for one step, then the students can shift roles for coding the next step.
- Students use the handout for each step in the **Engineering Design Process** as they solve the problem of translating the task into Blockly code.
- Assign one team to code Sequence 1 and a different team to work on Sequence 2.

Code Sequence 1

`Next, Dash will turn to the` **right** `by 45 degrees, go forward 30 cm at normal speed, turn to the` **left** `by a 45 degree angle and go backwards 30 cm at normal speed.`

Code Sequence 2

`Next, Dash will turn to the` **left** `by 45 degrees, then go forward 30 cm at normal speed, turn to the` **right** `by a 45 degree angle and go backwards 30 cm at normal speed.`

Mentors assist students as they go through the process:

- Students write down their plan before they begin to create their code on the tablet.
- Have students run their tests on the Dash and discuss before they move on.

6. Independent Practice -- ~ 5 minutes
This part of the lesson allows the students to apply what they have learned on their own.

Mentor tells them they are on the last two steps of the sequence, which they will complete on their own.

Code Sequence 1, continued

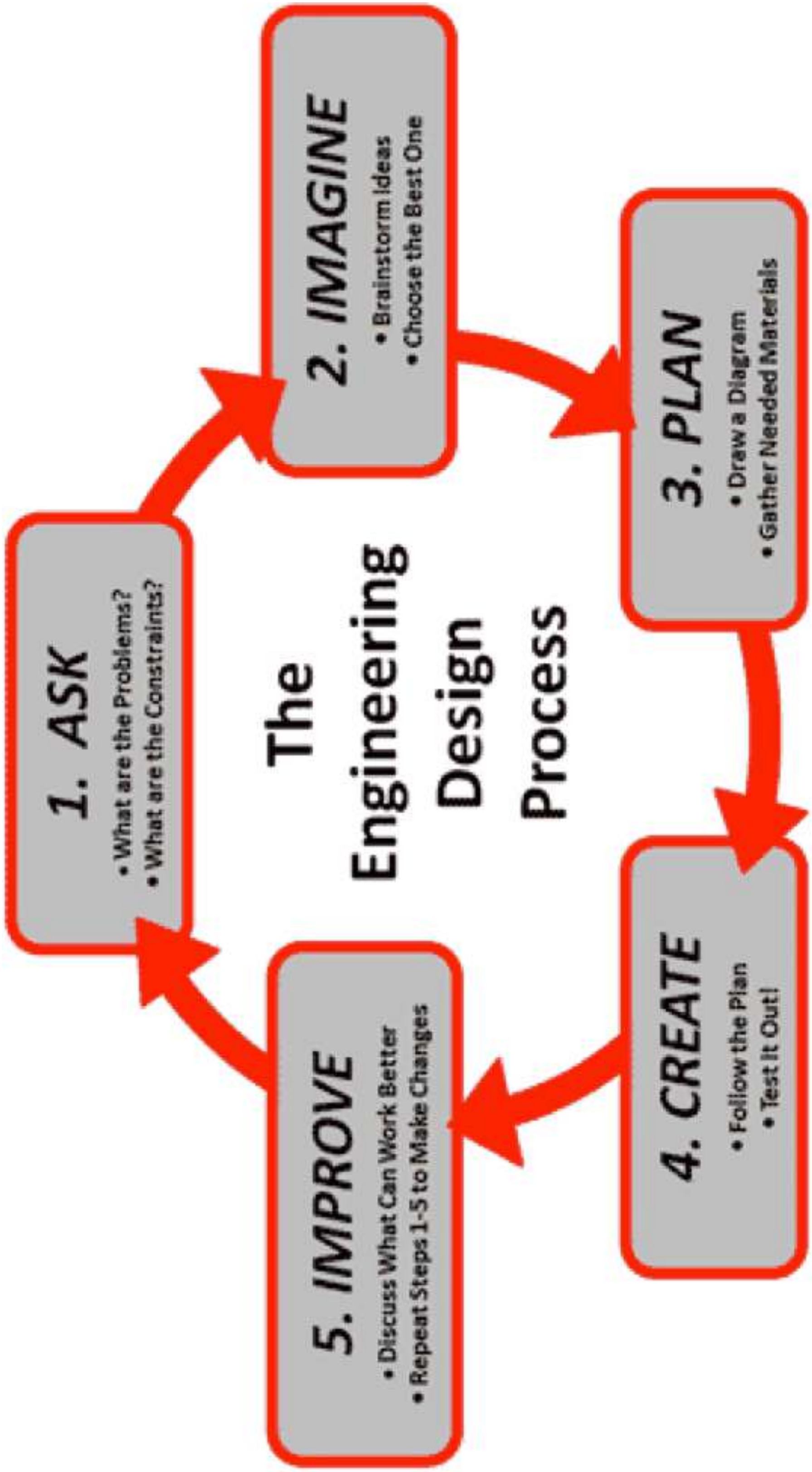
`Then Dash will turn to the` **right** `by 90 degrees, then turn back by 180 degrees.`

Code Sequence 2, continued

`Then Dash will turn to the` **left** `by 90 degrees, then turn back by 180 degrees.`

7. Closure/Evaluation -- ~ 10 - 15 minutes *This section of the lesson concludes the lesson for the students by requiring them to summarize or reflect on what they have learned.*
a) One by one, each group within the pod will run its code while everyone else watches. After each run, have the entire pod discuss what they see, what they learned in the process and what questions remain. Make any improvements to the code. b) Then put two Dashes next to each other, the one with Code Sequence 1 on the right and the one with Code Sequence 2 on the left. Start the two Dashes at the exact same time and see how the experience of the dance changes with two Dashes programmed differently. c) The mentor will review the learning by asking students to complete the final portion of their handout. d) Reflection Questions (if there is time): ➢ *When the robot doesn't complete a task, is it because the robot makes a mistake? Explain your reasoning.* ➢ *Coders often make mistakes, which is why we use the Engineering Design Process. Which of the C.O.D.E.R. characteristics do you think help coders not to feel frustrated when they make a mistake? Why?* Have students share their opinion and reasoning.
7. After Class
Mentors and the teacher should always take 5 – 10 minutes after class to check in with each other about how the class went – what worked well, what needs improvement (let us know your ideas – we welcome your suggestions), which groups seem to be off to a good start, notice any behavioral challenges. Email vast@usc.edu about any absences and cc your classroom teacher. Arrange for Assessment 2 to be delivered to the VAST office.

Mentors show this diagram when demonstrating the Engineering Design Process



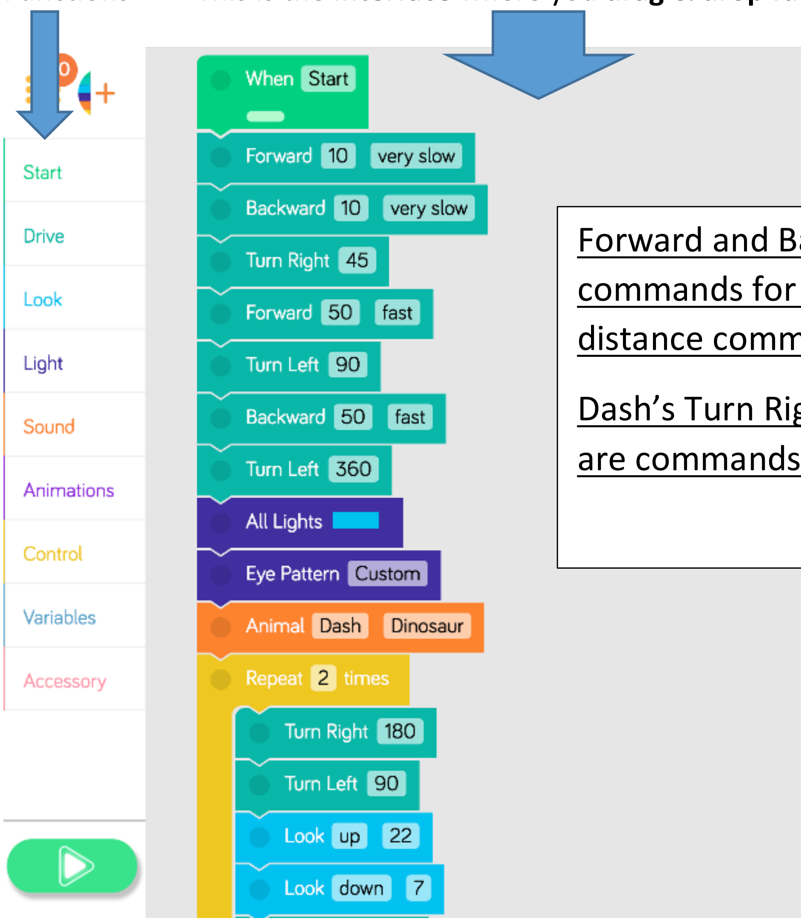
Name ____ School ___________________________

Date _____________________________

Below is the Blockly programming language or code that provides the input for Dash. Discuss and write what you see happening in this code.

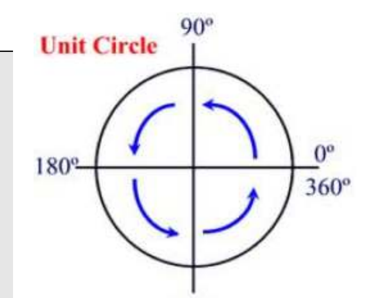
Blockly Menu of Functions

This is the **interface** where you **drag & drop functions**



<u>Forward and Backward movements are commands for distance & speed. Circle the distance commands in the Blockly code.</u>

<u>Dash's Turn Right and Turn Left movements are commands for degrees on a circle</u>



The **Drive function** tells Dash to move.
What are four ways Dash moves in the code above?

1.____

2.____

3.____

4.____

Which function in the Blockly Menu of **functions** will you use to code today's engineering design challenge?

____

What you will code Dash to do today:

The robot will move forward 30cm at normal speed, then spin in a complete circle (360 degrees) to the right. Next, it will move backward 30cm at normal speed, then spin in complete circle (360 degrees) to the left.

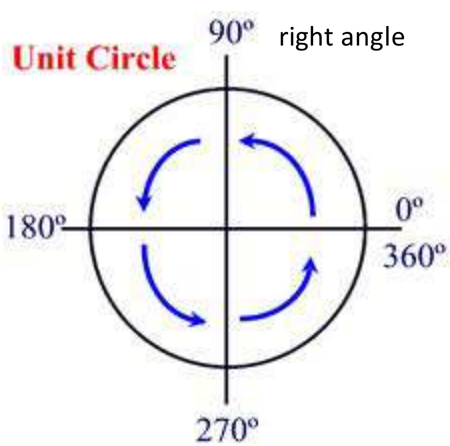
Then, it will turn to the right by 45 degrees, go forward 30 cm at normal speed, turn to the left by a 45 degree angle and go backwards 30 cm at normal speed.

Then Dash will turn to the right by 90 degrees, then turn back by 180 degrees.

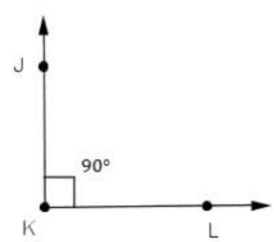
Measurements used in today's computer program:

DISTANCE : 1 centimeter (cm) = 0.4 inch

Below, the diagram shows how many degrees (◦) in a circle or angle.



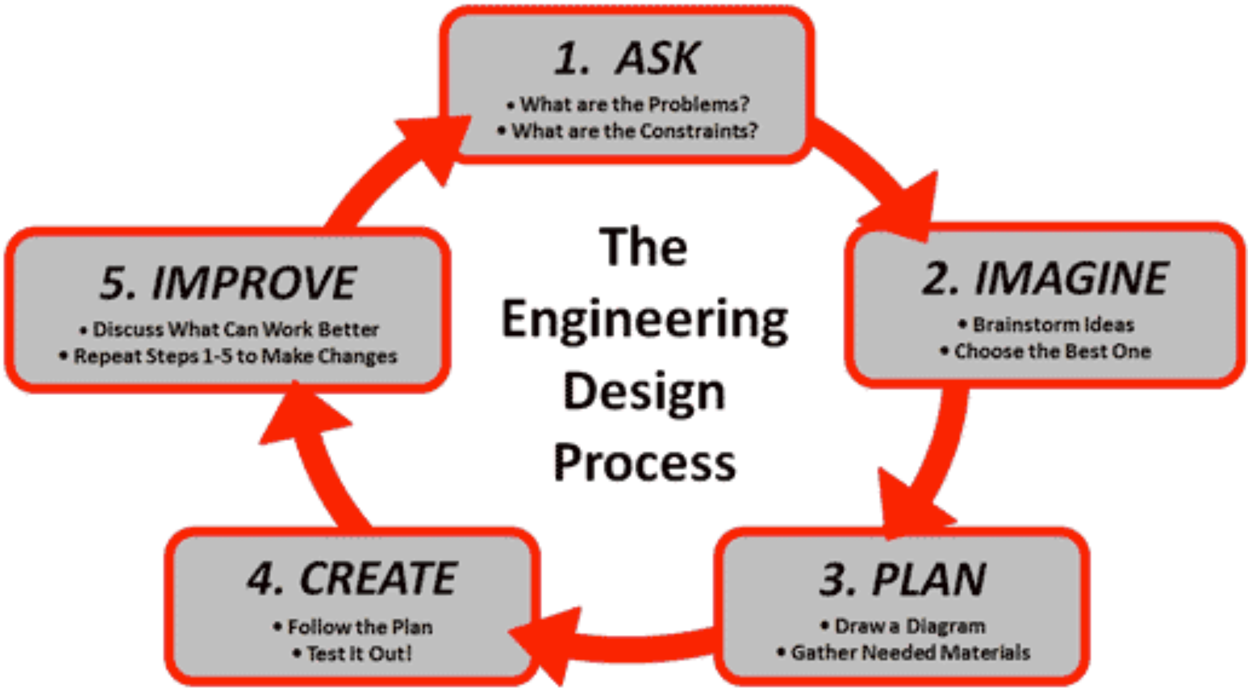
Source of this unit circle: http://study.com/academy/lesson/radians-and-degrees-definition-examples-quiz.html



How many degrees ◦ in a full circle? _______________ in a half circle?___________

How many degrees ◦ in a right angle? ____________________________

How many degrees ◦ in ***half*** of a right angle?________________________________



Use the Engineering Design Process with your team to break down the code into a sequence (algorithm):

ASK
What is the problem/challenge?

Task for Dash (criteria) in Lesson 3
Follow along while your mentor shows this process

```
The robot will move forward 30cm at normal speed, then spin in
a complete circle (360 degrees) to the right. Next, it will
move backward 30cm at normal speed, then spin in complete
circle (360 degrees) to the left.
```

Imagine
Ideas

What are all the steps you will need to code for? Act out the movements so you can imagine each step.

Plan
Draw a simple diagram

The person serving as the recorder makes sure everyone writes down the plan here.

Create **Watch your mentor code the plan in Blockly**

Improve
Run the code with Dash and compare to criteria (test).

What changes do you need to make?

ASK

What questions do you have now?

Now use the Engineering Design Process with your team to code the next part of the dance.

ASK
What is the problem/challenge?

Task for Dash (criteria) in Lesson 3
Follow along while your mentor shows this process

First, code the section already worked out with the mentor:

The robot will move forward 30cm at normal speed, then spin in a complete circle (360 degrees) to the right. Next, it will move backward 30cm at normal speed, then spin in complete circle (360 degrees) to the left.

Now, your mentor will tell you to code either Sequence 1 or 2:

Code Sequence 1

Then, it will turn to the **right** by 45 degrees, go forward 30 cm at normal speed, turn to the **left** by a 45 degree angle and go backwards 30 cm at normal speed.

Code Sequence 2

Then, it will turn to the **left** by 45 degrees, go forward 30 cm at normal speed, turn to the **right** by a 45 degree angle and go backwards 30 cm at normal speed.

Imagine
Ideas

What are all the steps you will need to code for?

Plan
Draw a simple diagram

What are all the steps you will need to code for?

Create **Code your plan in Blockly**

Improve
Run the code with Dash and compare to criteria (test).

What changes do you need to make?

ASK: What questions do you have now?

Did you finish coding today or do you need more time?

Circle one: Finished Need more time

*When the robot's **output** is incorrect and Dash does not perform the directions, is it because the robot makes a mistake? Explain your reasoning.*

____

____

____

____

____

____

Coding is a trial and error process, so mistakes are common. The Engineering Design Process makes room for mistakes (also called "bugs"), so the whole process focuses on finding and fixing the mistakes. Which of the C.O.D.E.R. characteristics do you think help coders avoid feeling frustrated when they make a coding mistake? Why?

Collaboration Open-Mindedness Determination Enthusiasm Respect

____

____

____

____

____

Lesson 4
Using Loops for Coding Efficiency
Monday 10/24, Wednesday 10/26
65 - 75 minutes

Common Core, NGSS, ITEEA Standards	**Common Core**: CCSS.ELA-LITERACY.W.5.2.D Use precise language and domain-specific vocabulary to inform about or explain the topic. CCSS.ELA-LITERACY.SL.5.3 Summarize the points a speaker makes and explain how each claim is supported by reasons and evidence. **NGSS**: Support an argument with evidence, data, or a model.(5-ESS1-1), (5-PS2-1). **Cross Cutting Concept**: Scale, Proportion and Quantity **Science and Engineering Practices**: Analyzing and Interpreting Data **ITEEA**: 9. Students will develop and understanding of engineering design. 17. Students will develop and understanding of and be able to select and use information and communication technologies.
Instructional Objective	Students will work with algorithms that repeat in order to learn how to use loops in coding. Students will discuss and articulate how a repeat function in coding increases efficiency. They will collaboratively engage in the **Engineering Design Process** to program dance steps with repetitions. They will run tests, make careful observations, and compare the robot's movements to their coding task in order to engage in discourse and then to draw conclusions about ways to improve the efficiency of the code.

Telling Objective: (What mentor/teacher will announce to students)
"Today, you are going to learn to become more efficient coders by using the repeat function. Computer programmers call the repeat function a 'loop.' You will use the **Engineering Design Process** to write efficient programs focused on changes in speed and direction as we continue programming Dash in a dance we will present in December."

Lesson Purpose
For students to be able to use loops to program algorithms that include changes in speed and direction.

Materials and Supplies
Mentor and Teacher will have:
Fully charged Dash Robot and tablet
White board/chart paper and markers
Student handout for Lesson 4

ADVANCE PREPARATION
- ☐ Review programming tasks and think through pseudocode
- ☐ Review today's student handouts
- ☐ Chart or scratch paper for each mentor
- ☐ Make sure all robots and tablets are charged

- Have class roster to take roll. Let Lead Mentor and Teacher know of absences.

Vocabulary:
loop, repeat, efficiency, algorithm, pseudocode

1. Anticipatory Set/Motivation/Hook ~ 5 minutes
The anticipatory set should hook the students by engaging them in an interesting, thought-provoking task related to the lesson. An exciting anticipatory set motivates students to want to learn more.

a) In small groups, the Mentor will read the Telling Objective:
"Today, you are going to learn to become more efficient coders by using the repeat function. Computer programmers call the repeat function a 'loop.' You will use the **Engineering Design Process** to write efficient programs focused on changes in speed and direction as we continue programming Dash in a dance we will present in December."

b) Ask students to pretend to be robots and narrate out loud as they move step by step to make two squares in a row.
   - How will students create **input** to perform the **output** of two squares? Ask students to work out this challenge and demonstrate their answer. They can work together or independently.

c) Mentors pass out the Lesson 4 handout. Use the white board or paper while students use page 1 of today's class handout to write in simple pseudocode the algorithm for each step of the two squares. Review together what everyone has written, discussing what makes a pseudocode clear (there will be no correct pseudocode, so just model how to evaluate when a sketch gives sufficient information).

d) Ask students how to make this coding more efficient? It may help to work out the following questions together on white board or paper while students work on their handouts:
   - How many turns for one square? How many commands for one right turn and one forward movement?
   - How many squares?
   - Can they suggest ways to indicate repeated movements in pseudocode?

2. Link to Existing Background Knowledge ~ 5 minutes
This portion of the lesson helps students to access the knowledge they already have and introduces them to what they are about to learn.

Mentors ask students to turn to page 2 of today's handout and read the Blockly algorithm, interpret the shape, and determine how many times that shape repeats. Have students write answers in the space below the screenshot, then discuss students' ideas and plans for making the code more efficient.

- If students have trouble conceiving how to make the code more efficient, have them draw a line between the code for each of the two squares and write a pseudocode for each square on the handout.
- Then have students think about how many turns there are in one square (4 turns of Forward and Right Angle) – work out in pseudocode. Have students write in simple pseudocode on their handouts. (Distance and speed of forward can be arbitrarily chosen by students as long as the choices are consistent.)

Next, mentor demonstrates how to turn on the tablet, how to link it to the Dash, then asks each student to code one side of a square and one right turn. Each student codes just one portion and passes tablet to the next person – talk about how long it takes to program so many sides and turns of each square.

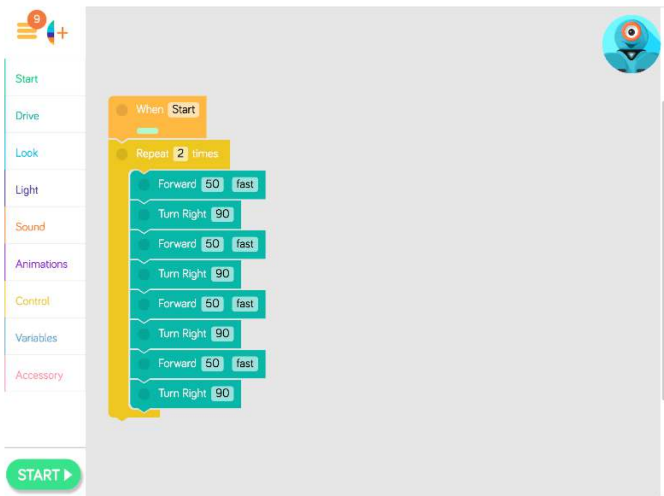
When four students have input the code, ask students what to do next to make this process more efficient. Tell students they will use the **Engineering Design Process** to solve the problem of coding efficiently.

- Ask students to turn to page 3 of today's handout and fill in the blanks on the engineering process circle.
- Students can work together or independently, but have them think about the name of each step, using the hints below the blanks so they reason out the steps rather than merely memorize the name of each step.

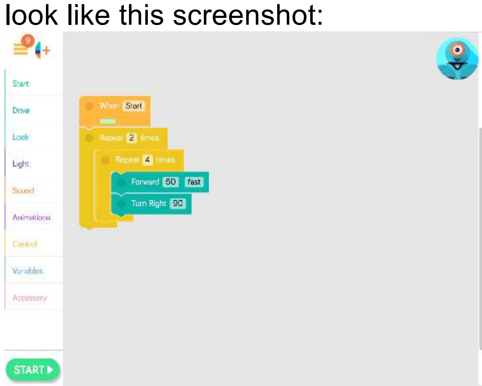
3. Demonstration ~ 10-15 minutes (10 minutes for mentors at 32nd Street)

In this section of the lesson, the mentor/teacher will provide the students with the new information they need to know to meet the objective for the lesson.

Return the group's attention to the tablet, where the students previously coded the four steps of one square (Forward + Turn Right 90 degrees).
Mentor demonstrates the REPEAT function on the tablet and then show them the code to turn that one programmed square into two squares (which should look like this screenshot):



Discuss and make sure everyone understands. Then ask students how to make this algorithm even more efficient? Then show everyone the code on the tablet (which should look like this screenshot:



Run Dash to show the **output** that matches the **input** using loops. Remind students that the REPEAT function programs what coders call LOOPS.

Tell students they will now begin working on today's coding challenge by using the **Engineering Design Process.**

Students will program two different dance steps that require repetition, the **Dash Shuffle** and the **Dash Boogie**, using loops. Ask students to follow along with today's coding challenge on their handouts as student volunteers slowly read out loud each step in the **Dash Shuffle**.

Encourage students to circle words or act out steps as the algorithm is read out loud:

```
Dash Shuffle (2 times)
Dash turns right by 90 degrees, moves forward fast by 10cm
3 separate times, turns 180 degrees to the left, moves
forward fast 3 separate times by 10 cm each time, then
faces forward toward the audience.
```

The Mentor models the **Engineering Design Process**, stating the name of each step and asking students to repeat the name of each step (in red, below) and to watch the progression of the Process on p. 3 of their handouts**:**

Ask: The ASK step merely requires students to re-read the directions carefully and think about: "What are the steps I will need to code, and how can I code them efficiently?"

Imagine: Mentor and student talks through and act out the instructions together:

- Visualize the angles.
  - Right turn is 90 degree angle.
  - What is a 180 degree turn?
- Discuss how to code the repeated moves forward by 10 cm
- Think about which way Dash turns to face the audience (turn R by 90 degrees).
- Don't forget the REPEAT the sequence.

Plan: Mentor asks everyone to write simple pseudocode on p. 3 of today's handout:

- Use the white board or scratch paper to work out simple pseudocode to the students.
- Emphasize the REPEAT function and how to use it.

Create: Model how to turn on the tablet, connect to the Dash, then how to name a new project with the team's name (have team create a name). Then, have students take turns programming each step – remind them to use C.O.D.E.R. characteristics to work collaboratively and with respect. Everyone watches. Mentor can answer questions but do not improve mistakes so students can go through the IMPROVE step next.

Improve: Mentor has a student use START to test the code and models narrating each step out loud as Dash moves so the students learn how to check the output against the intended input. Go back over the criteria/challenge description, having students read through each step in order to confirm that all elements have been completed.

Ask: "Is there anything else that should be improved or changed?" Make sure students update the pseudocode on their handouts so each one has a written record of the **Dash Shuffle**.

4. Check for Understanding ~2 minutes

Generally, before guided practice, the mentor/teacher will want to do a class-wide check to make sure the students understand the information that they have just been provided.

Ask if there are any questions before students move on to code the **Dash Boogie**.

5. Guided Practice ~15 minutes

Here, the teacher/mentor and student practice the application of the new material together. The teacher may model the activity as the students follow along.

Mentor tells students they will code the **Dash Boogie** by yourselves.
Asks students to read the steps out loud together:

Dash Boogie (repeat 2 times)
Dash looks to the right by 90 degrees, looks to the left by 180 degrees, looks forward toward the audience, turns a 90 degree angle to the right, turns 180 degrees to the left, faces forward toward the audience and spins a 360 degree turn.

Ask: What is the first step of the **Engineering Design Process** you will use? (Have students answer the blank at the bottom of p. 4 of their handout.)

- Tell students to turn to p. 5 of their handouts. What are the next two steps of the Engineering Design Process they will use? Students fill in the blanks on p. 5
- Have students count off each step on the handout so that each student is in charge of planning the code of one movement, then the group can collaborate on the subsequent moves.
- Monitor the collaborative work to ensure that everyone is taking turns, respecting each student's thought process and letting that student solve the problem in his/her own way (students should not solve another person's challenge). Everyone writes pseudocode for each step on the handout.
- Use the C.O.D.E.R. chart on p. 6 whenever students need to be reminded of working collaboratively and with respect.

6. Independent Practice ~15 minutes

This part of the lesson allows the students to apply independently what they have learned on their own.

a) Mentor answers any questions.
b) When students have written pseudocode for each step on p. 5, hand them the tablet and tell them to add to the previous code for the Dash Shuffle the new code for the Dash Boogie. Tell them to carefully add new code because it is easy to disrupt the previous code.
c) Each student takes a turn coding her/his step while the group watches until the entire code is written.
d) Now run Dash so it executes both the Dash Shuffle and the Dash Boogie. Does any coding need to be improved?
e) The mentor will review the learning with a brief Fill-in-the-Blanks activity with small groups:

The algorithm is a sequence of steps or commands to followed the same way each time. In programming, a loop is a sequence that is repeated. Programmers use loops to be more efficient. The command on Blockly for a loop is repeat.

7. Closure/Evaluation ~8 minutes (longer for mentors at Alexander & Foshay) *This section of the lesson concludes the lesson for the students by requiring them to summarize or reflect on what they have learned.*
a) In the last five minutes of class, the Lead Mentor asks for everyone to gather in the middle of the classroom so we can see the Dashes perform their dance together. b) Each mentor brings her/his group's Dash to the center of the room, lining up all the Dashes so they are all facing forward – have the audience all move on one side rather than encircle the Dashes. The mentors all press start at the exact same time so the students can see the Dashes performing the dance steps in synchrony. Clap at the end. If there is time, repeat. c) Tell students that next week they will learn how to program all the Dashes so they dance together in an even more advanced way. ➢ If there is time, tell students they will learn to use the WAIT function so that each Dash will execute a turn at different times. Collect student notebooks and put your name on the tablet and Dash you have been using so your students' code will be saved.
8. Post-Class Mentor-Teacher Group Reflection
Mentors and the teacher should always take 5 – 10 minutes after class to check in with each other about how the class went – what worked well, what needs improvement (let us know your ideas – we welcome your suggestions), which groups seem to be off to a good start, notice any behavioral challenges. **Reflection Questions:** *What went well? Did the students show evidence of problem solving? Did the students show evidence of meeting the lesson objective? What were the challenges of this session? Are there any recommendations for improvement?*

Name ____ Date ______________________________

Circle School: 32nd Street Alexander Foshay

Working with your partners, pretend to be a robot and make two squares in a row. Walk through this shape twice to determine how many turns and forward movements you need to make.

How many right angle turns do you make when walking through two squares? _______________

How many times do you go forward between right angle turns? _____________________

Pseudocode is the name for a quick sketch of the code a programmer needs to write. Below, sketch out some **pseudocode** for a robot that needs to make two squares:

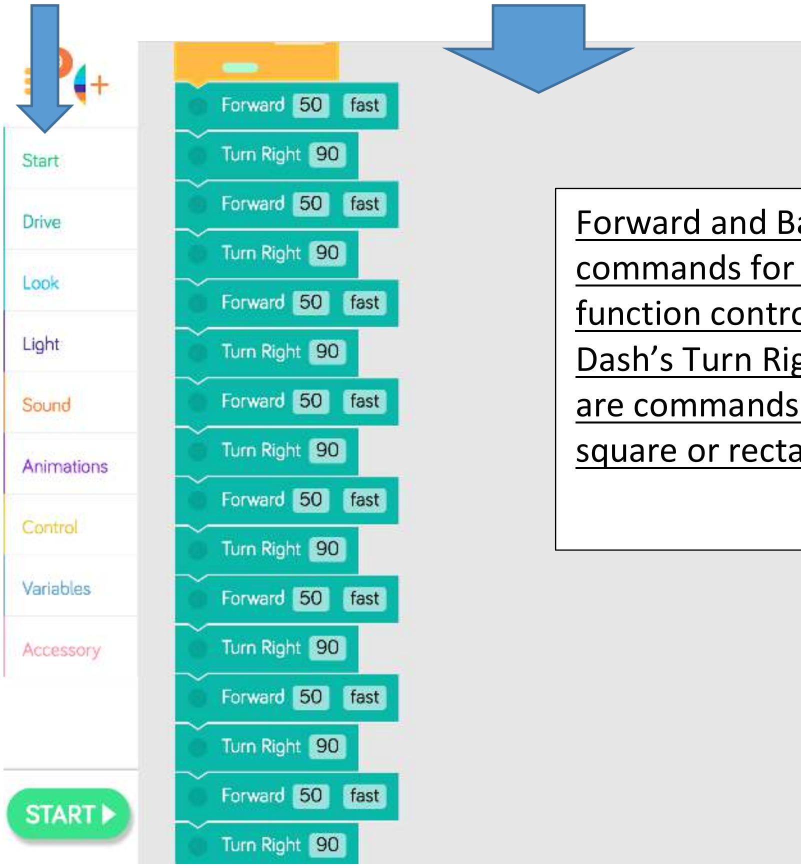
Now write **pseudocode** that will make these algorithms more efficient:

- How many squares?______________How many turns for one square? _______________
- What commands can be repeated?

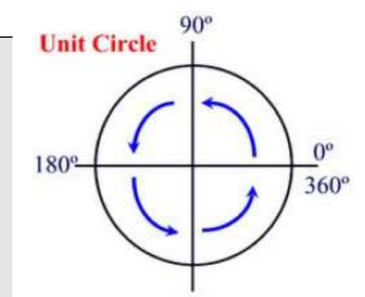
The Blockly algorithm below is **input** to Dash. What shape will Dash's **output** be?

1. Circle one: Square circle triangle rectangle
2. Does this shape repeat? Yes No

Blockly
Menu of
Functions This is the **interface** where you **drag & drop functions**



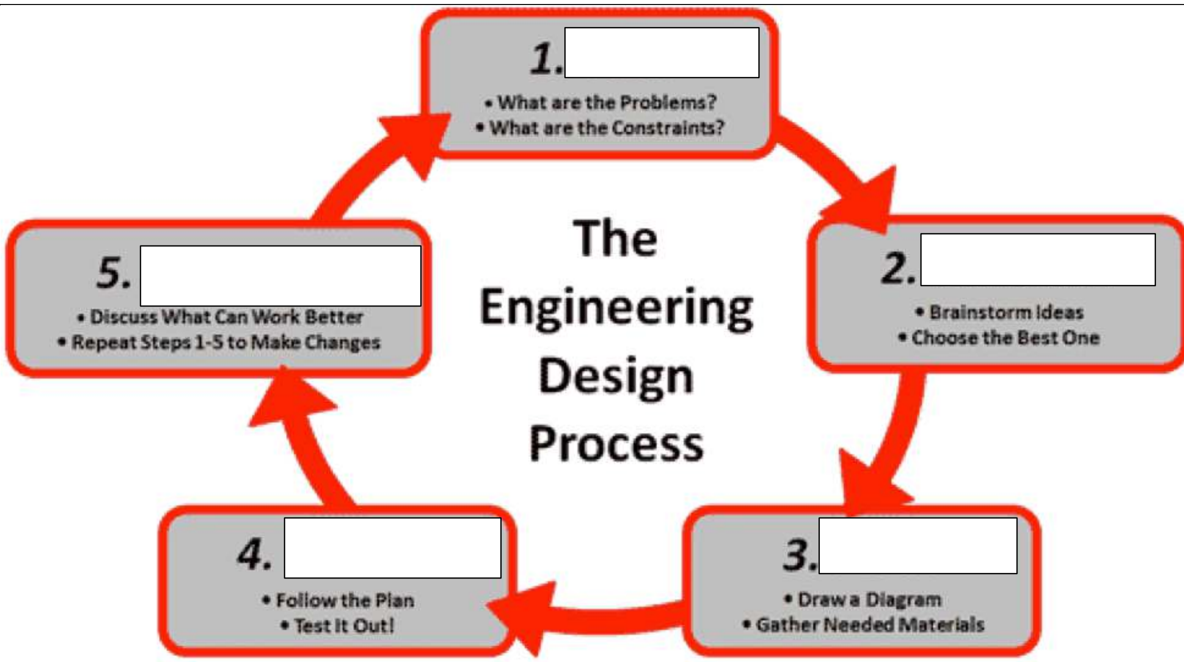
<u>Forward and Backward movements are commands for distance & speed. Which function controls movement? ______</u>
<u>Dash's Turn Right and Turn Left movements are commands for degrees on a circle or in a square or rectangle.</u>



This algorithm is very long and takes a lot of repetition to program. Coders want to make programming efficient. What do you notice about the repetitions, and what are some ways to make this algorithm more efficient?

Use the words below to fill in the blanks on each step of the Engineering Design Process:

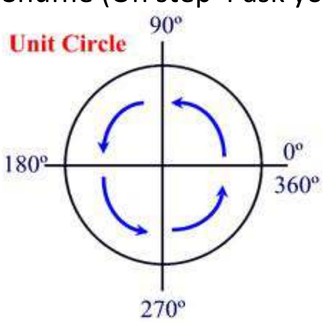
Improve Imagine Create Plan Ask



Today, you will program Dash in two dance steps: The **Dash Shuffle** and the **Dash Boogie.**

<u>Dash Shuffle (2 times)</u>
Dash turns right by 90 degrees, moves forward fast by 10cm 3 separate times, turns 180 degrees to the left, moves forward fast 3 separate times by 10 cm each time, then faces forward toward the audience. Repeat.

Write notes here as you use the **Engineering Design Process** to plan your code for the Dash Shuffle (On step 4 ask your mentor to let your team write the code and test it out):



Independent Practice:

Working on your own in groups, use the **Engineering Design Process** to plan your algorithm for the Dash Boogie. Below, write the names of your role and your partner's roles:

Dash Boogie (repeat 2 times)

Dash looks to the right by 90 degrees, looks to the left by 180 degrees, looks forward toward the audience, turns a 90 degree angle to the right, turns 180 degrees to the left, faces forward toward the audience and spins a 360 degree turn. REPEAT

Use the Engineering Design Process here to imagine and plan the algorithm:

Step 1: (write here the name of the first step)____________________

Step 2: Step 3:

Work out these steps as a group, with each student in charge of planning the code of one movement:

1. **Dash looks to the right by 90 degrees,**
2. **looks to the left by 180 degrees,**
3. **looks forward toward the audience,**
4. **turns a 90 degree angle to the right,**
5. **turns 180 degrees to the left,**
6. **faces forward toward the audience and spins a 360 degree turn.**
7. **REPEAT**

Write pseudocode here for the entire algorithm, then run the program (Step 4) and improve (Step 5):

Check off each C.O.D.E.R. quality you demonstrated today:

☐ **C**ollaborative ☐ **O**pen-minded

☐ **D**etermined ☐ **E**nthusiastic ☐ **R**espectful

How do you know when you are being treated collaboratively and respectfully? What does it look, sound, or feel like?

____

____

____

Match the words in the box to the correct blanks in the paragraph:

loop	repeat	algorithm	efficient

The ________________________is a sequence of steps or commands to followed the same way each time. In programming, the ______________________ is a sequence that is repeated. Coders use loops to be more __________________. The command on Blockly for a loop is ______ _____________.

What does it mean to be efficient? Give an example.

____

____

____

Lesson 5

Programming for Timing
Monday 10/31, Wednesday 11/2
65 – 75 minutes

Common Core, NGSS, ITEEA Standards	**Common Core:** CCCS.MATH.CONTENT.4.MD.A.2 Use the four operations to solve word problems involving distances, intervals of time, liquid volumes, masses of objects, and money, including problems involving simple fractions or decimals, and problems that require expressing measurements given in a larger unit in terms of a smaller unit. Represent measurement quantities using diagrams such as number line diagrams that feature a measurement scale. **NGSS:** **Cross Cutting Concept**: Scale, Proportion and Quantity **ITEEA:** 9. Students will develop an understanding of engineering design. 17. Students will develop an understanding of and be able to select and use information and communication technologies.
Instructional Objective	Students will begin to create code for more advanced algorithms, controlling the Drive function with the “wait” feature. Students will program different times of execution for the turns of each individual robot in order to execute a ripple spin and then reverse it. They will collaboratively problem solve and plan, running tests to evaluate the accuracy of the robots’ timing while executing the algorithm in order to make adjustments to their code.

Telling Objective: (What mentor/teacher will announce to students)
“Today, your teams are going to work together to write code that controls the time your robots need to create a ‘ripple spin.’ You will do this with the wait control, and you will have to determine how long your particular robot should wait depending on where it is in the ripple.”

Lesson Purpose
Program a more complex algorithm that includes wait time in between commands.

Materials and Supplies
Mentor & Teacher will have:
Description of challenge

Advance Preparation:
- Ask the teacher to charge the Dash and tablets.

- Mentors can download the ripple video onto their phones or tablets to show in class (LAUSD has a firewall that makes it difficult to access the internet when in the classroom) https://www.youtube.com/watch?v=udW2sROLyJU
- Mentors should review last week's programming on the tablet to confirm if the Dash Shuffle and Dash Boogie are still there and still correct. For Lesson 5, you can drag Lesson 4's algorithms to the right, but keep them within the interface so that they can later connect today's programming task to the dance steps already coded. If necessary, rebuild the Dash Shuffle and Dash Boogie (hopefully with the students' help) after today's coding challenge. Lesson 4's programming instructions can be found at the back of today's lesson plan.

Each Student will have:
Handout to add to their binders with a description of today's challenge and Engineering Design worksheet.

Vocabulary:
time, units, seconds, clock, ripple

Assessment (5 minutes)
When all the students are seated with their mentor, the mentors will pass out Assessment 3 and give students five minutes to complete both sides. When mentors collect Assessment 3, make sure each student has put her/his name on the top along with the date and school name.

1. Anticipatory Set/Motivation/Hook (5 minutes)
The anticipatory set should hook the students by engaging them in an interesting, thought-provoking task. An exciting anticipatory set motivates students to want to learn more.

Whole Class:
The Lead Mentor will have all the students get in line and perform the "wave" by lifting their arms one at a time while the mentors stand near their pod and make sure they understand and perform the directions.

- Ask students: "How could you program your Dash robots to do this or something similar?" (Students respond, and mentors prompt them to think about wait time).

Lead Mentor reads the Telling Objective: "Today, your teams are going to work together to write code that controls the time your robots need to create a 'ripple spin.' You will do this with the wait control, and you will have to determine how long your particular robot should wait depending on where it is in the ripple."

Next, the Lead Mentor will explain what a **ripple** is and ask them to perform a ripple by turning in a full circle, one student beginning a turn as soon as the previous student turns a full circle. The other mentors help the students understand and perform the instructions.

Lead Mentor counts off each pod, naming one as Robot 1, another as Robot 2, etc., assigning each group its place in the ripple (spots 1 – 4). While everyone is still working as one large group, have all the students in group 1 turn 360 degrees while everyone else watches. When group 1 finishes the full 360 turn, everyone in group 2 begins its 360 degree turn. When group 2 finishes the full 360 turn, then group 3 begins, etc.

Clarification: some people prefer the term "cannon" to "ripple," however we made the decision to proceed with the term "ripple" for the Academy.

2. Link to Existing Background Knowledge (5 minutes)
This portion of the lesson helps students to access the knowledge they already have and introduces them to what they are about to learn.

Return to the group pods. The mentors will explain:

> *Dance is fundamentally about moving through space a particular way at a particular time. If dancers did not have timing, the dance would be chaos: they would be out of sync with the music, dancers would bump into one another, and the audience would not be able to perceive patterns that make the dance aesthetically pleasing. As we saw, performing a ripple is one dance move where timing is extremely important. Again, a ripple is where one dancer moves at a time, one right after the other to create a wave or ripple effect.*

Optional to watch first 30 seconds of video:
https://www.youtube.com/watch?v=udW2sROLyJU

Mentors introduce wait function by telling students:

> *So far, we have programmed the Dash to dance by itself. Now we are programming algorithms that require Dashes to work together. In order to achieve the ripple with your robots, you will have to use the wait function to control when your robot moves and coordinate that with the other Dash robots..*

3. Demonstration (15 minutes)
In this section of the lesson, the mentor/teacher will provide the students with the new information they need to know to meet the objective for the lesson.

Mentors present coding challenge:

Coding Challenge

Create a ripple of all the robots spinning completely (360 degrees) to the right, one by one, starting with robot 1, which turns completely before robot 2 turns completely, then robot 3 turns completely before robot 4 turns completely. Then create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1.

LATER: Insert this ripple before the Dash Boogie.

NOTE: if there are more Dashes available and mentors can capable of working with extra more advanced students, then add more than 4 robots to the ripple.

Mentors work within their pods, each using one robot to code a ripple. Count off among the mentors – 1, 2, 3, 4 – so each group knows how to code the wait time for the robots that spin first.
i.e., Robot 1 turns without a delay
Robot 2 turns after Robot 1 completes its turn (about 2 seconds)
Robot 3 turns after Robot 2 completes its turn
Robot 4 turns after Robot 3 completes its turn

In pods, the mentors will go through the sections of the handout organizer; however, the handout is for the independent task later. Students should be engaged in the discussion and not writing on the worksheet.

Ask: *What is the task for your robot?*
Answer: A ripple where each robot spins, one right after the other.

Imagine: *What are all the steps our robot will need to execute a ripple? Let's look closely at all of the instructions and underline the key words.*
Answer: Spin completely (360 degrees); right; ripple; reverse; left

Mentor - *Now to visualize this sequence, let's act out once again the moves as if we were the robots ourselves* **(Mentors or students act out the ripple)**

Plan: *Now, I will write down the actions of my robot in symbols.* Mentor writes symbols as pseudocode for actions needed for a single robot to do steps in ripple.

Create: Mentor codes in Blockly in front of the students, using the plan draft and having student help out with suggestions. *(Deliberately plan for timing to be slightly off)

Improve: The mentors and all the pods come back together to test out the ripple – the mentors are in charge of starting the Dashes all at the same time. In testing, the students will notice the mistake in timing. Ask students: *What adjustments need to be made?*

4. Check for Understanding (10 minutes)
Generally, before guided practice, the mentor/teacher will want to do a class-wide check to make sure the students understand the information that they have just been provided.

Have students narrate to their partners or the group the steps of the process they just witnessed. Make sure to answer any clarifying questions.

Bring together the entire class to demonstrate the ripple as it is performed by all the robots together. Mentors should line up robots 1 – 4 (or 1 – 6) and all mentors press start at the same time. How did that work? Discuss as a group if any adjustments need to be made. Feel free to run the ripple multiple times if students enjoy or if repetition helps clarify.

5. Guided Practice (10 minutes)

Here, the teacher and student practice the application of the new material together. The teacher may model the activity as the students follow along.

Return now to pods.
Mentors guide students through the steps of the **second ripple**. As before, mentors walk students through the steps of performing the first and second ripple, so students understand the different wait times for the second ripple.

Ask: *What is the rest of the task for your robot?*

Create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1.

Imagine: *What are all the steps your robot will need to execute? Let's look closely at all of the instructions of the coding challenge and write down the key words.*

Keywords: Spinning completely (360 degrees); left; reverse

Now visualize these actions from beginning to end. What other factors will you have to account for at each phase?

Answer: Timing of next/previous robot

(Next, have students pretend to be the robots and act out the instructions.)

6. Independent Practice (10 - 15 minutes)
This part of the lesson allows the students to apply what they have learned on their own.

Students will continue through the steps of the handout organizer in their own teams/groups independently.

Plan: *Now, write down a draft of what you will code. You can use symbols, but be sure to include directions, angles, and times.*

Create: Have students work in teams to write the code for their robots.

Improve: Students run a test two times as a whole class. Mentors will facilitate the tests. Have students compare the robot's performance with the criteria and take notes on what they noticed at the beginning, middle, or end in timing, movements or position. In pods, mentors will ask students what adjustments need to be made.

If there is time, insert the ripple between the Dash Shuffle and the Dash Boogie and let Dash perform the growing dance routine.

7. Closure/Evaluation (5 minute)
This section of the lesson concludes the lesson for the students by requiring them to summarize or reflect on what they have learned.

Students will complete the questions on the handout, answering the following questions:

1. What did you learn about planning and coding through this challenge?
2. How did your robots know when to turn?
3. How could you make this robot more autonomous (more able to make decisions on its own)? How do you know when to move in a wave? What sensor could you add to allow your robot to collect more input from the environment to respond to?

Post-Class Mentor-Teacher Group Reflection

Mentors and the teacher should always take 5 -10 minutes after class to check in with each other about how the class went, what worked well, what needs improvement (let us know your ideas – we welcome your suggestions), which groups seem to be off to a good start, notice any behavioral challenges.

NOTE*: Lead mentor arranges to turn in the assessment and roll sheets to the VAST office (DRB # 254) prior to Lesson 6. Contact kmills@usc.edu or 213-740-0237 if you have questions or need help turning in these materials.

Here's a review of the Dash Shuffle and Dash Boogie so that you can slip the ripple between the two:

Dash Shuffle (2 times)

Dash turns right by 90 degrees, moves forward fast by 10cm 3 separate times, turns 180 degrees to the left, moves forward fast 3 separate times by 10 cm each time, then faces forward toward the audience.

Ripple

Create a ripple of all the robots spinning completely (360 degrees) to the right, one by one, starting with robot 1, which turns completely before robot 2 turns completely, then robot 3 turns completely before robot 4 turns completely. Then create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1.

Dash Boogie (repeat 2 times)

Dash looks to the right by 90 degrees, looks to the left by 180 degrees, looks forward toward the audience, turns a 90 degree angle to the right, turns 180 degrees to the left, faces forward toward the audience and spins a 360 degree turn.

Name ____ School ___________________________

Date _____________________________

Programming for Timing

ASK
What is the problem/challenge?

Task for Dash (criteria):

Create a ripple of all the robots spinning completely (360 degrees) to the right, one by one, starting with robot 1, which turns completely before robot 2 turns completely, then robot 3 turns completely before robot 4 turns completely. Then create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1. LATER -- <u>Insert this ripple before the Dash Boogi</u>

Imagine
Ideas

What are all the steps you will need to code for?

Plan
Draw a diagram

Create **Code in Blockly**

Improve
Run the code with Dash and compare to criteria (test).

What changes do you need to make?

Programming for Timing

ASK

What questions do you have now?

What did you learn about planning and coding through this challenge?

____

____

____

____

____

How did your robots know when to turn?

____

____

____

____

How could you make this robot more autonomous (better able to make decisions on its own)? How do you know when to move in a wave? What sensor could you add to allow your robot to collect more input from the environment to respond to?

____

____

____

____

Name ______________________________________ School _________________________

Date ___________________________

Last time we met, you programmed your Dash to participate in a Ripple (forward and reverse spins). Today, work together to **improve** the timing of all the Dashes of the Ripple in both directions and then add the Ripple between the Dash Shuffle and Dash Boogie.

Where is *your group's Dash* in the line-up for the ripple? (There may not be 6 Dashes in your class, so skip 6 if that is the case.)

Forward 1 2 3 4 5 6

Backward 1…2…3…4…5…6

<u>First: Calculate Wait Times</u>

If Dash takes 2 seconds to spin a 360 degree spin, figure out how long each Dash will have to wait before it ripples forward then in reverse?

Forward Ripple (turn right)	**1.** Dash 1 spins right 360 degrees	**2.** Then Dash 2 spins right 360 degrees	**3.** Then Dash 3 spins right 360 degrees	**4.** Then Dash 4 spins right 360 degrees	**5.** Then Dash 5 spins right 360 degrees	**6.** Then Dash 6 spins right 360 degrees
How many seconds does each Dash wait?						
Reverse Ripple (turn left)	**12.** Then Dash 1 spins left 360 degrees	**11.** Then Dash 2 spins left 360 degrees	**10.** Then Dash 3 spins left 360 degrees	**9.** Then Dash 4 spins left 360 degrees	**8.** Then Dash 5 spins left 360 degrees	**7.** Then Dash 6 spins left 360 degrees
How many seconds does each Dash wait?						Wait time? Spin time?

(Note: there may not be 6 Dashes in your classroom, but calculate as if there were 6 in this exercise)

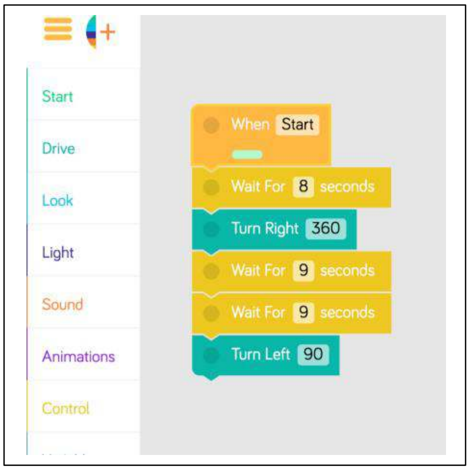
Figure out how long you need to program your Dash to wait in the Forward and Reverse Spins?

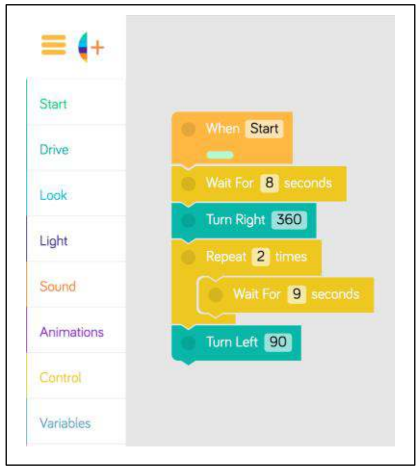
Forward ____________________seconds Reverse________________________seconds

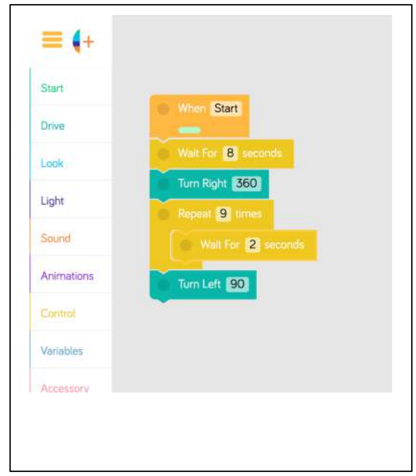
Coding the Wait Command and Testing Options:

See the three different ways to code the wait time for Dash 5 in a ripple. Discuss.

Version 1	Version 2	Version 3







The class use version 3. Why is it important that we all agree on the same version?

Imagine – Plan - Create

- *BE SURE YOUR GROUP IS SAVING CODE – USE "CREATE NEW PROJECTS" in upper left corner*

With your mentor, check the code created last week for the ripple in both directions.

Create a ripple of all the robots spinning completely (360 degrees) to the right, one by one, starting with robot 1, which turns completely before robot 2 turns completely, then robot 3 turns completely before robot 4 turns completely. Then create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1.

Improve (15 - 20 minutes after class begins)

Line up all the Dashes to test and improve the ripple. Below, students will write observations about what improvements are necessary to make the ripple work for all the Dashes in the classroom. Repeat this process as many times as necessary to get the ripple working.

Dash 1	Dash 2	Dash 3	Dash 4	Dash 5	Dash 6

Now, place the ripple between the Dash Shuffle and Dash Boogie, re-coding any portion of this earlier work as necessary.

Dash Shuffle (2 times)
Dash turns right by 90 degrees, moves forward fast by 10cm 3 separate times, turns 180 degrees to the left, moves forward fast 3 separate times by 10 cm each time, then faces forward toward the audience (which is Right turn by 90 degrees).

Use this space to plan:

Ripple
Create a ripple of all the robots spinning completely (360 degrees) to the right, one by one, starting with robot 1, which turns completely before robot 2 turns completely, then robot 3 turns completely before robot 4 turns completely. Then create a ripple in the reverse order, having the robots spin completely to the left, starting with robot 4 and ending with robot 1.

Dash Boogie (repeat 2 times)
Dash looks to the right by 90 degrees, looks to the left by 90 degrees, looks forward toward the audience (looks right 90 degrees), turns a 90 degree angle to the right, turns 180 degrees to the left, faces forward toward the audience (turns right 90 degrees) and spins a 360 degree turn.
(NOTE: Dash looks is a different command from Dash turns)

Now, run the entire sequence with all the Dashes. Make all improvements needed.

Finally, as your last step to finish your dance for the December 2 Dance Presentation, your team will create an original algorithm that you collaboratively imagine, plan, create, and improve. There will be 4 parts to the final dance: 1) Shuffle, 2) Ripple, 3) Boogie, 4) Original sequence you create.

This original algorithm must include:
1. At least one repeat function,
2. At least one wait function,
3. Any combination of lights and sound.
4. The Dash should not go further than 30 cm in any direction.

Your robot will not interact with the other robots except during the ripple.

This will be the finale to the performance for December 2, when you and your family are invited back to USC. This finale will end the Dash's sequence with your team's unique coding. This final sequence should not be longer than 30 seconds, total.

ASK
What is the problem/challenge?

Imagine
Ideas

Plan
Draw a diagram

Create **Code in Blockly**

Improve
Run the code with Dash and compare to criteria (test).

Test your algorithm at least two times before the end of the last class. What changes do you need to make?