AN ABSTRACT OF THE THESIS OF

Asma Alghamdi for the degree of Master of Science in Computer Science presented on June 10, 2019.

Title: The Effectiveness of Using Robots in a Computer Science Orientation Class

Abstract approved:

Jennifer Parham-Mocello

One goal of using robots in introductory computer science (CS) courses is to improve motivation among learners. In this study, we investigate the effectiveness of using the Cozmo and Lego Mindstorm robots to improve students' motivation in a CS orientation course, and we describe our experience using these robots in the lab. To evaluate the effectiveness, we asked participants in pre- and post-surveys about their motivation to learn about CS and programming due to the robots, as well as follow-up questions about the different robots and their engagement in the class because of the robots. Even though our results show students significantly more motivated to use the robots to learn about CS and programming before taking the class, this might be due to the heavy use of the Lego Mindstorm, which students rated less motivating than the Cozmo. While the use of robots negatively impacted students' motivation, their overall interests in CS and coding decrease similarly to students in a course without robots.

The Effectiveness of Using Robots in a Computer Science Orientation Class

by
Asma Alghamdi

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 10, 2019
Commencement June 2020

Master of Science thesis of Asma Alghamdi presented on June 10, 2019

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Asma Alghamdi, Author

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Jennifer Parham-Mocello, my research supervisor, for her patient guidance, enthusiastic encouragement and useful critiques of this research work.

I would also like to thank her advice and assistance in keeping my progress on schedule.

My grateful thanks are also extended to all the sponsors in the graduate writing center for their help to improve my writing skills specifically Jason Schindler.

Nobody has been more important to me in the pursuit of this project than the members of my family.

I would like to thank my parents; whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

Most importantly, I wish to thank my loving and supportive husband, Abdulkarim, and my two wonderful children, Osama and Seba, who provide unending inspiration.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

Chapter 1 Introduction

Teaching programming to students requires creative approaches to make students more interested in choosing computer science (CS) as a field of study. Studies show that the use of robots in teaching and learning activities motivates the students to learn more in the class, as well as arouse their curiosity to explore and experiment with other topics (McGill, 2012). Supplementing these findings, other studies show that using Lego Mindstorms provide a motivated learning environment to students spurring their critical thinking capabilities (McWhorter & O'Connor, 2009). Introducing robots into CS courses is important in order to improve student learning (Fagin and Merkle, 2003). It is noticed that working with robots in class encourages intrinsic motivation, problem-solving and creativity among students (McGill, 2012).

For these reasons and several others, we introduced the application of programming robots to the labs in the CS Orientation course in fall 2018. The labs in the past lacked real-world relevance, and the use of robots in the lab provides the experience of testing and seeing what is taught in the CS orientation course. In addition, attaining a learning outcome such as teamwork is challenging to achieve, and research shows that this is due to the absence of an affirmed interesting CS class (Thobbi & Weihua, 2010). The research continues to explain that since most programming assignments support the lecture materials and many students face difficulties relating to real-world applications, the incorporation of robots in classes can improve the educational experiences. For instance, students learning with real robots, relate to the concept of being taught and can apply the skills they learn in their actual life (Thobbi & Sheng, 2010).

Research demonstrates that using robots lead to an increase in the effectiveness of teaching programming in computer science (Fagin and Merkle, 2003), but it was noted by Imberman and Klibaner (2005) that it's quite challenging to make an introductory CS class captivating or interesting to students in a computer laboratory session. Nevertheless, the research in this paper empirically represents the impact of using robots in the labs in a CS orientation course and provides an evaluation of that impact by analyzing students' motivation, interest, and engagement in computer science and programming before and after using the robots, as well as students’ attitudes toward using the different robots in the lab.

Chapter 2 Background

Many research studies support using robots in CS1 (Introductory to Computer Science classes) to address the issue of motivation and retention, as students find relevance in what they learn, and practical education can reduce the chances of forgetting some concepts (Shiomi et al., 2015). The ability of the students to apply the concepts learned in programming the robots gives the teacher feedback about how well students capture the knowledge and progress with their class studies (Rutten, Joolingen, & Veen, 2012). Another research document shows that incorporating a robotics learning approach in CS1 courses reduces the chances of a student skipping classes; the experiential learning and program experience makes learning interesting rather than the traditional method of learning theory for a prolonged period (Eguchi & Uribe, 2017). In other words, the use of robots in classrooms enhances cooperative learning among students; students engage with one another to foster their relationships and increase the urge to help each other in complex concepts.

The practical skills that are taught help students in the application of knowledge in real life situations. Several authors suggest that the relevance of robots in real life situations, and they concluded that it helps students by raising their curiosity and eagerness to learn (Sternberg & Lubart, 1999; Lauwers, Nourbakhsh, & Hamner, 2009). Research also indicates that robots could be effectively used to teach computer programming because students tend to view the lessons as relevant and practical (Lauwers et al., 2009).

Besides, one of the critical reasons of integrating robots in class learning is to tap on the student's creativity and innovative nature (Apioal, Lattu, & Pasanen 2010). Creativity can be brainstorming, mind mapping, or a new way of solving the problem, and Zawieska and Duffy (2015) find that the interaction of students with robots from the early stages of learning boosts their creativity. Students mostly use robots during lab hours in which students are given the opportunity to perform experiments, and the Cozmo and Lego Mindstorm are examples of robots used in classes to teach students programming. Cozmo is an assembled robot used to teach automation (Choi & Lee, 2003). In contrast, Lego Mindstorm is a platform system with which students build their own robot. A few researchers show that exposure to robots during laboratory time promotes experimentation beyond their required activities, which gives the students the opportunity to be creative (Huang, Yang, & Cheng, 2013). The researchers further note, this creative approach leads to the development of new solutions to problems that are more efficient in terms of time and the resources used. Consequently, the introduction of robots to students' academic life at an early age might enable them to discover more knowledge and skills besides completing their coursework.

One of the studies reveals that the use of robots in classrooms reduces the possibility of omitting some parts of the course since the curiosity of students to manipulate the robots makes it almost impossible to ignore some syllabus requirements (Ryan & Deci, 2000). Robots also encourage cooperative learning, and the process of students engaging in helping each other fosters their ability to perform well in social activities while reinforcing their knowledge (Meisalo & Lavonen, 2000). Based on their prior research, we believe using robots in teaching computer science is an effective strategy for boosting the motivation of students to learn how to program. Our goal is to use robots in the labs to help students learn the concepts from lecture in a different way that will build their creativity in programming, as well as problem-solving ability, with different programming environments and different type of robots to reinforce computational concepts.

Apart from enhancing motivation among students, the benefit of using robots for students in learning influences their attitudes towards robotics and programming (Markham & King, 2010). Evidence from McGill (2012) shows positive interactions with robots help the students to develop their attitudes towards programming in a CS0 class. Most of the students believed that programming is complex and its application in robotics makes it even harder; however, a prior introduction to robots proves to them otherwise (McGill 2012). Another research study suggests that students using robots in a class spend extra time on exercises related to the course work but not required; this work was self-directed by the students showing their intrinsic motivation and interest in the study (Markham & King, 2010). Students engaged in practical sections of their coursework to reinforce the clarity of the concepts taught. Repetition created by the use of robots in programming lessons helps the students to master the skills learned, thereby changing their attitudes from fearful novices to skilled programmers (Kandlhofer & Steinbauer, 2016). These researchers also show that the use of robots increase the effectiveness of teaching programming in computer science by increasing students' conceptual understanding of programming topics and retention.

However, our goal is to investigate the effectiveness of using the Cozmo and Lego Mindstorm robots to improve students' motivation in a CS orientation course. With this goal in mind, this research study answers the following research questions.

Impacts of Robotics on Motivation:

- *RQ 1: Does the use of robots improve students' motivation to learn more about CS and programming/coding?*
- *RQ 2: Does this motivation differ among males and females?*
- *RQ 3: Are students more motivated using the Cozmo or Lego Mindstorm robot?*

Follow-up Robotics Questions:

- *RQ 4: Are students interested in using robots before the class, and how well do the robots engage them in the class?*
- *RQ 5: Which robot do students enjoy using and programming more?*
- *RQ 6: Should we use Scratch, Python or both with the robots in a university CS orientation class?*

Curriculum Impacts on Student Interest:

- *RQ 7: Does this new robotics course improve students' interest in CS programming and the class?*
- *RQ 8: How do students' interest in this new course compare to students' interest in the traditional offering of the course?*

Chapter 3 Materials and Methods

In fall 2018, Oregon State University received funding to offer a section of the CS Orientation course that provides a real-world experience in the labs using robots, 3-D printing, and interdisciplinary teams of CS and mechanical, industrial, and manufacturing engineering (MIME) students. Prior to the course, we obtained IRB approval to collect participants from the CS Orientation course and analyze student responses to pre and post surveys to evaluate the new curriculum and use of robots in the lab.

3.1 CS Orientation Structure

The College of Engineering at Oregon State University requires students to take any engineering orientation class to achieve a BS degree, and the university offers Computer Science Orientation (CS 160) once a year in the fall quarter to meet this requirement for students interested in majoring in CS. Since a CS student can take any engineering orientation course, this class is not a prerequisite for any other class, and since any other major can take this class, the class does not have any prerequisites. Primarily entering first-year students who declare CS as their major take this course, but any student outside the major and at any level may also take the course.

The prerequisite structure of this class makes it the ideal place to introduce robotics into the curriculum, which aligns with existing research that suggests developing a course introducing robotics that requires no prerequisites to ensure that all the students are adequately orientated to computer science regardless of their academic history (Burhans, 2007). Other research suggests that the use of personal robots in the introductory stages of programming courses with non-computer science students boosts the motivation of CS and non-CS students to learn to program (McGill, 2012). Our primary goal of this new course is to introduce the Cozmo and Lego Mindstorm robots in the lab curriculum to teach students basic programming concepts, including conditional execution, repetition, functional decomposition, and array/lists.

3.2 Procedure

In the Fall 2018, we created a new section of the CS Orientation course using the robotics equipment combined with MIME students in the labs, and we continued to teach another section in the traditional way it has been for the past 5 years. At this point, we will refer to these sections as the non-traditional and the traditional sections. Both sections had two one-hour lectures and one two-hour lab each week, and the lectures primarily consisted of presentation slides to teach computational concepts with a live coding demonstration through a terminal to teach the Python syntax. Both courses had 10 weekly assignments with programs to write using the Python language and exercises to orient students to the CS major at Oregon State University and the CS field in general.

In the two-hour lab, the traditional section focused on teaching Python programming concepts that would help with their current assignment in the class and problem-solving activities that concentrated on writing pseudocode, analyzing the code or finding a solution to a problem, and the students have the option of working in pairs or individually for each lab. The robotics labs focused on small team (3-4 students) activities each week and a term-long final project that applied concepts to real-world problems using robots, 3-D printing, and interdisciplinary teams of CS and MIME students (see Table 1).

For example, one of the labs examined pathfinding algorithms for detecting objects. The learning objectives of this lab were to understand pathfinding algorithms, to manipulate variables and arrays, and to create and use functions. Another lab activity helped students learn about design solutions for a prototype ramp based on each pillar of sustainability (economic, environmental, and social) and develop new code to program Cozmo to drive up the ramp and collect data to support their ramp evaluations. The labs and final project required the use of Lego Mindstorm more than Cozmo (see Appendix A for full details).

Table 1: Labs Structure.

Week 1:	**Team building, and introduction for Cozmo as well as Lego Mindstorm.**
Week 2:	**Introduction to Variables, expressions, and input in Python.** Ramp design part 1 using 3D printing. Students will assess the design requirements based on each pillar of sustainability (economic, environmental, and social) and propose a design solution for a prototype ramp. Also, go through the TinkerCAD tutorials, download and install Dremel DigiLab. Learning how to import 3D model in Dremel DigiLab to get the amount of material used for 3D printing.
Week 3:	**Introduction to gathering and evaluating data.** Ramp design part 2 the MIME students will build the design and make adjustment while the CS students will develop a code to program Cozmo to drive up the ramp and collect data from the Cozmo to evaluate the ramp's performance. Also, Lego installation; students following the tutorial to build the driving base of the robot and add a color sensor.
Week 4:	**Introduction to using sensors in motion planning for robot-object interaction.** Forklift part 1, the goal was to design a moving base for the robot and use an ultrasonic sensor to locate an object, use a gyroscope to maintain movement in a straight line then uses a forklift to pick up and transport objects from one location to another which involved using Cozmo and Lego Mindstorm. At the end of the lab student's Complete 3D model box.
Week 5:	**Introduction to bar-linkage mechanism and it is implementation in robot-object interaction and continue learning conditional execution and repetition.** Forklift Part 2, the goal of this lab was learning how to program robot to perform pre-planned motion and to design and operate the system (Forklift) that will pick up and carry the 3D model box that students completed in lab four. Students will modify the algorithm to move the Lego Mindstorms in a straight line, detects box, picks it up, transports it and put it back on its designated platform. Also, students return to the Cozmo task from lab four modify the code so that Cozmo looks around and searches for the cube sets down the lightcube and unlocks at the end of the track.
Week 6:	**Introduction to pathfinding algorithms.** The goal was to focus on part of a pathfinding algorithm for detecting objects, create variables, and arrays and use Functions. Testtube collection part 1 students will working on building a robot using Lego Mindstorms that will pick up trash on the ground. It must search and locate a cup in an undefined test field and pick it up. Student will use the gyro sensor and the ultrasonic sensor to collect data.
Week 7:	**Continuation of pathfinding algorithms and mechanical component design.** The goal was implementing the pathfinding algorithm and incorporating a claw to your robot so that it can pick up the object. Testtube collection part 2 students will use mechanical design principles to design a claw, use 1D arrays and functions, implement a pathfinding algorithm, and execute motion-planning to grab the object.
Week 8:	**Work on Project.** The goal was to develop the first draft of 3D printed pill hopper design for the final project.
Week 9:	**Holiday Week: Work on Project**
Week 10:	While the final project was AID design, which is local non-profit, has requested MIME101/CS160 students to design a pill sorting mechanism for visually impaired patients surrounding the Oregon State University campus. Design teams must create a fully-autonomous system for sorting colored pills into correct corresponding containers. Robots must begin operation at the center of the test field. Students may use Cozmo, Lego Mindstorm, or a combination of both. Pill container and test field dimensions are standardized (see Appendix A).

3.3 Participants

The non-traditional section using robots had 117 consenting participants, and the traditional section had 80 consenting participants. Out of those 197 consenting participants, 157 took both the pre and post survey (64 from the traditional and 93 from the non-traditional section). These students are from different genders, majors (computer science, electrical engineering, psychology, etc.), and class standing (freshman, sophomore, junior, and senior), and they self-selected into one of the different thematic sections of the CS Orientation course (CS160) in Fall 2018 at Oregon State University. We asked students in both sections to participate in the study during the first week of the quarter, at which time we also asked them to take the pre-survey. Table 2 shows the breakdown of the different demographics in each class.

Table 2: Demographics of consenting participants.

Demographics	**Non-traditional section**	**Traditional section**
Male	84 (72%)	69 (86%)
Female	33 (28%)	11 (14%)
Pre-Computer Science	101 (86%)	60 (75%)
Non-Pre-Computer Science	14 (12%)	19 (24%)
Freshman	56 (48%)	47 (59%)
Sophomore	40 (34%)	24 (30%)
Junior	10 (8%)	7 (9%)
Senior	4 (3%)	2 (2%)

3.4 Materials

To evaluate the effectiveness of using robots to improve students' motivation in the non-traditional robotics section, we asked participants in pre- and post-surveys about their motivation to learn about CS and programming due to the robots, as well as follow-up questions about the different robots and their engagement in the class because of the robots. We also asked students in the non-traditional robotics section and the traditional section about their interests in the CS and programming before and after the class to compare the responses from students in the two different courses. Here are the questions asked in the surveys (see Appendix B and C) to answer our eight research questions about the impacts on motivation, the robots, and interests:

Impacts of Robotics on Motivation:

- How would/did the following affect your motivation to learn more about CS? - Using robots in the labs.
- How would/did the following affect your motivation to learn more about programming/coding? - Using robots in the labs.
- In this class, how motivated were you to learn about
  - Cozmo?
  - Lego Mindstorm?
- How much did the following motivate you to learn how to program/code?
  - Cozmo?
  - Lego Mindstorm?

Follow-up Robotics Questions:

- How would the following affect your interests in this class? - Using robots in the labs.
- How did the following affect your engagement in this class? - Using robots in the labs.
- Which robot did you enjoy using more?
- Which robot did you enjoy programming more?
- How much did the following motivate you to learn to program/code?
  - Python
  - Scratch
- How much did the Scratch help you understand Python?
- Which language did you prefer for programming Cozmo?

Curriculum Impacts on Student Interest:

- Please rate your interest level in the following.
  - Learning more about computer science.

- Learning more about programming/coding.
- Majoring in computer science.

We developed a new lab curriculum for the non-traditional section using the Cozmo and Lego Mindstorm robots with real-world problems requiring the engineering of a new 3-D object and interdisciplinary teams of MIME and CS students to solve (see Appendix A). We used the Lego Mindstorm because it provides students the ability to design and build robots of different shapes and sizes and 3-D print new pieces easily, and we chose the Cozmo robot because it has more advanced features, such as a camera for facial recognition, a speaker and microphone, and a screen for displaying, and advanced programming features for the CS students. We used the block-based interface for both robots and Python programming for Cozmo. The block-based language for Cozmo is based on Scratch; whereas, the Lego Mindstorm uses the EV3 programming blocks. Block-based interfaces for programming remove syntax from the language and require selecting and connecting blocks representing the instructions to create a program. Python is a scripting language requiring the user to understand programming instructions and the syntax for the language.

3.5 Apparatus

We used the Cozmo robot and Lego Mindstorm to identify whether robotics can influence students’ motivation, interest, and engagement in the class. They used these robots during their 2-hour lab time. Cozmo is a palm-sized robot that students do not have to build, and programming the robot is achieved by selecting instructions from the Cozmo library. Cozmo can move in any direction, recognize faces, talk, lift objects, and display information on a small screen interface, while the Lego Mindstorm is a flexible platform that allows students to build their own robot (Figure1). The Lego Mindstorm has sensors and variables that can perform basic operations, such as driving, turning, iteration, and abstraction (how to store something like color). We also provided 3-D printers for students to design new objects and pieces for the Cozmo and Lego robots.

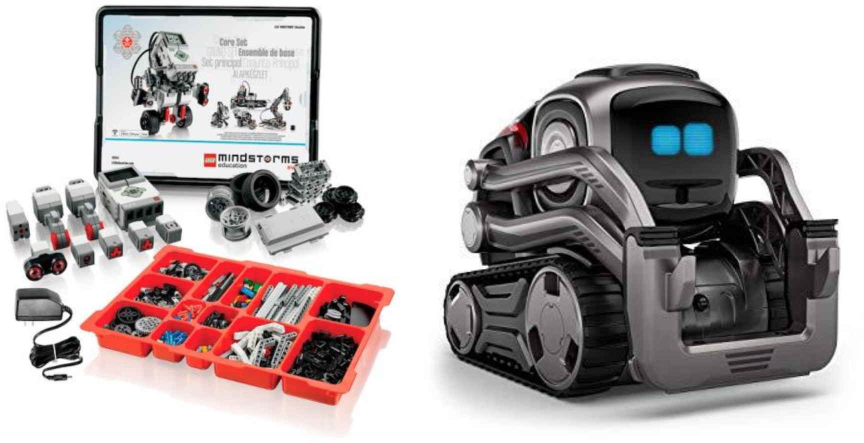


Figure1: Show the Difference Between the Two Robots. Note that the left robot is the Lego Mindstorm and the right robot is the Cozmo.

3.6 Design

Our study was a non-experimental design because it does not have a control and experimental group differing in only one treatment (the robots), which means it is considered an observational study. We do not statistically compare the two sections because there are many differences in the experimental, non-traditional section that we could not control, such as different instructors, different teaching assistants in the labs, working with MIME students, and 3-D printing. Instead, we evaluate students' interest, engagement, and motivation using the robotics in the non-traditional section, and we evaluate students' interest in learning more CS, learning more programming and majoring in CS in the other course to make some observational claims.

The survey questions consisted of a 3-, 4-, and 5-point Likert scale with responses of greatly increase, slightly increase, it wouldn't, slightly decrease, and greatly decrease, which are relabeled as 5,4,3,2,1 respectively. Another survey question consisted of a 4-point Likert scale with response of extremely motivated, very motivated, motivated, and not motivated, which are relabeled as 4,3,2,1 respectively. We used the Wilcoxon signed-rank test to evaluate the motivation questions that are the same on pre-and post-survey for the same students to measure students' motivation in the robotics class and side by side boxplot to show the responses, where the y-axis is the ordinal scale of the response and x-axis is the survey question. We use a descriptive diagram to evaluate students' interest at the beginning of the course and their engagement due to the robots at the end of the course, and to evaluate which robots did students enjoy and programming more and which language has impact student’s motivation to learn how to program. We also used the Wilcoxon signed-rank test to compare students’ reactions toward the Cozmo versus Lego robot to provide more understanding of which robot had the most impact. Moreover, we used Kruskal-Wallis rank sum test to evaluate the differences between males and females’ motivation to learn more about CS and programming using robots in the labs, and Wilcoxon signed-rank test to evaluate the differences in general in students’ interest to learn more CS, learn more coding, and majoring in CS.

Chapter 4 Results

The primary goal of this research is to answer the following question.

> *"What is the effectiveness of using robots in the computer science orientation class (CS 160) at Oregon State University?"*

To answer this question, we formulated 8 general research questions, some of which we tested using statistical tests with null hypotheses at a significant level of alpha = 0.05 and 95 percent of confidence, and others we analyzed using descriptive analysis. We test the hypotheses using pre- and post-survey questions designed to measure differences and determine whether there is an improvement in motivation before and after using robots in the lab, as well as overall CS and programming interests after the traditional and non-traditional curriculum. We also evaluated other questions about the robotics curriculum with follow-up questions and descriptive analysis.

4.1 Impacts of Robotics on Motivation:

RQ 1: Does the use of robots improve students' motivation to learn more about CS and programming/coding?

To answer this question, we asked two questions about motivation on a 5-point Likert scale to learn more about CS and programming/coding before and after the robotics course (see section 3.4 and 3.6). We formulated two null hypotheses for each question to test for significant differences.

Hypothesis 1: There is a difference in students' motivation to learn more about CS before and after using robots in the labs.

A Wilcoxon signed-rank test comparing ratings on the pre- vs. post survey reveals that there is a statistically significant difference in students' motivation before and after, such that students were more motivated to learn about CS before using the robots than after, z (-4.691), $p < .001$, $d = -0.34$; d represent the effect size which is small (Figure 2).

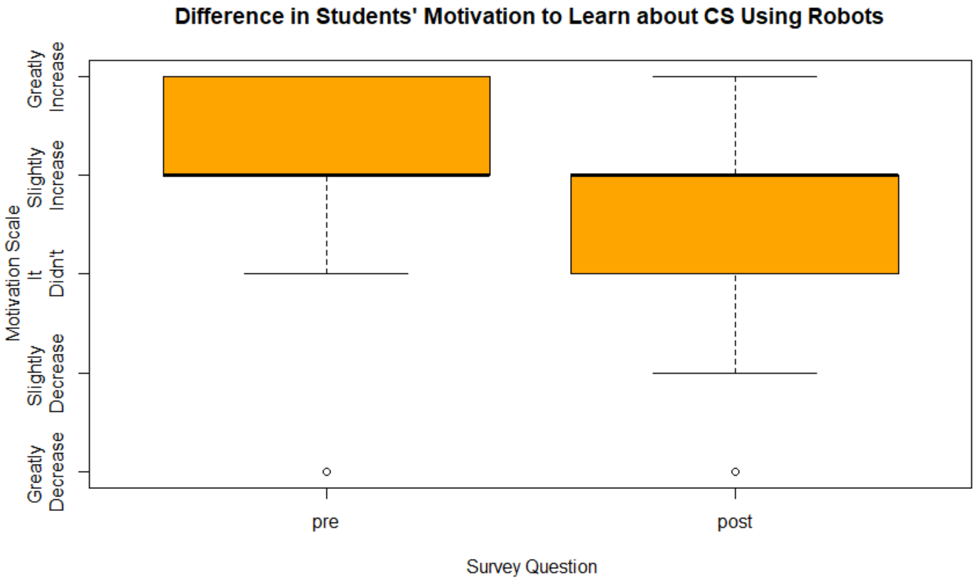


Figure 2: Distribution of the difference between students' motivation to learn more about CS before and after using robots in the labs. Note that the left bar is the distribution of the mean of the student's motivation before using the robots, the right bar is the mean distribution of the student's motivation after use it. The error bars among the confidence interval of the differences lower (0.99) upper (1.15). Additionally, the dots represent outliers.

Hypothesis 2: There is a difference in students' motivation to learn more about programming/coding before and after using robots in the labs.

A Wilcoxon signed-rank test comparing ratings on the pre- vs. post survey reveals that there is a statistically significant difference in students' motivation before and after, such that students were more motivated to learn about programming/coding before using the robots than after, z (-3.844), p < .001, d = -0.28 (Figure 3).

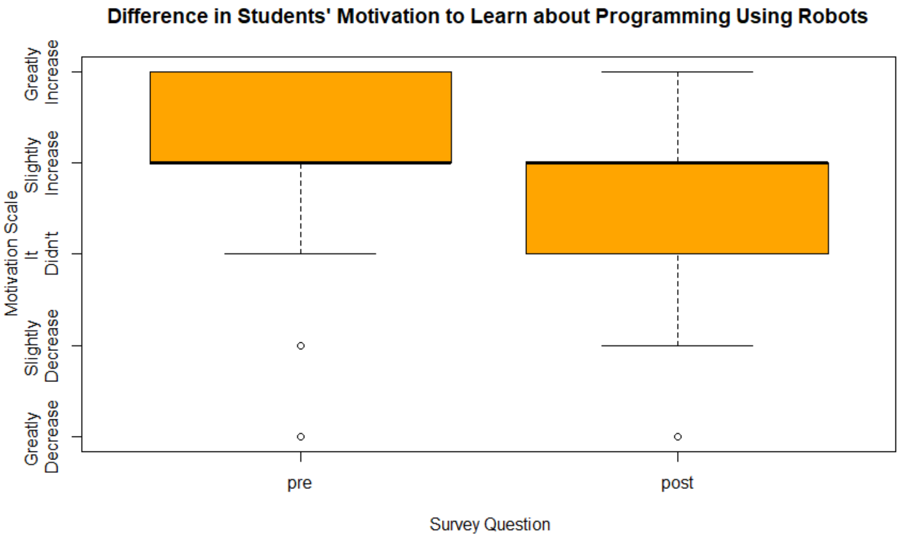


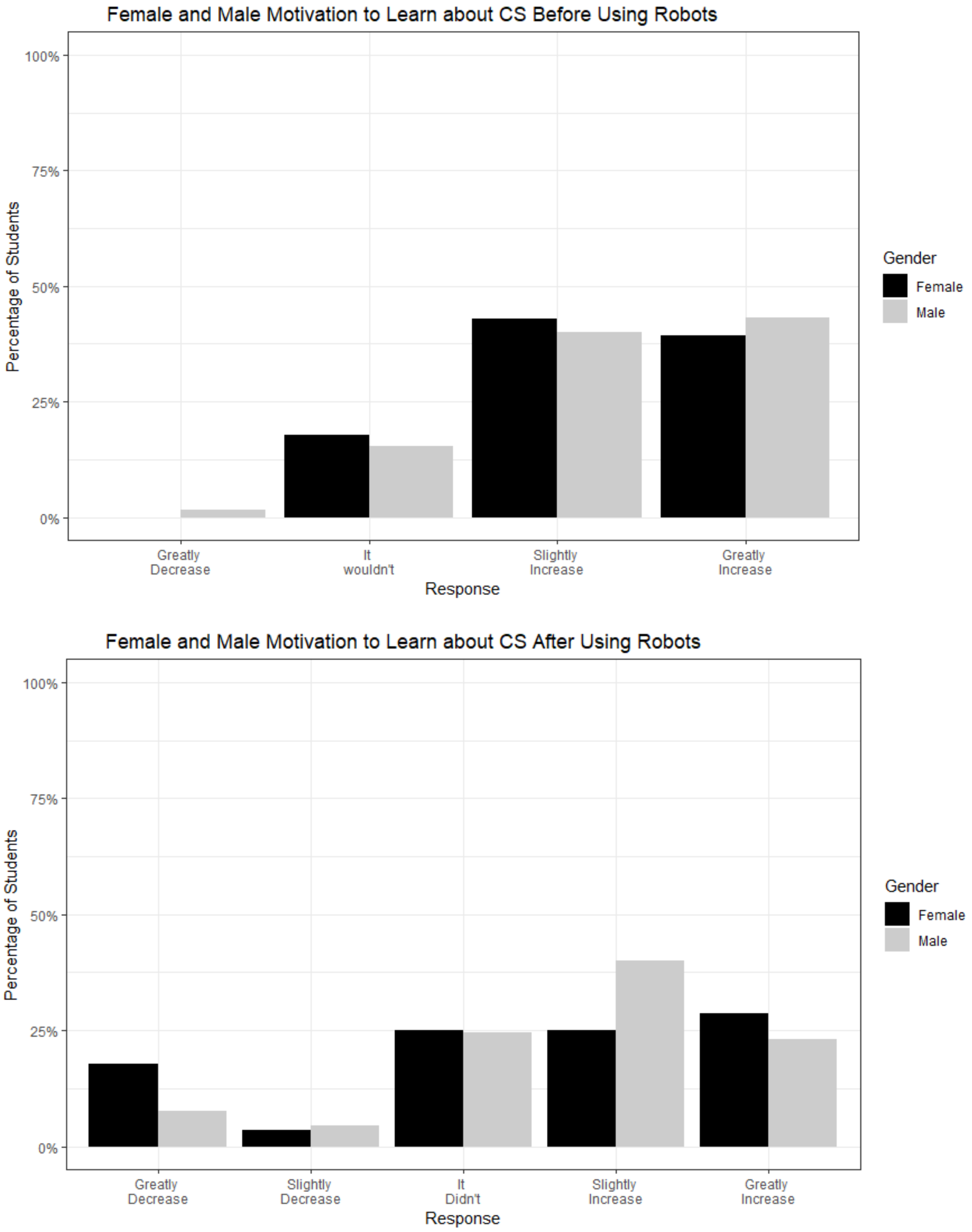
Figure 3: Distribution of the difference between students' motivation to learn more about programming/coding before and after using robots in the labs. Note that the left bar is the distribution of the mean of the student's motivation before using the robots, the right bar is the mean distribution of the student's motivation after use. The error bars among the confidence interval of the differences lower (0.49) upper (1.49). Additionally, the dots represent outliers.

RQ 2: Does this motivation differ among males and females?

To answer this question, we grouped students by gender and analyzed their responses about motivation to learn more about CS and programming/coding before and after the robotics course, as well as their changes in responses to motivation. We formulated two null hypotheses for each question to test for significant differences.

Hypothesis 3: There is a difference between males' and females' motivation to learn more about CS using robots in the labs from the pre to the post-survey.

A Kruskal-Wallis rank sum test showed that there is not a significant difference between males and females' motivation to learn more about CS using robots in the lab, *chi-squared = 0.54077, p=0.46* (Figure 4). From the distribution we see that female and male motivation to learn more about CS before using robots was approximately the same, but it is interesting that neither males or females chose the slightly decrease option and no females indicated greatly decrease. We also observe that the female and male motivation to learn more CS after using the robots in the lab is roughly the same, but it is interesting that almost 20% of female participants selected greatly decrease after the class compared to 0% before the class. From that we can conclude that the use of the robots in the labs did not negatively impact the males as much as it did the females.

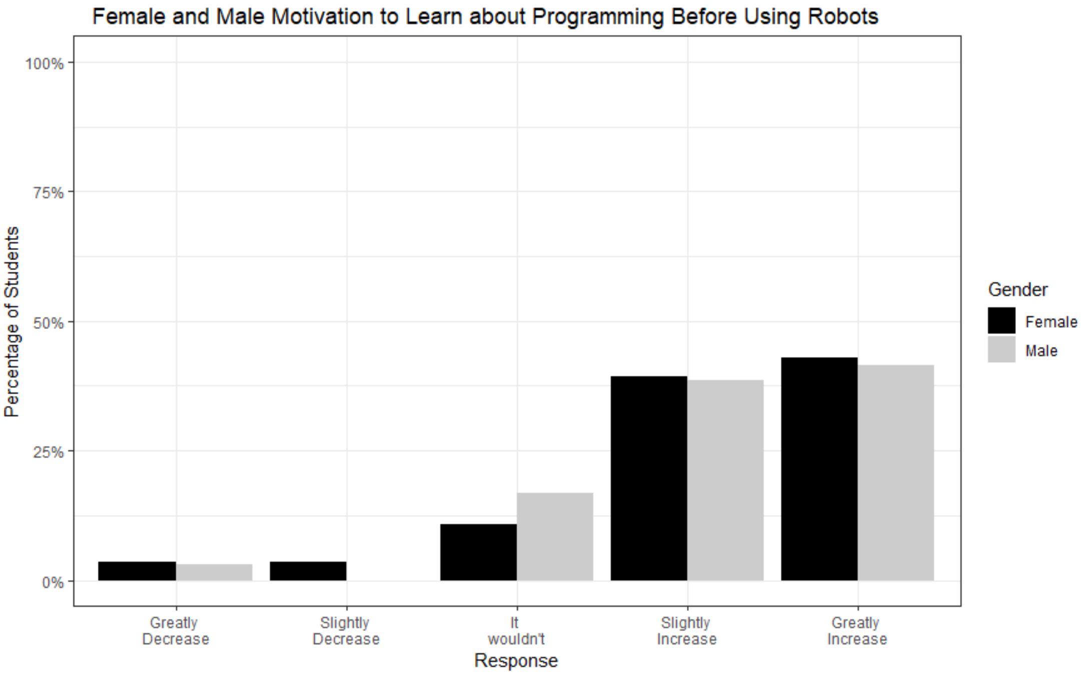

Female and Male Motivation to Learn about CS Before Using Robots

Female and Male Motivation to Learn about CS After Using Robots

Figure 4: Distribution of the difference between males' and females' motivation to learn more about CS using robots in the labs from the pre to the post-survey.

Hypothesis 4: There is a difference between males' and females' motivation to learn more about programming/coding using robots in the labs from the pre to post survey.

A Kruskal-Wallis rank sum test showed that there is not a significant difference between male and female participants' motivation to learn more about programming/coding using robots in the lab from the pre to the post-survey, *chi-squared = 0.031794, p= 0.86* (Figure 5). From the distribution, we observe that females' and males' motivation to learn about programming before using the robots were approximately the same. However, even though more females chose greatly increase after the class, more females also chose greatly decrease.



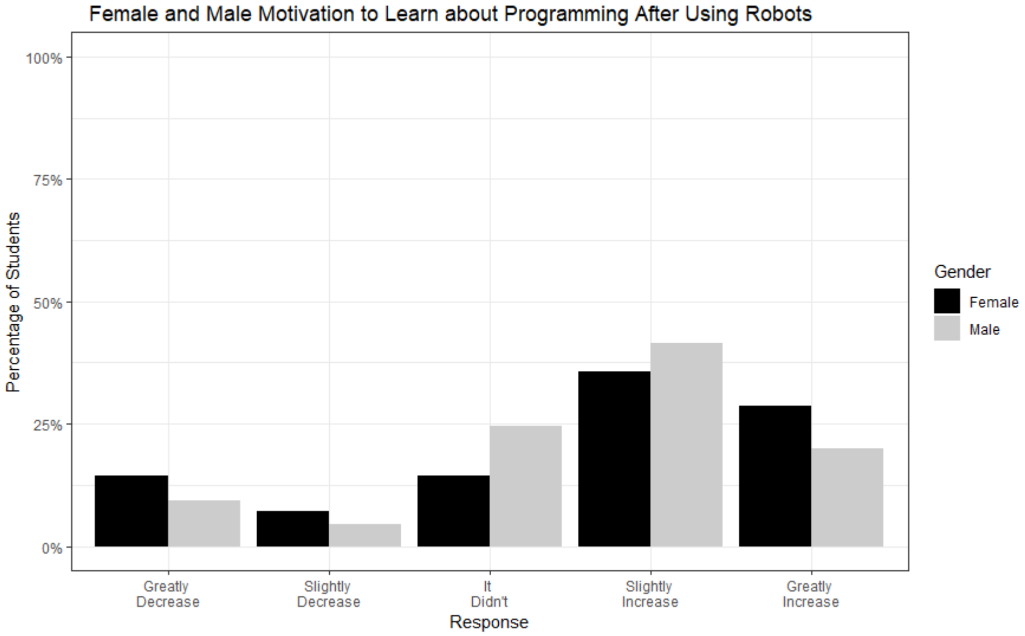


Figure 5: Distribution of the difference between males' and females' motivation to learn more about programming/coding using robots in the labs from the pre to the post-survey.

RQ 3: Are students more motivated using the Cozmo or Lego Mindstorm robot?

To answer this question, we asked two questions about which robot motivated students to use more and learn more about programming/coding. Based on the questions asked in the post survey, we formulated two null hypotheses for each question to test for significant differences.

Hypothesis 5: There is a difference in students' motivation to learn how to program/code using Cozmo and Lego Mindstorm.

A Wilcoxon signed-rank test showed that there is a statistically significant difference in students' motivation to learn how to program using Cozmo and Lego Mindstorm. Students were more motivated to learn to program\code using Cozmo than using Lego Mindstorm, *z* (-6.442), *p* <.001, *d* = -0.47; d represent the effect size which is small (Figure 6).

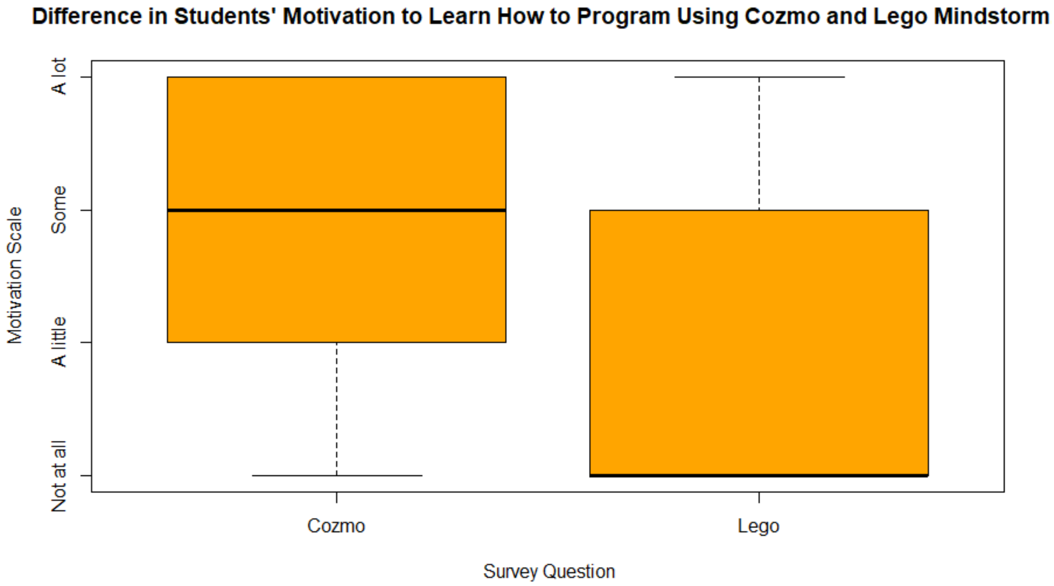


Figure 6: Distribution of the difference between students' motivation to learn more about programming/coding using Cozmo vs. Lego Mindstorm. Therefore, the left bar is the distribution of the mean of the student's motivation to learn program/code using Cozmo, the right bar is the mean distribution of the student's motivation to learn program/code using Lego Mindstorm. The error bars among the confidence interval of the differences lower (0.49) upper (2.45).

Hypothesis 6: There is a difference in students' motivation to learn about Cozmo and Lego Mindstorm.

A Wilcoxon signed-rank test showed that there is a statistically significant difference in students' motivation to learn about Cozmo and Lego Mindstorm. Students were more motivated to learn about Cozmo than learn about Lego Mindstorm, *z* (-6.635), *p* <.001, *d* = -0.48; noted the effect size here is small (Figure 7).

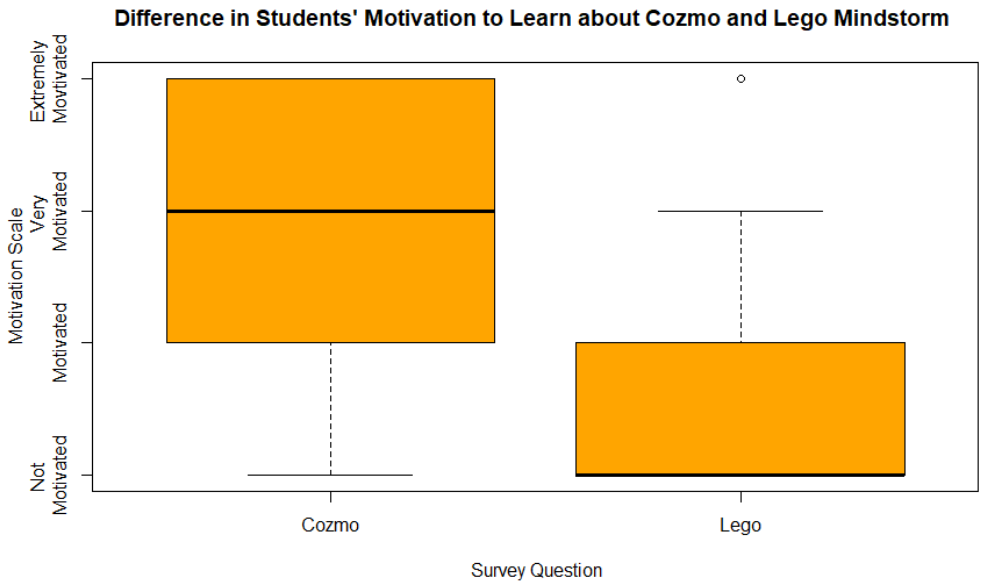


Figure 7: Distribution of the difference between students' motivation in the class to learn more about Cozmo vs. Lego Mindstorm. Therefore, the left bar is the distribution of the mean of the student's motivation to learn about Cozmo, the right bar is the mean distribution of the student's motivation to learn about Lego Mindstorm. The error bars among the confidence interval of the differences lower (`1.49`*) upper (*`1.99`*). Therefore, the dots represent out outliers.*

4.2 Follow-Up Robotics Questions:

Not only do we want to compare students' motivation before and after using robots in the lab, we also want to know if students were interested in using robots before entering the class, if the robots affected their engagement in the class, how the students felt about the Cozmo and Lego Mindstorm robots, and how they felt about the different languages used to program the robots.

RQ 4: Are students interested in using robots before the class, and how well do the robots engage them in the class?

A little over 80% of the students said that the use of robots in the lab would greatly or slightly increase their interest in the class, but only about 70% said using the robots in the labs greatly or slightly increased their engagement (see Figure 8). Actually, almost 20% of the students said that using robots in the labs greatly decreased their engagement in the class; whereas, hardly any students said that the use of robots in the labs would greatly decrease their interest before the class.

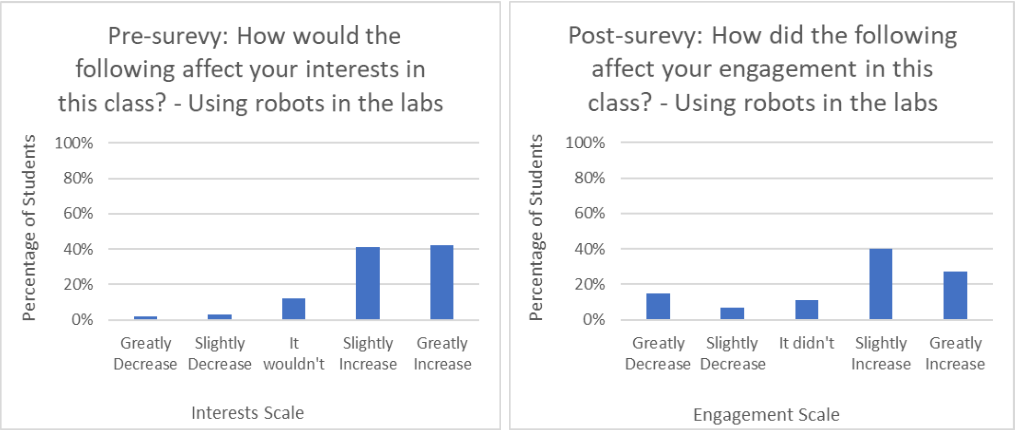


Figure 8: The data shows students' interests in a class in the pre-survey and students' engagement in the post-survey using robots in the labs respectively.

RQ 5: Which robot do students enjoy using and programming more?

Results from descriptive bar graphs show that students enjoy using and programming Cozmo more than Lego Mindstorm (Figure 9). This aligns with student responses that using the Cozmo robot had a positive influence on their motivation to learn more about programming/coding.

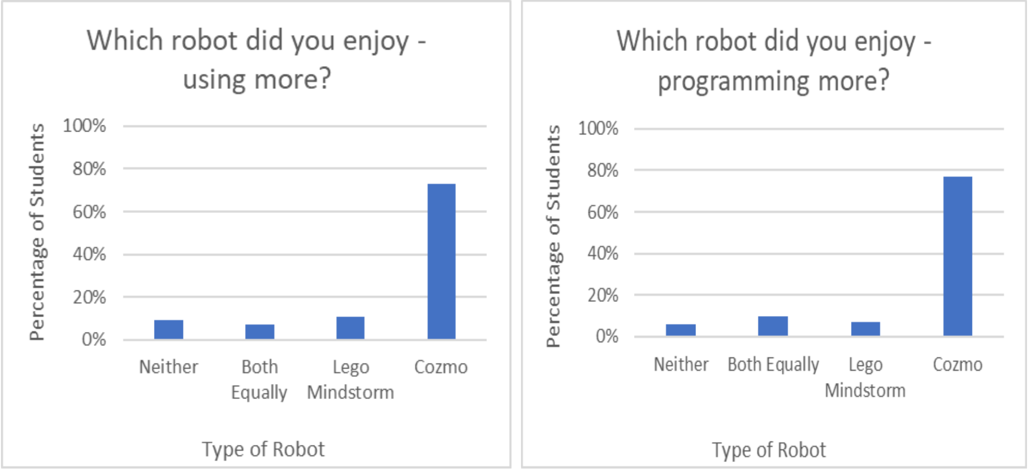


Figure 9: The data shows which robot students enjoy using and programming more.

RQ 6: Should we use Scratch, Python or both *with the robots in a university CS orientation class?*

To answer this question, we asked students which language motivated them to learn to program/code, if using Scratch helped them understand Python better, and which language they preferred using with Cozmo. Python motivated approximately 85% of the students a lot to somewhat compared to Scratch, which only motivated approximately 20% of the students (see Figure 10), and less than 20% of the students think that Scratch helps them understand Python better (see Figure 11). Similarly, over 90% of the students preferred programming Cozmo using Python (see Figure 11).

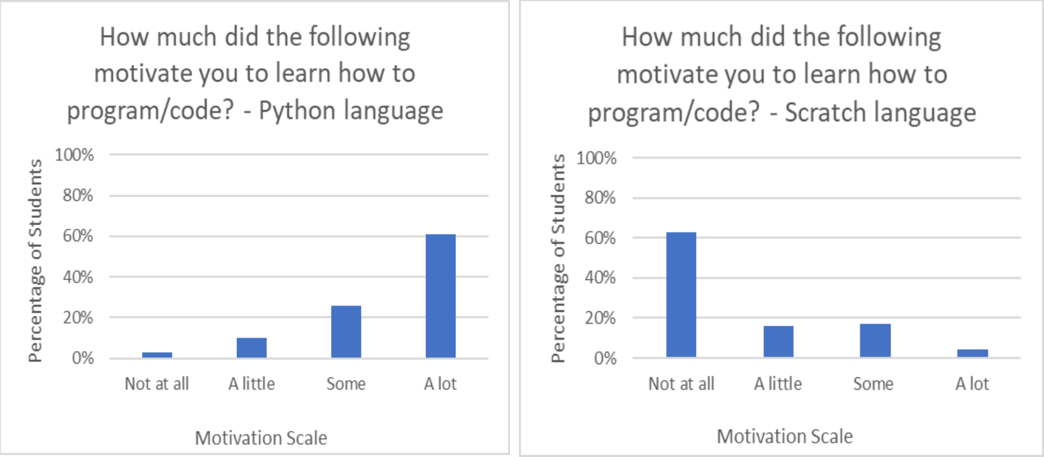


Figure 10: The data shows which language motivate students to learn programming more.

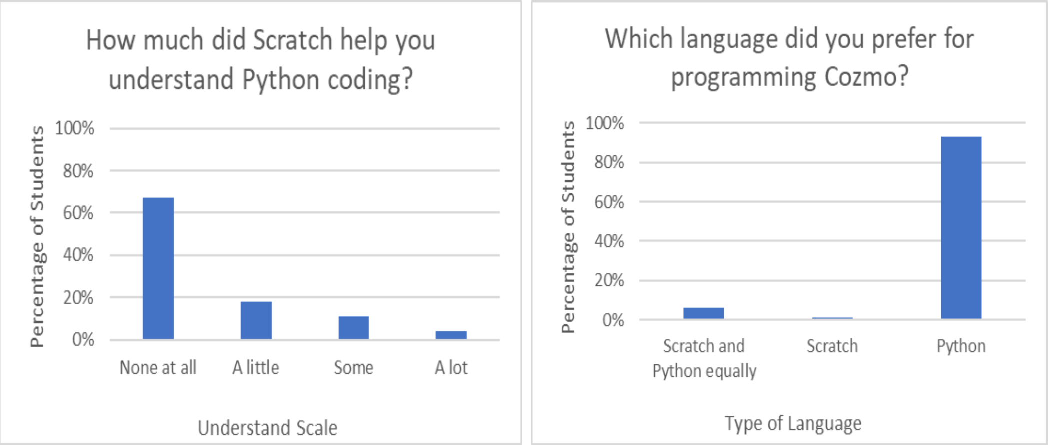


Figure 11: How much Scratch helps students understand Python, and the preferred programming language for Cozmo.

4.3 Curriculum Impacts on Student Interest:

In addition to understanding how using robots in the labs motivate students in a computer science orientation class, we also wanted to know how the new robotics curriculum impacts students' overall interests in CS and programming and how this compares with the traditional curriculum used to teach the class in the past. To answer these questions, we asked students in the new non-traditional robotics section and the traditional section about their interest in the class, learning more about CS, learning more about programming, and majoring in CS before and after taking the class.

RQ 7: Does this new robotics course improve students' interest in CS, programming and the class?

First, we analyze the impacts the new non-traditional curriculum had on the student interests before comparing it to the traditional section. Based on the questions asked in the pre and post survey, we formulated three null hypotheses for each question to test for significant differences. We asked participants in the pre- and post-survey to rate their interest in learning more about CS, learning more about programming/coding, and majoring in CS as a field of study.

Hypothesis 7: There is a difference in students' interest to learn more about CS before and after taking the class.

A Wilcoxon signed-rank test showed that there is a statistically significant difference in students' interest before than after the class to learn more about CS, z (-2.236), p = 0.025, d = -0.2 (Figure 12). As the pre and post distribution shows, there is a decrease in student interest in learning more about CS after the new robotics curriculum.

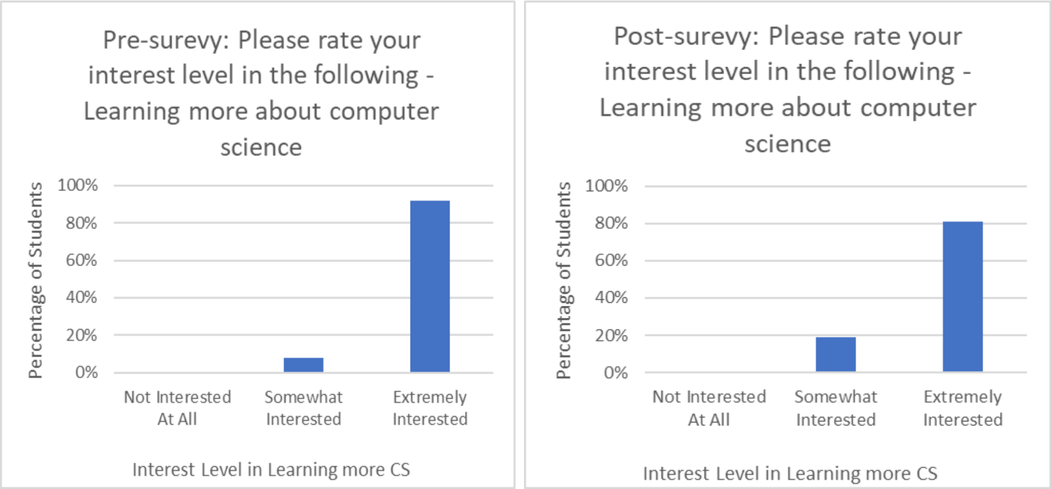


Figure 12: Difference in students' interest in learning more about CS in the pre- and post-survey respectively in the non-traditional robotics class.

Hypothesis 8: There is a difference in students' interest to learn more about programming/coding before and after taking the class.

A Wilcoxon signed-rank test showed that there is a statistically significant difference in students' interest before and after the class to learn more about programming, *z* (-1.964), *p* = 0.05 (Figure 13). The same as hypothesis 7, there is a decrease in interests in learning more about programming/coding after the class.

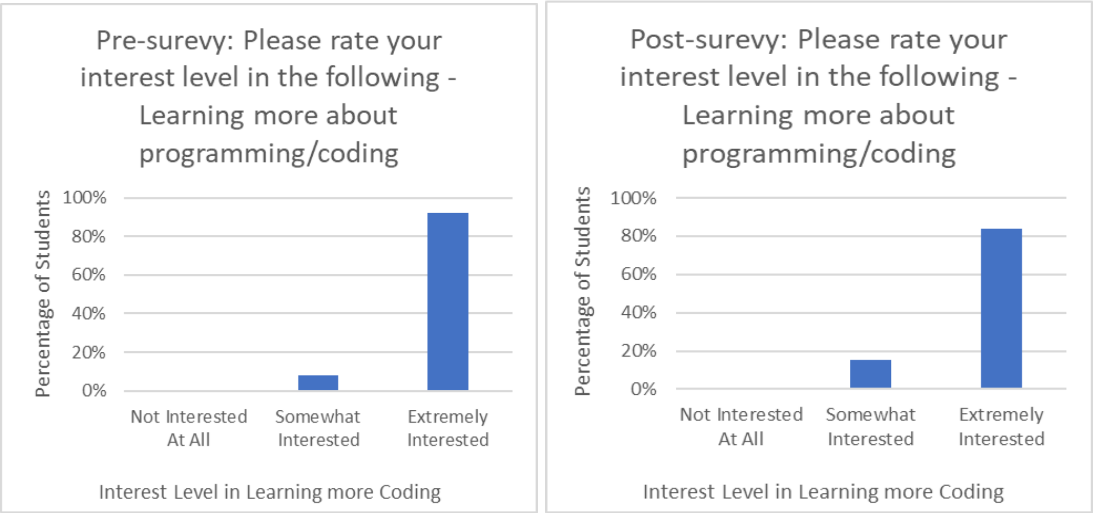


Figure 13: Difference in students interests in learning more about programming in the pre- and post-survey respectively in the non-traditional robotics class.

Hypothesis 9: There is a difference in students' interest in majoring in CS before and after taking the class.

A Wilcoxon signed-rank test showed that there is not a statistically significant difference in students' interest before and after the class to majoring in CS, *z* (-1.877), *p* = 0.06 (Figure 14). However, we see there is still a decrease in their interest in majoring in CS, and the p-value is very close to .05.

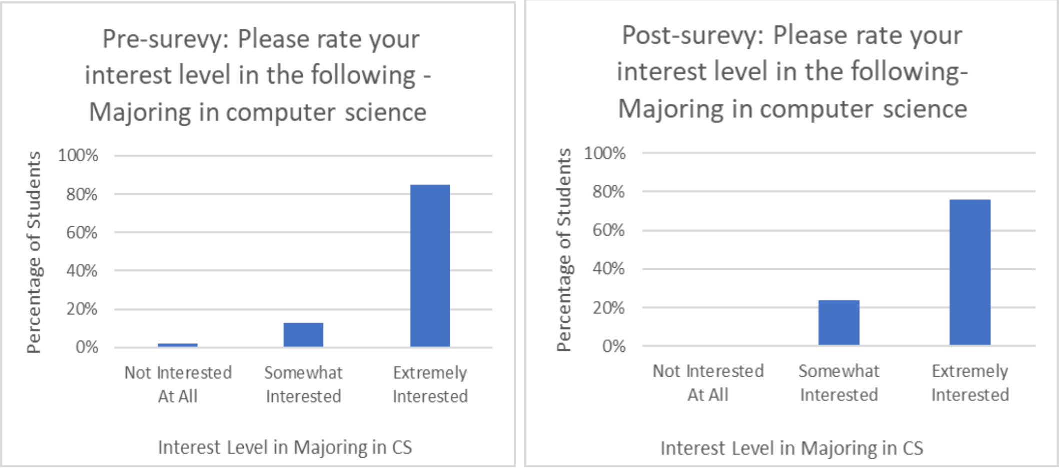


Figure 14: Difference in students interests in majoring in CS as a field of study in the pre-and post-survey respectively in the non-traditional robotics class.

Post-survey results show that most students only somewhat liked the new, non-traditional robotics approach to teaching this class, and a little over 45% of the students dislike the approach somewhat or a great deal (see Figure 15). Even though many students do not like the approach, most students overall liked the class a lot to a moderate amount.

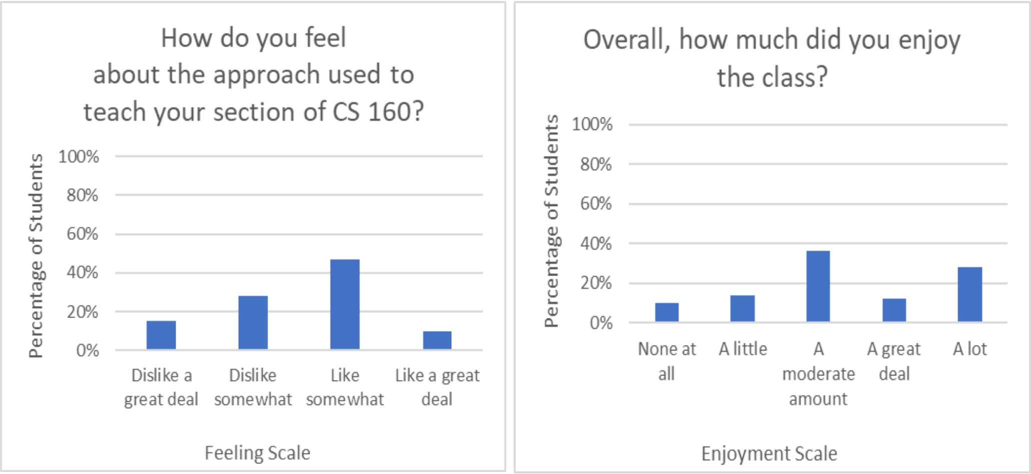


Figure 15: Student feelings about the approach used in teaching the course and overall enjoyment in the non-traditional robotics class.

RQ 8: How do students' interest in this new course compare to students' interest in the traditional offering of the course?

Next, we analyze students' interest in the traditional section to descriptively compare these to the non-traditional approach. We asked the students in the traditional section the same pre and post survey questions, so we tested the same three null hypotheses for each question, as we did for the non-traditional section.

Hypothesis 10: There is a difference in students' interest to learn more about CS before and after taking the class in the traditional section.

A Wilcoxon signed-rank test showed that there is not a statistically significant difference in students' interest before than after the class to learn more about CS, *z* (-1.500), *p* = 0.13 (Figure 16). We can also see that there is not as much of a decrease in interest in the traditional section as there is in the non-traditional section.

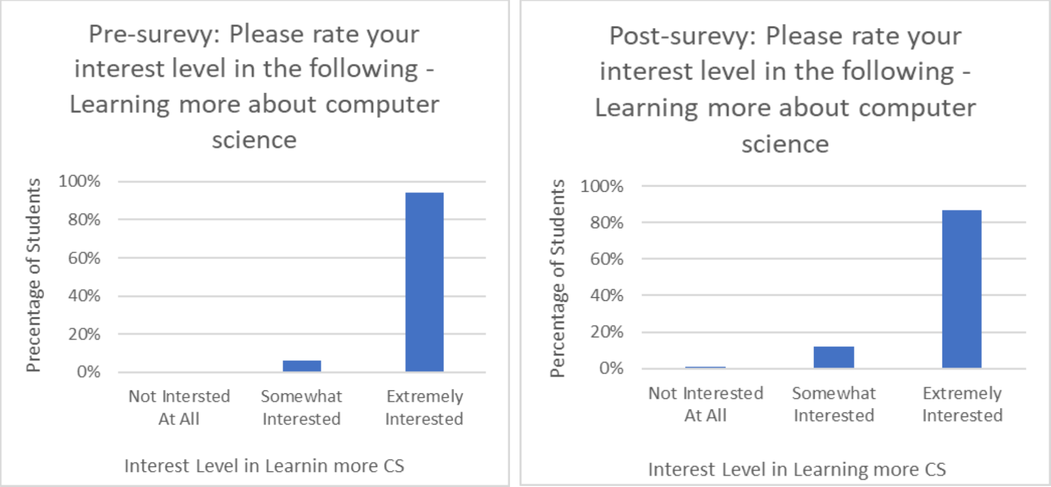


Figure 16: Difference in student interests in learning more about CS in the pre- and post-survey respectively in the traditional class.

Hypothesis 11: There is a difference in students' interest to learn more about programming before and after taking the class in the traditional section.

A Wilcoxon signed-rank test showed that there is a statistically different in students' interest before and after the class to learn more about programming, *z* (-2.138), *p* = 0.03. There was almost a 15% decrease in students extremely interested in learning more about programming/coding after taking the traditional section, which is about the same as the non-traditional class (Figure 17).

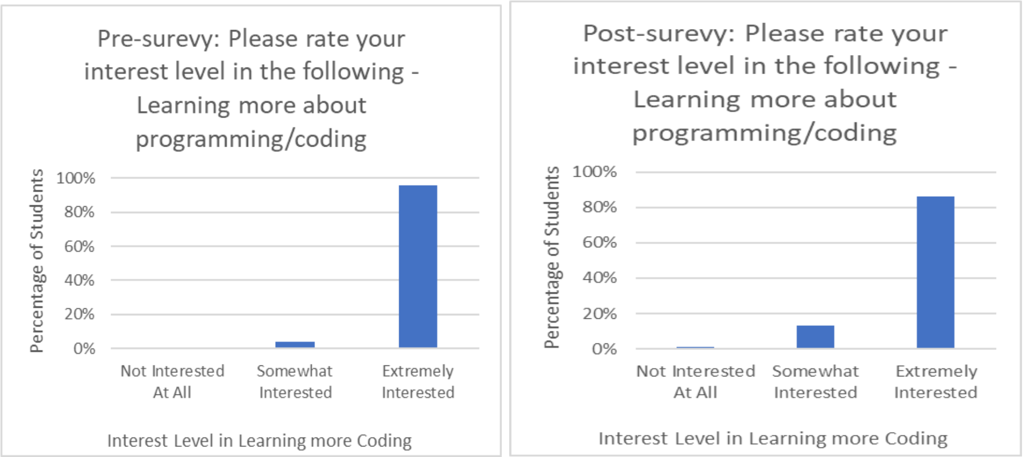


Figure 17: Difference in students interests in learning more about programming/coding in the pre- and post-survey respectively in the traditional class.

Hypothesis 12: There is a difference in students' interest in majoring in CS before and after taking the class in the traditional section.

A Wilcoxon signed-rank test showed that there is not a statistically significant difference in students' interest before and after the class to majoring in CS, *z* (-1.734), *p* = 0.08 (Figure 18). However, it is worth mentioning that there is a decrease in interests in majoring in CS after the traditional section similar to the non-traditional section.

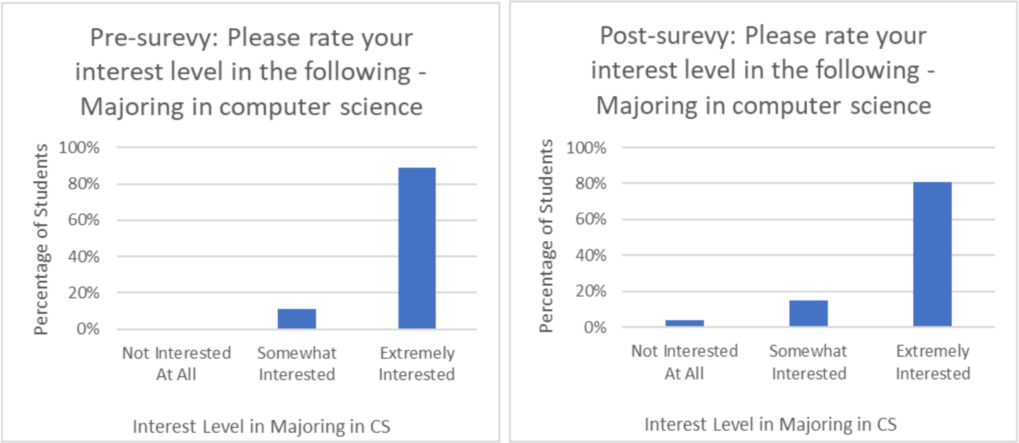


Figure 18: Difference in students interests in majoring in CS as a field of study in the pre-and post-survey respectively in the traditional class.

We also asked students in the post-survey how they felt about the approach used in teaching the traditional section, and their responses show that almost 85% of students like the approach used to teach the traditional section somewhat or a great deal, which is better than the non-traditional section (Figure 19). This is also true for the overall enjoyment of the class, where approximately 85% of the students enjoyed the class a moderate amount to a lot (Figure 19).

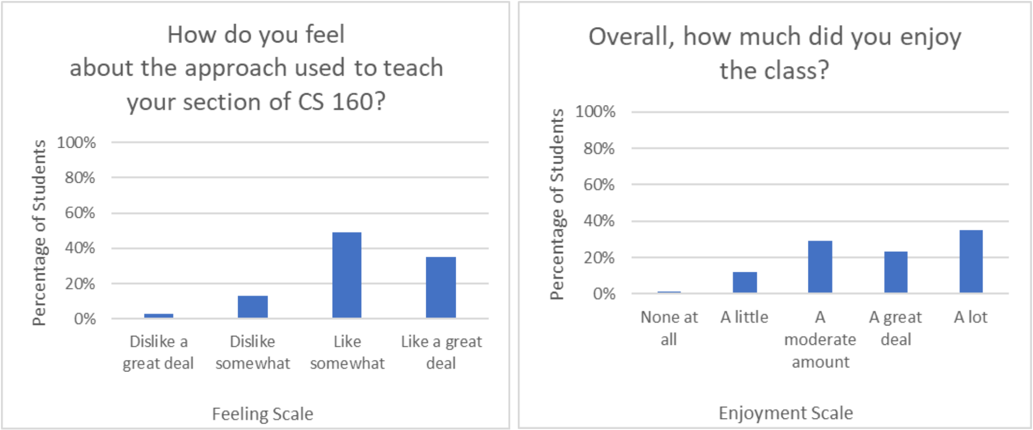


Figure 19: Student feelings about the approach used in teaching the course and overall enjoyment in the traditional class.

Chapter 5 Discussion

The results from this research study suggest that the use of robots in the CS orientation course did not improve student's motivation. Even though our results show students significantly more motivated to use the robots to learn about CS and programming before taking the class, this might be due to the heavy use of the Lego Mindstorm, which students rated less motivating than the Cozmo. Another possibility could be because this was the first time we used robots in the labs, and the lab curriculum and lecture curriculum did not align. This misalignment might be due to combining the labs with the Mechanical, Industrial, Manufacturing Engineering (MIME) Orientation labs to give students an interdisciplinary experience with robotics or it could be because the homework assignments did not use the robots; whereas the labs did.

Even though we did not see that the use of robots increases students' motivation to learn about more about CS or programming/coding, we estimate that Cozmo robots extremely motivate CS students to learn more about coding, more so than the Lego Mindstorms, even though students used Cozmo less in their lab. CS students may like Cozmo more because it is a small, self-contained robot that students do not need to build and is designed to be programmed in higher-level language that is not block-based, while the Lego Mindstorm requires students to build their own robot and program it in a block-based, less-capable language.

The results of this study are similar to the findings in the McWhorter and O'Connor (2009) study, as well as Fagin and Merkle (2003). These studies suggest that the LEGO group is less motivated in learning the material compared to the control group (non-robotics section). More specifically, Fagin and Merkle (2003) found that students who used the Lego Mindstorm did not show as much improvement in learning and retention, as those who did not use it (the control group). Another study found that the Lego Mindstorm does not have an impact on the attitudes of the students towards learning to program compared to the traditional method (Korkmaz, 2016). Since the Cozmo is new, there are less studies using this robot, but recently, Skågeby (2018) researched the interaction of humans with an artificial embodied agent using the Cozmo robot. The study sought to establish the aspect of partner relationships by initiating human-machine relations, and the results established that humans interacting with robots like the Cozmo affected their sense of agency, similar to when interacting with fellow humans (Skågeby 2018). Therefore, structuring from this level, students' interaction with the Cozmo robot might maximize their creativity and awareness of their own actions on the world, as well as increase their motivation to learn more about CS or coding.

Chapter 6 Conclusion

It seems that the different types of robots and language may lead to different results in terms of students' motivation, interests, and engagement in the class. In this study, we asked students not as interested in CS anymore why they changed. Some of the responses suggest that students did not like learning to code using Lego Mindstorm and the block-based interface, and some of the students had a hard time interacting with the Lego Mindstorm. Other responses suggest that the class did not feel like an orientation to CS class for students without any prior knowledge to Python and coding. However, it does not seem that it is the curriculum with or without robots that adversely impacts student interests in CS and coding.

While only a little over half the students like the class somewhat or a great deal, this question is not specific to the robots used in the labs, so it could be because of the 3D printing, combining labs with MIME students, misalignment in the lab and lecture material, etc. Since 70-75% of the students overall enjoyed the class a moderate amount to a lot, this suggests that students benefit from the course regardless of the decrease in motivation. Since almost 30% more students in the traditional Python programming section liked their approach to teaching the class more than the new, non-traditional robotics section, this also suggests that this it was the approach to teaching the class, rather than just the use of robots in the labs, that led to the decrease in students' motivation.

As with any research, there are always limitations. The self-selected sample of students enrolling in the class and then consenting to be in the study could also impact the level of students' motivation to learn more about CS and programming in the post-survey, due to the differences in reasons for enrolling in the class, their majors, and background with programming. We understand that another limitation to this study could be that those who did not consent to be in the study are more motivated or do not differ, instead of becoming less motivated. In addition, we used the Cozmo and Lego robots only during the 2-hour lab time each week, and students might find difficulty meeting the learning outcomes during lab time because their assignments were not related. Another limitation could be the structure of the lab, i.e. time split between the Cozmo, Lego, 3-D printing, and interdisciplinary teams of CS and MIME students, and different tasks assigned for different robots might also affect these results.

In the future, it would be useful to ask the students about their feelings after each task they perform with the robots to understand which tasks and robot are more effective. It would also be beneficial to control for one new factor in the course to see consistent results and compare with another class not using robots. Future research will compare the differences using only the Cozmo (and possibly the Lego Mindstorm) robot in the computer science orientation labs without the 3D printing and interdisciplinary teams of CS and MIME students to research the effectiveness of only introducing robots into the CS orientation labs, as well as compare the results with the other CS orientation course that does not use robots in the lab to reinforce lecture concepts. The results of this study provide a better understanding of the effectiveness of using robots to teach students about CS and programming, and the study also has the potential to apply to high school education.

References

Apiola, M., Lattu, M., & Pasanen, T. A. (2010). Creativity and intrinsic motivation in computer science education. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education - ITiCSE '10*. doi:10.1145/1822090.1822147

Burhans, D. T. (2007). A robotics introduction to computer science. In Aaai spring symposium: Semantic scientific knowledge integration (pp. 20–23).

Choi, M. H., & Lee, W. W. (2003). Quantitative evaluation of an intuitive teaching method for industrial robot using a force/moment direction sensor. *International Journal of Control, Automation, and Systems*, *1*(3), 395-400.

Eguchi, A., & Uribe, L. (2017). Robotics to promote STEM learning: Educational robotics unit for 4th grade science. *2017 IEEE Integrated STEM Education Conference (ISEC)*. doi:10.1109/isecon.2017.7910240

Fagin, B., & Merkle, L. (2003). Measuring the effectiveness of robots in teaching computer science. In Acm sigcse bulletin (Vol. 35, pp. 307–311). doi: 10.1145/792548.611994

Huang, K., Yang, T., & Cheng, C. (2013). Engineering to See and Move: Teaching Computer Programming with Flowcharts vs. LEGO Robots. *International Journal of Emerging Technologies in Learning (iJET)*, *8*(4), 23. doi:10.3991/ijet.v8i4.2943

Imberman, S. P., & Klibaner, R. (2005). A robotics lab for cs1. Journal of Computing Sciences in Colleges, 21 (2), 131–137.

Lauwers, T., Nourbakhsh, I., & Hamner, E. (2009). Csbots: design and deployment of a robot designed for the cs1 classroom. ACM SIGCSE Bulletin, 41 (1), 428–432. doi: 10.1145/1539024.1509017

Jung, S. (2013). Experiences in Developing an Experimental Robotics Course Program for Undergraduate Education. *IEEE Transactions on Education*, *56*(1), 129-136. doi:10.1109/te.2012.2213601

Kandlhofer, M., & Steinbauer, G. (2016). Evaluating the impact of educational robotics on pupils' technical- and social-skills and science-related attitudes. *Robotics and Autonomous Systems*, *75*, 679-685. doi:10.1016/j.robot.2015.09.007

Korkmaz, Ö. (2016). The Effect of Lego Mindstorms Ev3 Based Design Activities on Students' Attitudes towards Learning Computer Programming, Self-Efficacy Beliefs and Levels of Academic Achievement. *Online Submission*, *4*(4), 994-1007

Markham, S. A., & King, K. N. (2010). Using personal robots in CS1. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education - ITiCSE '10*. doi:10.1145/1822090.1822148

McGill, M. M. (2012). Learning to Program with Personal Robots. *ACM Transactions on Computing Education*, *12*(1), 1-32. doi:10.1145/2133797.2133801

McWhorter, W. I., & O'Connor, B. C. (2009). Do LEGO® Mindstorms® motivate students in CS1? *ACM SIGCSE Bulletin*, *41*(1), 438. doi:10.1145/1539024.1509019

Meisalo, V., & Lavonen, J. (2000). Bits and processes on markets and webs. an analysis of virtuality, reality and metaphors in a modern learning environment. Journal Teacher Researcher, 2, 10–27.

Rutten, N., Van Joolingen, W. R., & Van der Veen, J. T. (2012). The learning effects of computer simulations in science education. *Computers & Education*, *58*(1), 136-153. doi:10.1016/j.compedu.2011.07.017

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of EFFECTIVENESS OF USING COZMO AND LEGO MINDSTORM 9 intrinsic motivation, social development, and well-being. American psychologist, 55 (1), 68. doi: 10.1037/0003-066X.55.1.68

Shiomi, M., Kanda, T., Howley, I., Hayashi, K., & Hagita, N. (2015). Can a Social Robot Stimulate Science Curiosity in Classrooms? *International Journal of Social Robotics, 7*(5), 641-652. doi:10.1007/s12369-015-0303-1

Skågeby, J. (2018). “Well-Behaved Robots Rarely Make History”: Coactive Technologies and Partner Relations. *Design And Culture, 10*(2), 187-207. doi:10.1080/17547075.2018.1466567

Sternberg, R. J., & Lubart, T. I. (1999). The concept of creativity: Prospects and paradigms. Handbook of creativity, 1 , 3–15. doi: .org/10.1017/CBO9780511807916.003

Thobbi, A., & Weihua Sheng. (2010). Imitation learning of arm gestures in presence of missing data for humanoid robots. *2010 10th IEEE-RAS International Conference on Humanoid Robots*. doi:10.1109/ichr.2010.5686324

Zawieska, K., & Duffy, B. R. (2015). The social construction of creativity in educational robotics. In *Progress in Automation, Robotics and Measuring Techniques* (pp. 329-338). Springer, Cham.

Appendix A

LAB 1: TEAM BUILDING & LEGO INTRODUCTION

Team creation, Lego Mindstorm and Cozmo Introduction

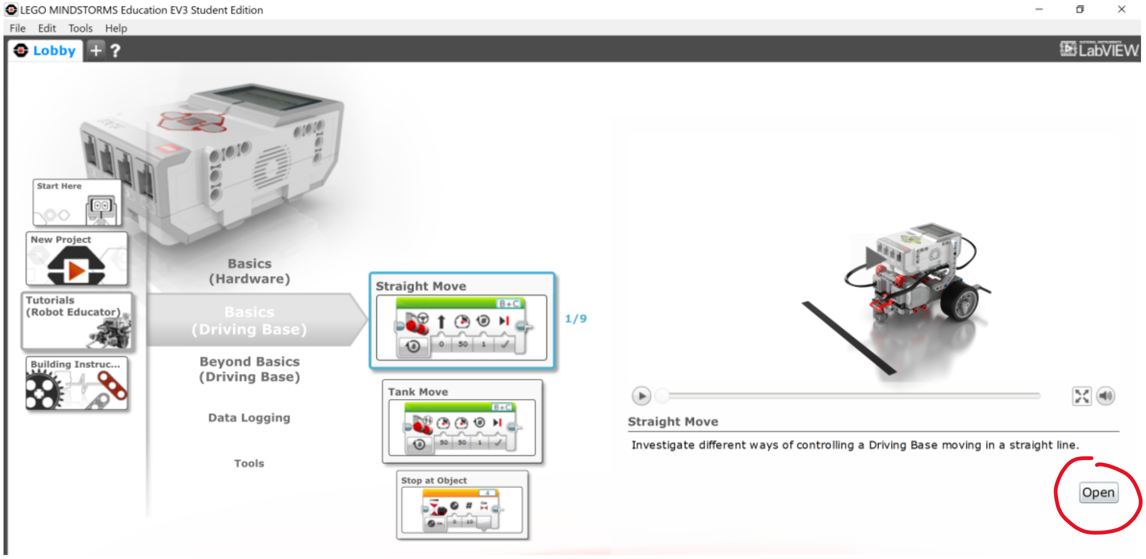
BACKGROUND: Welcome to week 1 of the course! For the remainder of the term you will complete labs and projects in teams in this lab session. Today you will form your team and get acquainted with the Lego Mindstorm and Cozmo robots.

Part A (2 points):

Create teams by XXX.

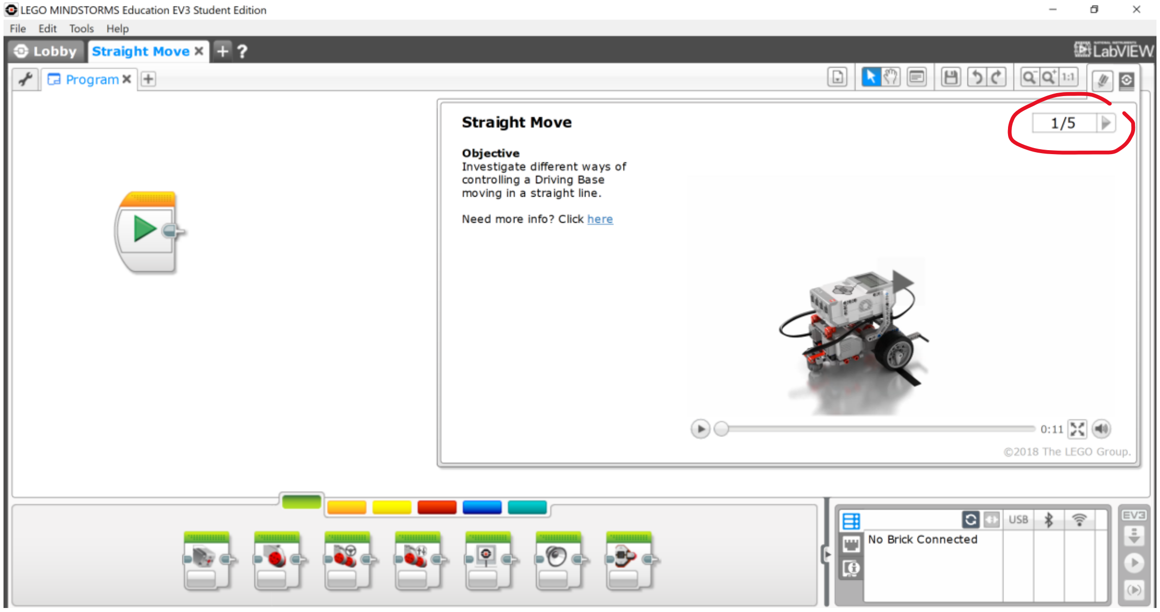
Part B (4 points): Lego Mindstorm Introduction

1. Open "LEGO MINDSTORMS Education EV3 Student Edition" software on one laptop in your team.
2. Select "Tutorials (Robot Educator)" > "Basics (Driving Base)" > "Straight Move" > "Open"



Investigate different ways of controlling a Driving Base moving in a straight line.

3. Select the right-pointing arrow to get to "step 2":



Straight Move

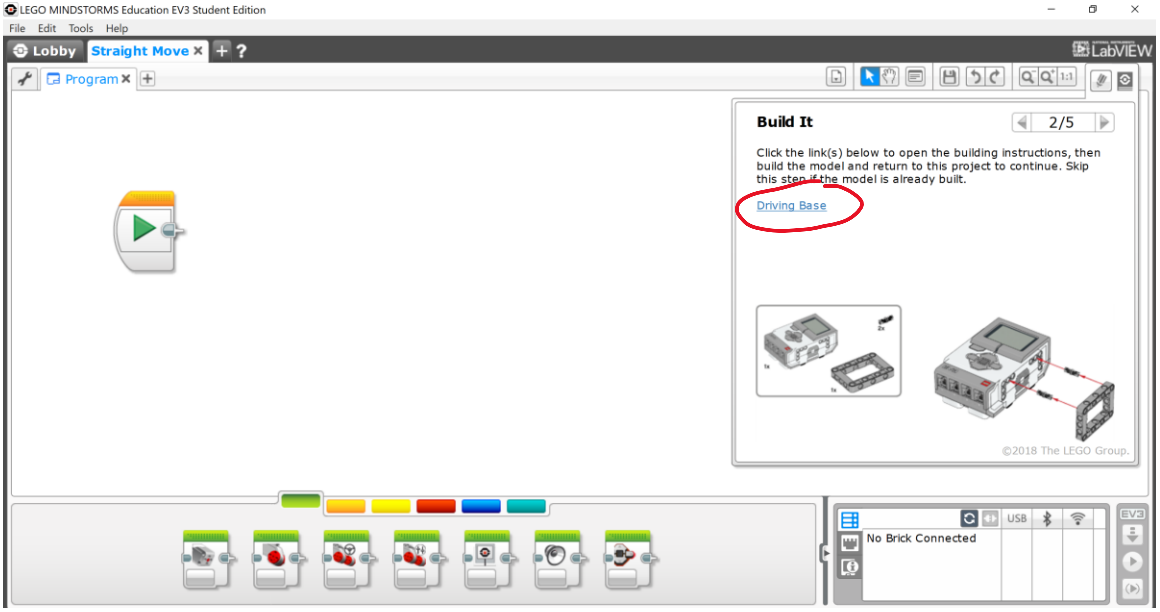
Objective
Investigate different ways of controlling a Driving Base moving in a straight line.

Need more info? Click here

©2018 The LEGO Group.

No Brick Connected

4. Select “Driving Base”:



Build It

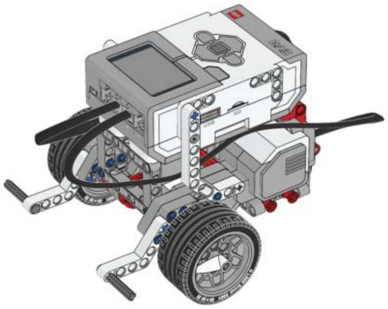
Click the link(s) below to open the building instructions, then build the model and return to this project to continue. Skip this step if the model is already built.

Driving Base

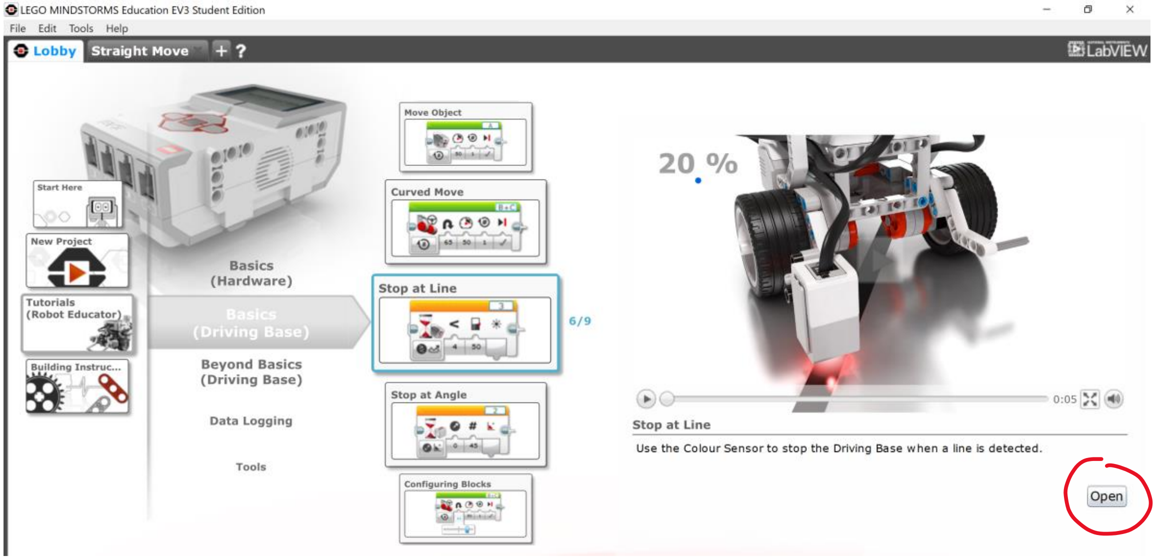
©2018 The LEGO Group.

No Brick Connected

5. Follow each step to build the driving base of the Robot:

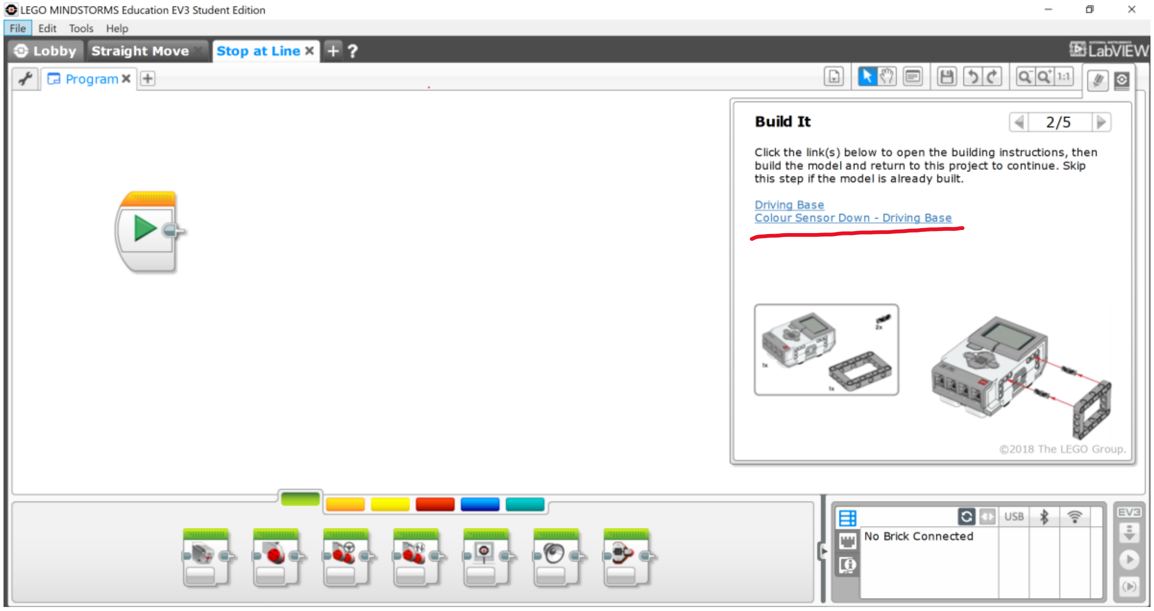


6. Once the driving base is built, connect the brick to your laptop with a USB cable. Follow the remaining 3 steps of the “Straight Move” tutorial.
7. Now that you know how to get the robot to move, let’s get it to stop! Go back to the “Lobby” tab and select the “Stop at Line” tutorial.



Use the Colour Sensor to stop the Driving Base when a line is detected.

8. Use the same driving base and now add a color sensor following the tutorial in “step 2”:



Click the link(s) below to open the building instructions, then build the model and return to this project to continue. Skip this step if the model is already built.

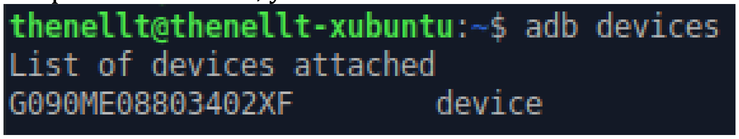
9. Once the color sensor is connected, continue the “Stop at Line” tutorial.

Part C (4 points): Cozmo Introduction

We will be using python3 and the Android Debugging Bridge (ADB) to allow us to directly control Cozmo via code. Follow the directions in the appropriate SDK Setup document for the operating system on your laptop. (Windows, macOS, and Linux Supported)

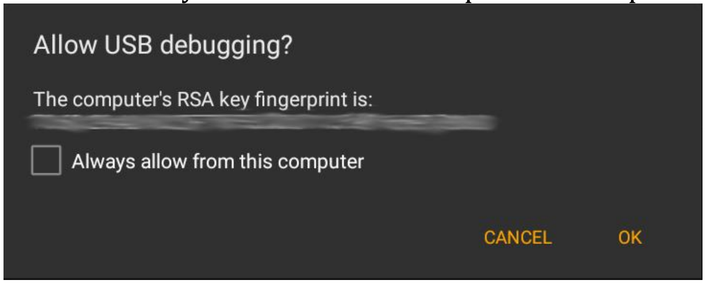
Once your computer is up and running with python and ADB, plug in the Fire tablet and verify it is recognized.

1. To do this, type: adb devices
2. If your setup was successful, you should see a result like the one below.



```
thenellt@thenellt-xubuntu:~$ adb devices
List of devices attached
G090ME08803402XF        device
```

3. If your output says "unauthorized" instead of "device", you will need to allow the connection on the Fire tablet. Shown below is an example of what the authorization message should look like. Select "Always allow from this computer" and tap "OK".



Allow USB debugging?

The computer's RSA key fingerprint is:
[redacted]

□ Always allow from this computer

4. If you are still unable to fully connect to the tablet, please have a TA check your setup before continuing.
5. Setup is complete. You are ready to move on to running your first piece of code for controlling Cozmo!

Python Hello "insert name here"

By the end of the course you will be able to fully control Cozmo via python code. For today you just need to be able to demonstrate that you have python successfully setup. We have supplied an example piece of python code to help you demonstrate your setup to the TAs.

1. Download lab1.zip from the course website lab page.
2. Extract the contents to a location you can remember. For example ~/Desktop/lab1
3. Inside the lab1 folder are several files. Open the one called "hello.py".
4. Modify the 6th line to include the names of the members of your group and save your changes.
5. In your terminal, change to your extracted folder: cd ~/Desktop/lab1
6. Run the program: python3 hello.py
7. If your setup works you should hear Cozmo introduce your group. Demonstrate this to a TA.

Structured Playtime/exploration

Below is a list of tasks that will help you get to know Cozmo and the sorts of functionality it has.

1. Using the Discover section of the Cozmo app, use the "Meet Cozmo" option to let Cozmo learn your faces and names. If asked, you should be able to demonstrate to a TA that Cozmo has been introduced to all the members of your group.

2. Also in the Discover section of the app, try out the Explorer mode. This will let you use the tablet as a remote to directly control Cozmo.
    a. Set Cozmo and the three cubes on one side of your workspace.
    b. Take turns using the tablet controller to move all the cubes to the opposite side of your workspace.
    c. Make sure Cozmo doesn't fall onto the floor!
3. Try out the Sandbox and Constructor modes. These parts of the Cozmo app will let you start programming Cozmo to complete basic tasks.
    a. Write a program in the Sandbox mode that uses at least one block from each category (Drive, Actions, Animations, Events, and Control).
    b. Once you are satisfied with your creation, move to the more powerful Constructor mode and recreate your program.
    c. Document what bloc
    d. ks they use, what did they come up with? What was their process.
4. Cozmo can tell each of the three cubes apart. Using the Constructor mode, program three unique routines for Cozmo that are specific to each of the three cubes.

Optional: Idea Swap

1. Come up with instructions for a task that takes at least 5 steps for Cozmo to complete. Must use speech or involve the cubes in some way, be creative!
2. Use the scratch interface to come up with a solution to your task just to make sure your idea is possible.
3. Trade instructions with another group.
4. Program a solution to the other groups task.
5. When both your group and the group you traded with are done implementing each other's instructions, come together and compare solutions. Did each group solve the problem in the same way? What was the different, the same?

LAB 2: RAMP DESIGN PART I

Introduction to the design process, mechanical failure modes, and 3D printing

BACKGROUND: Some buildings on OSU campus were built before the U.S mandates on accessibility for persons with impaired mobility. Can you think of some building entrances that are not universally accessible? (hint: Rogers Hall!) It wasn't until 1990 that the U.S Congress passed the Americans with Disabilities Act (ADA) which was the nation's first civil rights law addressing the needs of people with disabilities. In this lab, your team will design an ADA-complying ramp following the design engineering process. This lab is separated into two parts. In Part 1, you will assess the design requirements based on each pillar of sustainability (economic, environmental, and social) and propose a design solution for a prototype ramp. In Part 2, the MIME students will build the design and make adjustments if needed while the CS students will program the Cosmo robots to drive across the ramp. Then, all students will collect data from the Cosmo to evaluate the ramp's performance. At the end, you will be asked to reflect on the design process and performance of your final solution.

Part A (15 points):

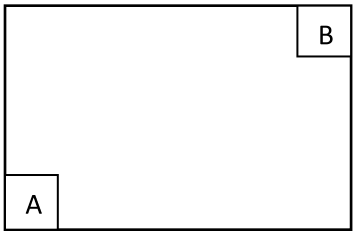
Read the design requirements and fill in the worksheet as you step through the design process.

Materials:

- Floor plan, 2' x 2' (provided)
- Cardboard ($0.50 per square inch, recyclable)
- Masking tape ($0.75 per inch, not recyclable)
- Popsicle sticks ($1 each, equivalent to 1.5 square inch, recyclable)
- Plastic cups ($2 each, equivalent to 11 square inches, not recyclable)
- 3D printed platform

Design Requirements:

- Build a prototype ramp that fits within the floor plan constraints 2 ft x 2 ft (start at point A and end at point B)

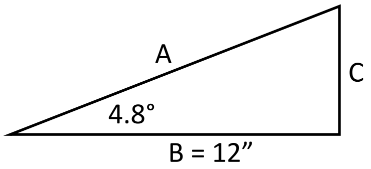


- The final landing platform (Point B) will be exactly 5 inches off the ground. A 3D printed platform will be taped on top of a plastic cup.
- The ramp must be able to support the Cosmo robot (not fail) as it drives from A to B
- The ramp must follow ADA requirements for slope (<4.8 degrees)
- Budget: $75
- At least 50% of all materials used must be recyclable (calculated in square inches)

DESIGN PROCESS:

1. *Ask & Research:* Identify the needs and constraints of you design

1a. *Technical parameters:* Fill in the missing calculations for the following:



Calculate C using trigonometry: C = ______

Calculate A using Pythagorean theorem ($A^2 = B^2 + C^2$): A = ________

How much distance on the floor plan (B) is required for the height (C) to be 5 in? __

How long will the ramp have to be? ___

How wide should the ramp be to accommodate the Cosmo robot? ___

What space is required if the Cosmo needs to turn? ___

What other technical aspects need to be investigated for this design?

1b. *Social parameters*: (parts A and B of the Design for Social Justice criteria)

i. List five stakeholders (one is given) involved in a ramp-building project like this and identify their specific needs (either research online or use you best guess/judgement):

1) ___Government___: Must follow ADA rules, comply with all U.S laws, etc.

2) ____________________:

3) ____________________:

4) ____________________:

5) ____________________:

ii. Identify three structural conditions that give rise to the need you are addressing:

1)

2)

3)

1c. *Environmental parameters*: Fill in the following table

Recyclable Materials	Non-recyclable Materials

To calculate percentage of recyclable vs non-recyclable using the ratio of total square inches. Fill in the empty boxes and complete this calculation:

Material:	**Cost**	**Number used:**	**Equivalent (in^2):**
Cardboard	\$0.50 per in^2	-	50
Masking tape	\$0.75 per in^2	-	20
Popsicle stick	\$1 per stick	5	
Plastic cup	\$2 per cup	2	

Total recyclable (in^2): _______

Total non-recyclable (in^2): _______

Percentage recyclable: _______

1d. *Economic parameters*: Fill in the empty cells in the following table to calculate the cost of a theoretical design:

Material:	**Cost**	**Number used:**	**Equivalent (in^2):**	**Total cost of material:**
Cardboard	\$0.50 per in^2	-	50	\$25
Masking tape	\$0.75 per in^2	-	20	
Popsicle stick	\$1 per stick	5		
Plastic cup	\$2 per cup	2		

Total cost of the design: ____________

2. Imagine!
   Develop possible design solutions considering all that you now know. Each team is allocated some materials to brainstorm with (have the resource manager collect from the LA). Remember, the ramp must support the Cosmo robot as it drives across. Take a look at the figure below and deliberate how your design could mitigates these common "failure modes". On this page, draw or write down AT LEAST ten possible solutions.

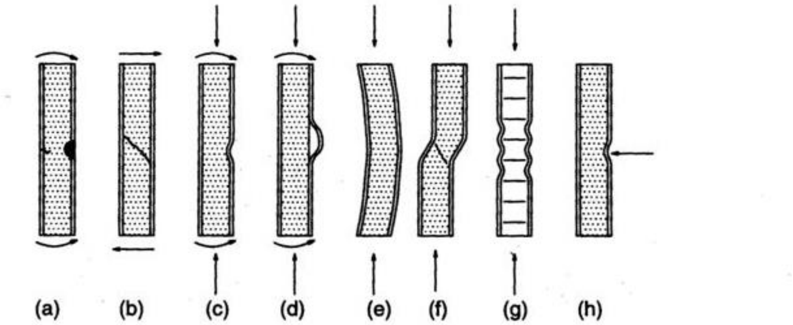


Figure 7.1 Failure modes in sandwich beams. (a) Face yielding/fracture, (b) core shear failure, (c and d) face wrinkling, (e) general buckling, (f) shear crimping, (g) face dimpling and (h) local indentation.

3. Plan: Select a promising solution that your team agrees on. Draw a sketch of it below and fill in the table.

Material:	**Cost**	**Number used:**	**Equivalent (in^2):**	**Total cost of material:**
Cardboard	\$0.50 per in^2	-		
Masking tape	\$0.75 per in^2	-		
Popsicle stick	\$1 per stick			
Plastic cup	\$2 per cup			

Total cost of the design: ____________

Total recyclable (in^2): ____________

Total non-recyclable (in^2): ____________

Percentage recyclable: ____________

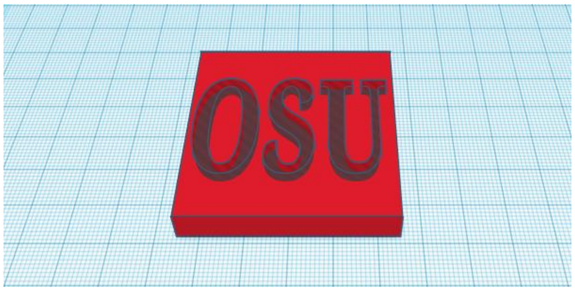
Part B (5 points):

MIME Students:

You will 3D print the final landing platform of your ramp. This piece will be 0.25 inches thick and taped on top of an upside-down plastic cup (4.75 in) so that the final platform is exactly 5 inches off the table.

Log into TinkerCAD online. Applying what you learned in the Pre-Lab assignment, build a platform with dimensions: 3.5” x 3.5” x 0.25”.

Now, get creative! Make a design (shapes or text) on the top surface of the platform that is 0.125” deep (using the “hole” feature). See below for an example:



Have your team’s final CAD drawing approved by the LAs. Export the file into an STL with the following format (Lab#_Team#_week2.STL).

CS Students:

Write a program which will take the amount of each material as input, and outputs the total cost of each material as well as the total materials cost.

To get you started, you may use this template:

```
   # Authors: <GROUP MEMBERS>
   # Assignment: Lab 2: Cost Calculation
   # Date: DD/MM/YYYY
      def main:
         cardboardCost = 0.50
         tapeCost = ________
         stickCost = ________
         cupCost = ________

cardboardAmnt = float(input(“How much cardboard (in^2) do you need? ”))

print(“The cardboard will cost : $“, cardboardCost * cardboardAmnt)

      main()
```

LAB 3: RAMP DESIGN PART II

Introduction to gathering and evaluating data.

BACKGROUND: Now that you have stepped through the design process, it is time to build your ramp design. During Part II, MIME students will build the design and make adjustments if needed while the CS students will program the Cozmo robots to drive across the ramp. Then, all students will collect data from the Cozmo to evaluate the ramp's performance. At the end, you will be asked to reflect on the design process and performance of your final solution.

Review the design requirements in Lab 2 and fill in the worksheet as you step through the activity.

DESIGN PROCESS:

1. (4pts) *Create:* Implement your design.

   MIME: Build your ramp. Follow your design as closely as possible. (If you are a MIME only team, delegate some of your members to complete each task.)

   CS: Create a program in python to drive Cozmo up the ramp. Start by using the design from Part I. You may use this template to get you started:

```
# Authors: <GROUP MEMBERS>
# Assignment: Lab 3: Ramp Building
# Date: DD/MM/YYYY

import CozmOSU

def main(robot):

    ### YOUR CODE TO DRIVE UP THE RAMP ###

robot = CozmOSU.Robot()
robot.start(main)
```

   You will need to reference the CozmOSU documentation, found here: http://classes.engr.oregonstate.edu/eecs/fall2018/cs160-030/CozmOSU/

   Once both MIME and CS groups have completed their tasks, test the cozmo program, and make any necessary modifications.

1a. *Plan Reflection (MIME):* Answer the following questions about your plan.

Does your ramp accurately reflect your design?

Did you make any modifications to your design?

Did you estimate the required materials accurately?

Did you use more or less materials than you initially planned for? What may have caused this?

1a. *Plan Reflection (CS):* Answer the following questions about your plan and program.

Did you have to modify your program to work with the ramp?

What kind of modifications needed to be made?

2. (4 pts) *Test and Evaluate:* Follow instructions and answer any questions.

Using your program to drive Cozmo up the ramp, add the following **bolded** lines.

```
# Authors: <GROUP MEMBERS>
# Assignment: Lab 3: Ramp Building
# Date: DD/MM/YYYY

import CozmOSU

def main(robot):

      # Calibrate pitch, must start on level ground
      robot.calibrateLevelPitch()

      # Name of output file, (will store locally to your
      #     computer)
      outfile = "slope-data.txt"

      # Time between each reading in seconds.
      #     Test different values here. Ex. 0.1, 0.5, …
      deltaTime = <FLOAT>

      # Start recording pitch to output file
      robot.recordPitch(outFile, deltaTime)

      ### YOUR CODE TO DRIVE UP THE RAMP ###

robot = CozmOSU.Robot()
robot.start(main)
```

Before moving forward, confirm that you have a file saved in the same directory as your program that contains a large collection of values. Do not worry if the values are not what you were expecting/hoping to see.

2a. Examine the saved file and answer the following questions.

Do you see any values that do not make sense? YES / NO

If yes, what are some? _____ , _____ , _____

What datatype should be used to store these values? Circle one

INT FLOAT STRING CHAR BOOLEAN

Are there any significant outliers? YES / NO

Give an example: ____________ Explain:

Are the values what you expected to see? YES / NO

Explain:

2b. *Analysis:*

Now create a new program using the first three values from **slope-data.txt** to answer the following questions.

Does the slope increase beyond 4.8 degrees?

What is the average slope of the ramp?

Does the slope of the ramp change?

What range do the slopes fall into?

You may use the following template to get you started. Replace <DATA_TYPE> with the datatype your selected above. (i.e, if you selected INT, use int(file.readline())

```
# Authors: <GROUP MEMBERS>
# Assignment: Lab 3: Ramp Building - Analysis
# Date: DD/MM/YYYY

def main():

      # open your file for reading
      file = open(“slope-data.txt”)

      # Get the first 3 values from the file
      val1 = <DATA_TYPE>(file.readline())
      val2 = <DATA_TYPE>(file.readline())
      val3 = <DATA_TYPE>(file.readline())

      file.close()

      # Your calculations here
      #     Use val1, val2, and val3

main()
```

3. (2 pts) *Reflection*: Answer the following questions

   What did information and skills do this lab reinforce?

   Are there any skills that you think you needed to complete this lab? Explain.

   Briefly, how would you improve your design from part I to better complete this task?

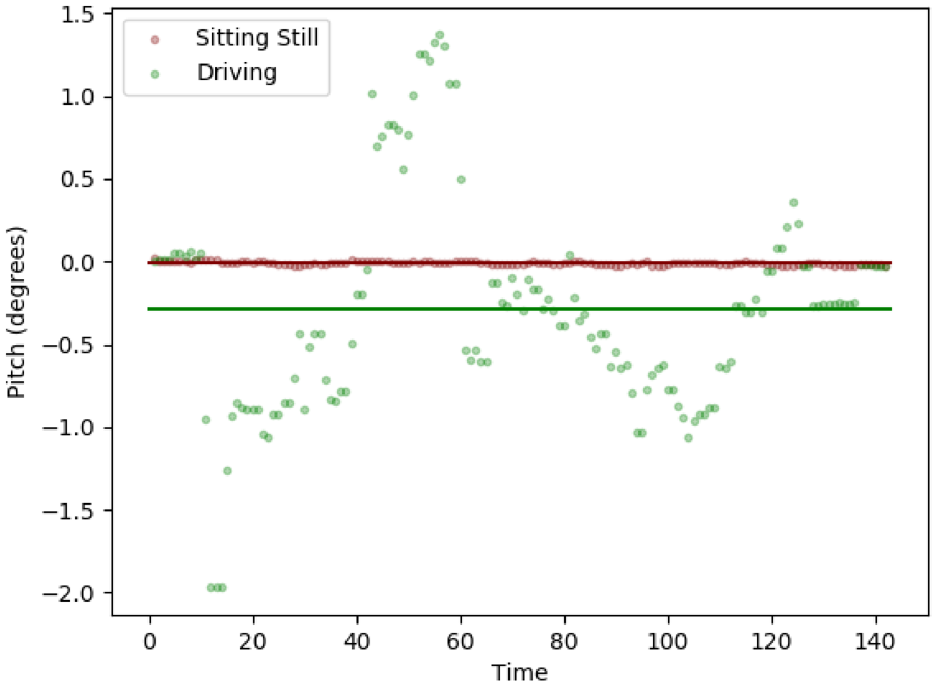
Extended Learning:

How might you improve your program to average all of the data points in your file?

Create a new program to get the average from your file. Use this template to get you started.

```
# Authors: <GROUP MEMBERS>
# Assignment: Lab 3: Ramp Building - Average
# Date: DD/MM/YYYY

def main():

      # Open your file for reading
      file = open(“slope-data.txt”)
      line = file.readline()

      # Iterate while the next line is valid
      while line:
            value = <DATA_TYPE>(line)
            print(value)

            # Keep this as the last line of the loop
            line = file.readline()

      # Close the file
      file.close()

main()
```

Consider the following graph. The data presented is the Cozmo robots pitch on a level surface in two scenarios for 7.5 seconds each. Red is when the robot is sitting still. Green is when the robot is driving. Each line represents the average respectively. Note that the target value is 0 degrees.



Based on the graph, is the sensor precise? When?

Based on the graph, is the sensor accurate? When?

Based on the graph, is using the average value of recorded angles reasonable?

What are some of the drawbacks to using an average? Consider the number of readings above, and when they are the most accurate. *Hint: time*

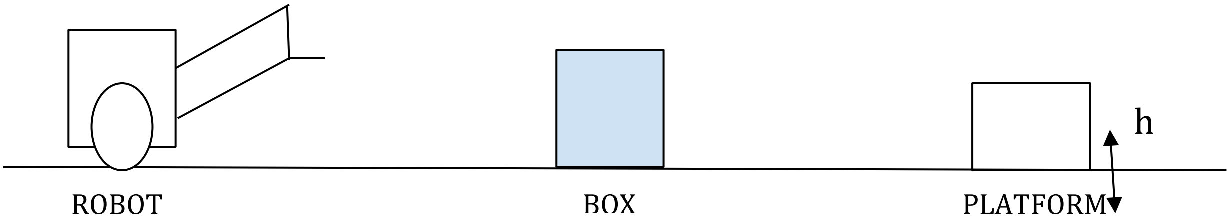
Discuss how the accuracy and precision of the Cozmo impact the results you recorded. Is the Cozmo robot a viable tool for measuring the incline of your ramp? In what cases is the Cozmo useful for measuring and incline?

LAB 4: FORKLIFT PART I

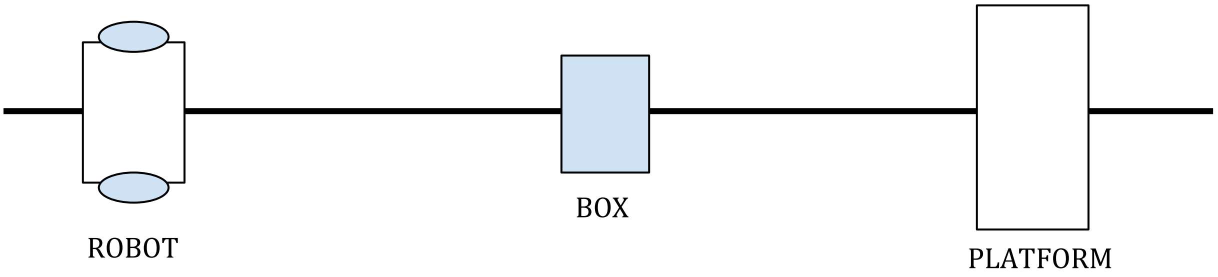
Introduction to using sensors in motion planning for robot-object interaction

BACKGOUND: Amazon Inc. has hired your company to build a robot that will autonomously organize boxes in their warehouses. Your goal is to design a robot that uses a forklift to pick up and transport objects from one location to another. You will build a moving base for the robot, use ultrasonic sensor to locate an object, use a gyroscope to maintain movement in a straight line, and learn how to use CozmOSU functions to program the Cozmo robot. Each team will be given a Lego Mindstorm Education Ev3 set, a Cozmo, platform, and masking tape (to mark the center line). Document your design both as a model on the Lego Digital Designer and in your notebook for the LA's to check at the end of lab.

SIDE



TOP VIEW



Box dimensions: 1.5 x 1.5 x 1.5 in
Platform dimensions: 1.75 x 2.25 x 3.0 in

Conditions and Constraints

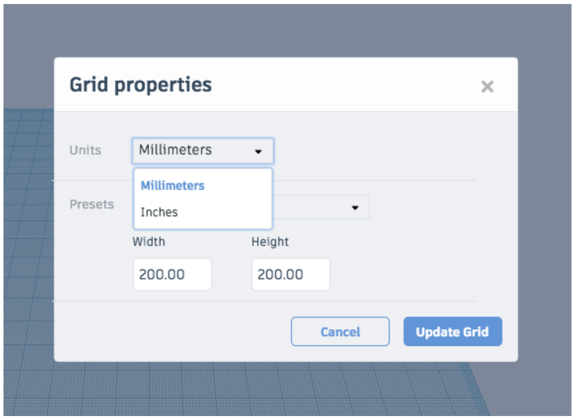
- You are allowed to modify the structure of the box (e.g adding material to it) to make it compatible with your forklift design.
- The dimensions of the box must be intact.
- You cannot take material away from the box (e.g cutting grooves or holes into it)

Part A (3 points):

1. Build a driving base for the robot by following the tutorial under 'Building Instructions' module in the **LEGO MINDSTORMS Education EV3 software**.
2. Brainstorm solutions to lift the box. Draw sketches of your solution with dimensions.
3. Use TinkerCAD to make 3D model of the box (with changes to fit your solution). It will be 3D printed and available in the next lab for you to use.
    a. Before starting to build the model, make sure your grid units are in inches.

- Go to > **Edit Grid** on the bottom right-hand corner of the screen.
- Change the units to **Inches**
- Click **Update Grid**

Tip: Turn off Snap grid for finer control and manipulation of model.



Grid properties

Units: Millimeters

Width: 200.00

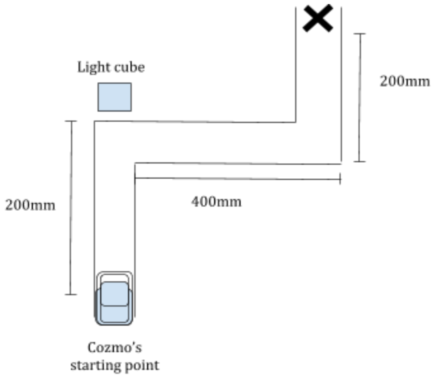
Height: 200.00

Part B (3 points):

4. Attach the ultrasonic sensor and gyro sensor to the moving base
    a. Use the ultrasonic sensor to stop at a right distance from the box and shelf
    b. Use the gyro to make sure the robot travels in a straight line
5. Design the algorithm needed to make the robot move in a straight line and stop when it detects an object.
    a. Break down the task into individual steps
    b. Think how the information collected by the sensors can help the robot decide what to do next
6. Write a program following the algorithm
7. Test and debug your code until your robot performs the tasks successfully.

Part C (4 points):

8. The next task involves using Cozmo to navigate a track and transport one of its cube. In this lab we'll focus on making Cozmo pick up the cube, navigate the track and stop at X.



Algorithm specifics:

1. Cozmo will see the Lightcube

a. Cozmo needs to do this first in order to calculate the best path to approach the cube in order to dock with it.

2. Lift the Lightcube
3. Turn 90 degrees to the right
4. Drive 400 mm forward
5. Turn 90 degrees to the left
6. Drive 200 mm forward
7. Set down Lightcube
8. Move backwards to undock with cube

Use the template below to get you started. Once you have this program, test it to verify that the Cozmo will identify and pick up the cube. *Tip: If no cube is identified, increase the timeout value.*

```
# Authors: <GROUP MEMBERS>
# Assignment: Lab 4-5: Cozmo Task
# Date: DD/MM/YYYY

import CozmOSU

def main(robot):

      # Make Cozmo look straight
      robot.moveHead(0)

      # Get the visible cube, search for 1 second
      timeout = 1
      cube = robot.getVisibleCube(timeout)

      # Go pickup the cube if a cube was found
      if cube is not None:
            robot.pickupCube(cube)
      else:
            print(“Could not find Cube”)

robot = CozmOSU.Robot()
robot.start(main)
```

Once your algorithm is working, you will need to implement the algorithm to drive the path as specified above. To do this, you will need to make use of the following 2 functions.

`robot.driveForward(distance)`	This function instructs Cozmo to drive in a straight line. Distance is an integer that specifies how far to drive in **mm**. To move forward, use a **positive** number. To move backward, use a **negative** number.
`robot.turn(degrees)`	This function instructs Cozmo to turn in place. Degrees is an integer that specifies how many degrees to turn. To turn left, provide a **positive** number. To turn right, provide a **negative** number.

For more information about these functions, visit the documentation:
http://classes.engr.oregonstate.edu/eecs/fall2018/cs160-030/CozmOSU/driving.html

Test your program and verify that it completes the task effectively

Evaluation Criteria

1. Complete 3D model of Box: **3 points *(and submit as .STL file on CANVAS before the end of the lab)***
2. Robot can detect obstacle (box) and stop: **1 points**
3. Robot can travel in a straight line: **2 points**
4. Cozmo successfully picks up cube: **2 points**
5. Cozmo successfully navigates track: **2 points**

Things to Consider

Sensors readings will always be relative to the initial position of the robot. So when using the gyro, make sure to reset it by adding the Reset Gyro block at the beginning of the program so that the initial position is always at 0 degrees.

Extended Learning : EV3 Python

After viewing the ev3dev Getting Started and ev3dev Motor documents in canvas, design and implement a program to drive 150 mm. Remember, the only function to drive is ***motor.run_timed(…)***.

Hint: circumference of your wheel is the distance it will travel in one rotation.

LAB 5: FORKLIFT PART 2

Introduction to bar-linkage mechanism and its implementation in robot-object interaction

Background

Amazon Inc. has hired your company to build a robot that will autonomously organize boxes in their warehouses.

Goal

This lab will continue focusing on designing a robot that uses a forklift to transport objects from one point in space to another.

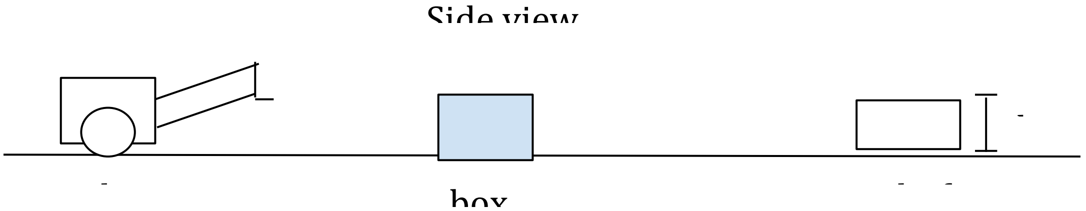
Learning objectives

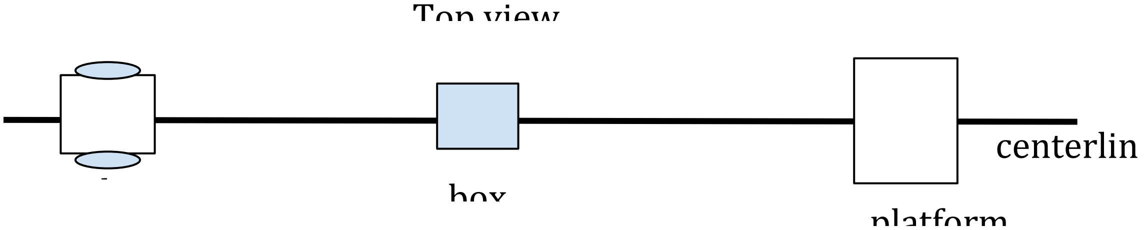
- Use Ultrasonic sensor to sense and locate object in the path.
- Use a gyroscope to maintain movement in a straight line.
- Build a forklift using the principles of a simple four bar linkage mechanism to convert rotary motion to linear motion in order to lift and transport object.
- Learn to use CozmOSU functions to program Cozmo search for Lightcube & The robot must locate the box in its path, pick it up and put it back on its designated platform. To remain within the scope of this lab, the robot, the box and the platform would be arranged in a straight lineoperate its lift.

Note: While two important goals of these labs are to give students exposure to working in teams and multiple disciplines for a well-rounded education, you are held responsible for completely understanding lab content specific to your course. For that reason, there is a hard copy of the lab available during lab after hours, and we will post an electronic copy of the lab in Canvas by Friday afternoon at 4pm. Even though you cannot get additional points for work completed outside of lab, you are encouraged to work on unfinished labs to ensure you understand the material you did not complete, as well as review sections of the lab that your teammates completed to make sure you understand all relevant material.

Task

. (*see figure below*)





Box dimensions: **L-1.5 x W-1.5 x H-1.5 in**
Platform dimensions: **L-3.0 x W-2.25 x H-1.75 in**

Materials

- 1 Lego Mindstorm Education EV3 Set
- Platform
- Masking Tape (to mark centerline)
- Box (3D printed with solution)
- 1 Cozmo
- 1 LightCube

Conditions and Constraints

- You **must** pick up the cube using your forklift by no other means **but** the 3D printed Box provided.

Part A (4 points):

- Build the two arms of the forklift using the principles of the four bar mechanism discussed in **Pre-Lab 5**.
- Look at the section **Introduction to Gears** in Pre-Lab 5 for reference. Design the appropriate motor and gear arrangement necessary to move the forklift. Look at the forklift displayed at LA desk for an example.

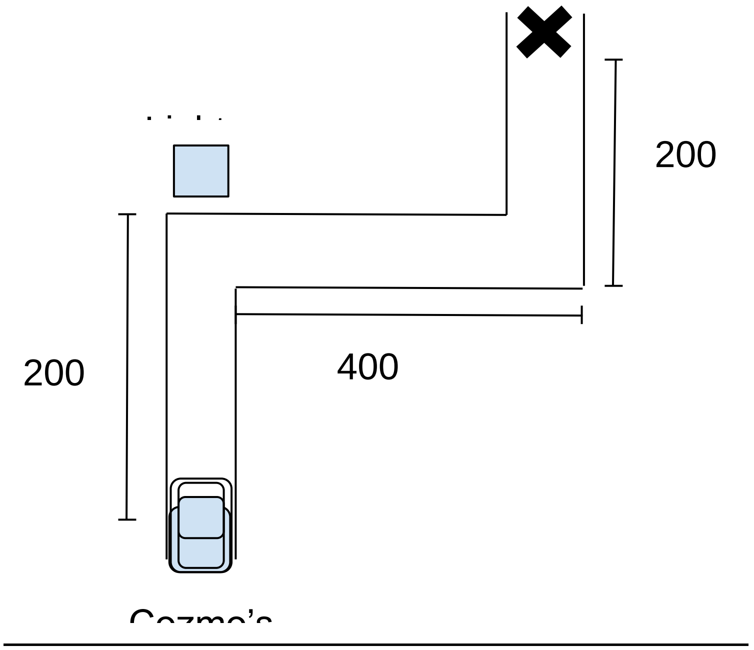
  Draw a sketch of the gear arrangement (and the shaft/shafts interacting with the four-bars).

Part B (3 points):

- Modify your algorithm so that the robot moves in a straight line, detects box, picks it up, transports it and stacks it on the platform.
    - Break down the task into individuals steps
    - Think how the information collected by the sensors can help the robot decide what to do next
    - Review the **Lego Mindstorms** section in the Pre-lab to know how to start thinking about it.
- Write a program following the algorithm
- Test and debug your code until your robot performs the task successfully.
    - Use the centerline to position box, platform and robot in a straight line.

<u>Part C (3 points):</u>

- Return to the Cozmo task from the previous lab. You are going to modify your code so that Cozmo
    - looks around and searches for the cube
    - sets down the Lightcube and undocks at the end of the track



- Consider the case where the cube is not in Cozmo's line of sight. Think of how you might scan the surrounding area.

- One solution is to rotate in place and check if the cube is in front of Cozmo.

    - The following code will scan 180 degrees around Cozmo.

```
cube = robot.getVisibleCube()
```

```
    if cube is None:
         robot.turn(30)
         cube = robot.getVisibleCube()

    if cube is None:
         robot.turn(30)
         cube = robot.getVisibleCube()

    if cube is None:
         robot.turn(30)
         cube = robot.getVisibleCube()

    if cube is None:
         robot.turn(30)
         cube = robot.getVisibleCube()

    if cube is None:
         robot.turn(30)
         cube = robot.getVisibleCube()
```

- How might you modify this so that it uses iteration? *Hint: while loop*.
    - Implement your solution so that Cozmo will continue rotating until a cube is found.
    - Test your solution.

      While testing, it may be helpful to comment out everything below
      `robot.pickupCube(cube)`

      Make sure to run tests starting Cozmo at different orientations.

      Finally, uncomment the path navigation and verify that your program works.

- Now, you will need to implement the undocking process.

  To lower the lift, you can use the function:

  ```
  robot.moveLift(0)
  ```

  After you lower the lift, instruct cozmo to drive backwards using the ***moveForward*** function. Any distance greater than 25mm is sufficient.

- **Test your program and verify that it completes the task effectively**

  For more information about these functions, visit the documentation.
  http://classes.engr.oregonstate.edu/eecs/fall2018/cs160-030/CozmOSU/driving.html

Evaluation Criteria

Lego Mindstorm Grading Rubric

Criteria	**Description**	**Points**
Performance	Lego design is proven to be capable of lifting box and placing on platform	2.5
Consistency	Forklift can pick up and place the box consistently .	2.5

Cozmo Grading Rubric

Criteria	**Description**	**Points**
Performance	Cozmo searches and locates the light cube. Cozmo is able to pick up and set down light cube.	2.5
Consistency	Cozmo can consistently complete the task.	2.5

Things to submit at the end of the recitation:

- **Lab Handout** to the LAs
- **Python code** (Submit this on Canvas). Use the naming convention: **"Lab5_SXX_TXX_.py"**
- **Video files:** Please submit the videos related to Lab #5 to the following link and be sure to use the naming convention for your file: "**Lab5_SXX_TXX_.vid**". (or other common video format extension)

  https://www.dropbox.com/request/9G92drq6rVj1OJqOxNan

MIME101/CS160 Fall 2018 Team No__________ Section Day & Time ______________________

Team Member Initials: ________________________________;_______________________________________

________________________________;_______________________________________

LAB 6: GRABBER PART I

Introduction to pathfinding algorithms and mechanical design

Background

Your neighborhood park has a serious littering problem. You have proposed to the city council that you can remedy the situation by building a robot that'll pick up trash on the ground.

Goal

This lab will focus on part of a pathfinding algorithm for detecting objects.

Learning objectives

- Understand path finding algorithm
- Create and manipulate variables and arrays
- Create and use functions (or MyBlocks)

Note: While two important goals of these labs are to give students exposure to working in teams and multiple disciplines for a well-rounded education, you are held responsible for completely understanding lab content specific to your course. For that reason, there is a hard copy of the lab available during lab after hours, and we will post an electronic copy of the lab in Canvas by Friday afternoon at 4pm. Even though you cannot get additional points for work completed outside of lab, you are encouraged to work on unfinished labs to ensure you understand the material you did not complete, as well as review sections of the lab that your teammates completed to make sure you understand all relevant material.

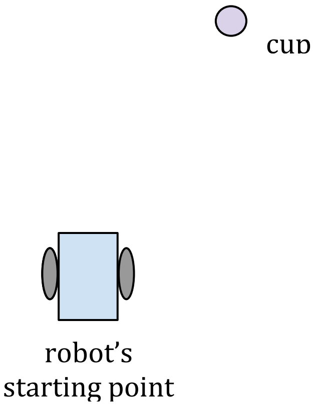
Things to submit at the end of the recitation:

- **Lab Handout** to the LAs
- **Lego Mindstorms EV3 code** (Submit this on Canvas) . Use the naming convention **"Lab6_SXX_TXX_.ev3"**
- **Video files:** Please submit the videos related to Lab #6 to the following link and be sure to use the naming convention for your file: "**Lab6_SXX_TXX_.vid**". (or other common video format extension)

  https://www.dropbox.com/request/HNfgeQbxNVAff1AmyouW

Task

The robot must search and locate a cup in the undefined test field and pick it up. (*see figure below*)



Materials

- 1 Lego Mindstorm Education EV3 Set
- Paper Cup

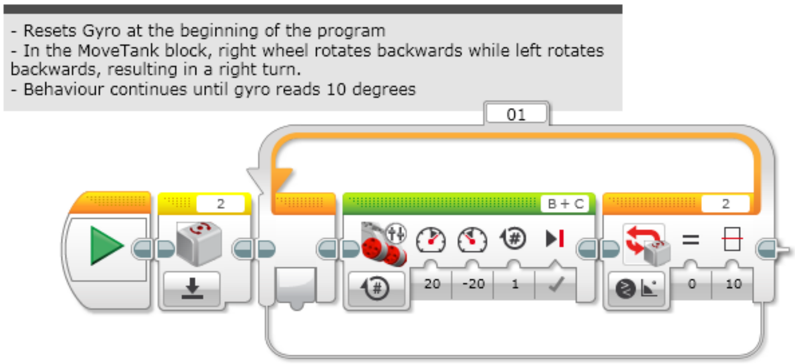
Conditions and Constraints

- The cup will be placed randomly at an unknown location
  - The figure only shows one possible scenario. The cup can be placed behind the robot.
  - The orientation of the cup will be upright and will not change for simplicity.
- The robot **must** perform a search to locate the cup and align itself to it.
- You cannot hardcode the distance of the object
- For simplicity, the cup will be the only object in its field of detection.

Part A (3 points):

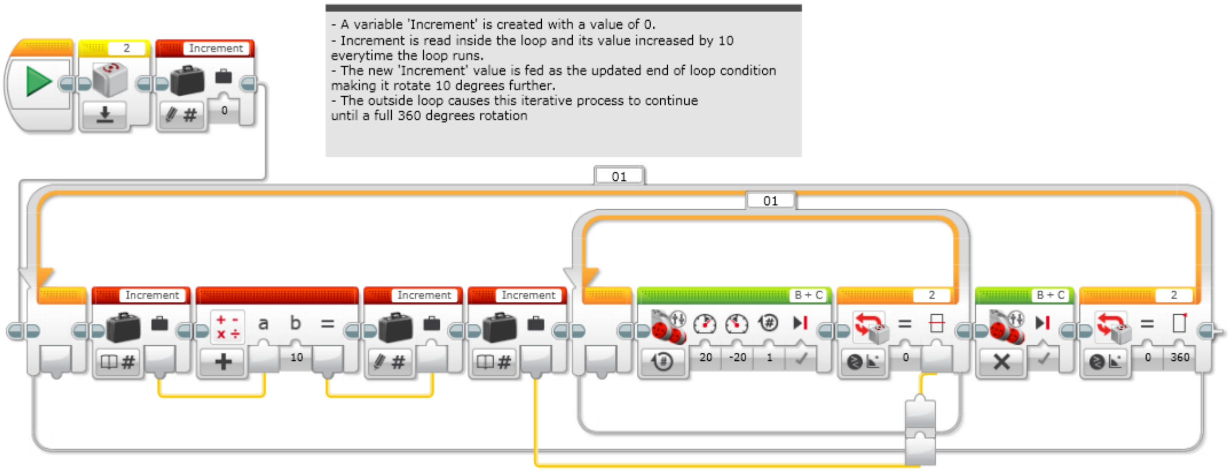
- You'll be using the gyro sensor and the ultrasonic sensor to collect data. Decide the optimal position for the sensors and attach them to your robot.
  - Since the only object in the robot's test field is the cup, we'll use the ultrasonic sensor to detect it. The smallest distance recorded is the first parameter to define cup's location.
  - To determine the direction it should be headed, the robot needs to scan all of its surroundings. This can be done by rotating in place at the starting position. The angle at which the cup is detected is the second parameter.
  - This defines the position of the cup relative to the robot in it's 2D test field space.
- You'll use the following algorithm for object detection

  - ○ Rotate 360 degrees in place and record distance at 10 degrees intervals.
  - ○ Align itself with the angle corresponding to the minimum recorded distance.
- ● Start with writing a simple program to make your robot rotate 10 degrees to the right.



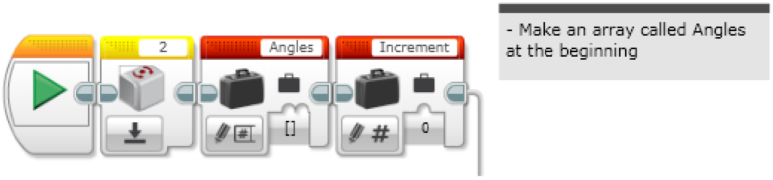
- Resets Gyro at the beginning of the program
- In the MoveTank block, right wheel rotates backwards while left rotates backwards, resulting in a right turn.
- Behaviour continues until gyro reads 10 degrees

- ● Change the program so that the initial program is iterated until the robot has rotated 360 degrees.



- A variable 'Increment' is created with a value of 0.
- Increment is read inside the loop and its value increased by 10 everytime the loop runs.
- The new 'Increment' value is fed as the updated end of loop condition making it rotate 10 degrees further.
- The outside loop causes this iterative process to continue until a full 360 degrees rotation

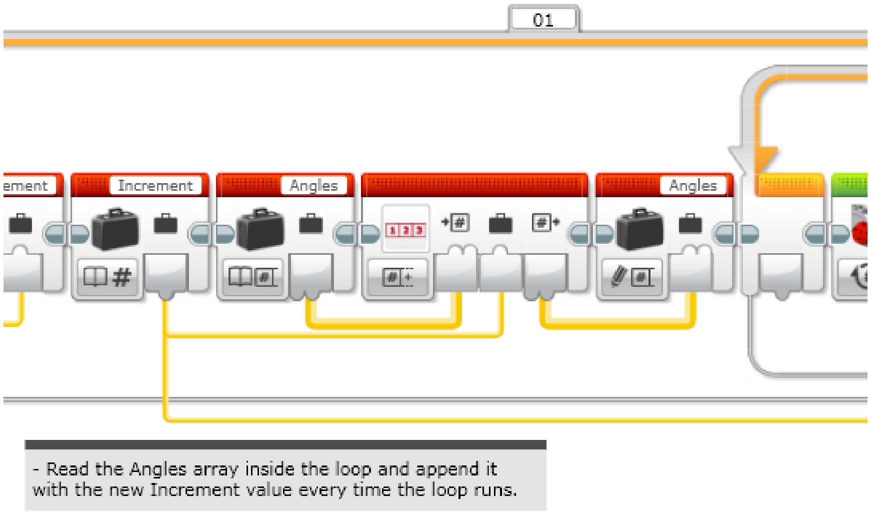
- ● Save the angles in an array



- Make an array called Angles at the beginning

Team Member Initials: _________________________________;_______________________________________

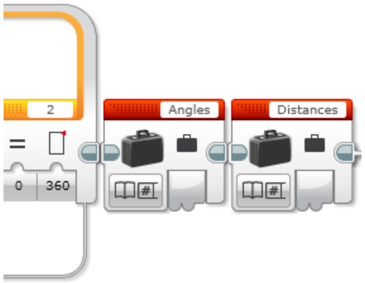
_________________________________;_______________________________________



- Read the Angles array inside the loop and append it with the new Increment value every time the loop runs.

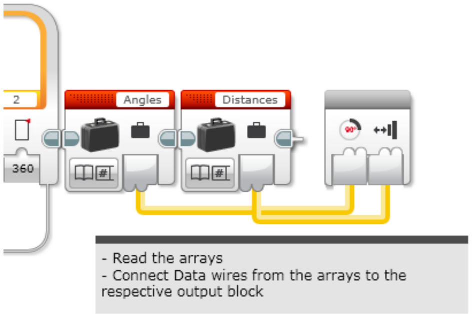
- Create an array called **"Distances"** and save readings from the ultrasonic sensor for every angular increment.

  <u>Tip:</u> Make sure that **Angles** *and* **Distances** *have the same length. This can be done by making the first index value 0 when they are created (at the beginning) and appending array later on. This will take care of repeated readings at 0 and 360 degrees.*
- We are going to turn this program into a block (or function) called **"DetectObject".** Blocks(or functions) make your code more manageable by compressing large pieces of code and making it reusable over several programs.
- Since this program performs a specific behavior it doesn't require inputs from the program but outputs two arrays, **"Angles"** and **"Distance"**.
  - To access these output arrays, read them at the end of the program.



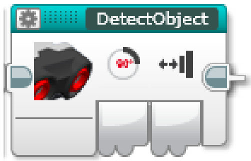
  - Name the block **"Detect_object".** Add parameters by clicking on the plus sign. The block outputs two arrays, **Angles** and **Distances**. Choose the parameter type which is ***Output*** and the data type is ***Numeric Array*** for both. Click ***Finish*** to create Block.

- ○ Connect data wires from the arrays to the output block.



- Read the arrays
- Connect Data wires from the arrays to the respective output block

- **Detect_object Block:**



Part B (3 points):

- Get checked off individually for the **Min_Block** you created in **Pre-Lab 6** by a LA.
  - 1.5 pts for writing an algorithm
  - 1.5 points for practice implementing variables, lists, and functions**.**

Part C (3 points):

- Use **Min_Block** to get the minimum distance recorded by the Ultrasonic sensor.
- Since we need to know angle at which the smallest angle was detected, we need to write another block that extracts the index of the minimum distance. The corresponding angle will have the same index as the minimum distance. Call this block **"get_Index"**.
  - The algorithm for this program is similar to **Min_Block.**
  - The block should take the ***value*** you want to find the index of, and the ***array*** it is in as inputs.
  - Think how your program might save the index at which the element in the array equals the value you entered.
- Combine all of the blocks that you just created in a new program.
- Think about how you can make your robot rotate and align to the angle where minimum distance was detected.
- Test your code to see if your robot detects and aligns itself with the object.

Evaluation Criteria

- Justification for optimal sensor position **(1 point)**
- Successfully created **Detect_object** block **(2 points)**
- Pre-lab **Min_Block** algorithm and implementation successfully outputs the minimum value **(3 points)**
- **get_Index** block successfully outputs the index of any value in an array **(3 points)**
- Successfully combines all the blocks to create a program that makes the robot detect and align with the object. **(1 points)**

MIME101/CS160 Fall 2018 Team No__________ Section Day & Time ________________________

Team Member Initials: __________________________________;______________________________________

__________________________________;______________________________________

LAB 7: GRABBER PART II

Continuation of pathfinding algorithms and mechanical component design

Background

Your neighborhood park has a serious littering problem. You have proposed to the city council that you can remedy the situation by building a robot that'll pick up trash on the ground.

Goal

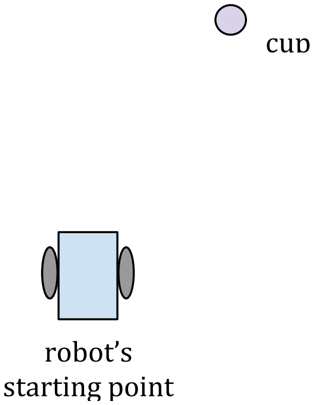
This lab is going to be a continuation of the previous lab. It'll focus on implementing the pathfinding algorithm and incorporating a claw to your robot so that it can pick up the object.

Learning objectives

- Use mechanical design principles to design a claw
- Use 1D arrays and functions (or MyBlocks)
- Implement path finding algorithm
- Execute motion-planning to grab object

Task

The robot must search and locate a cup in the undefined test field and pick it up. (*see figure below*)



Materials

- 1 Lego Mindstorm Education EV3 Set
- Paper Cup

<u>Conditions and Constraints</u>

- The cup will be placed randomly at an unknown location
    - The figure only shows one possible scenario. The cup can be placed behind the robot.
- The robot <u>must</u> perform a search to locate the cup and align itself to it.
- You cannot hardcode the distance of the object
- For simplicity, the cup will be the only object in its field of detection.

<u>Part A:</u>

- Troubleshoot your code from Lab 6 to make sure it does what it is supposed to do. The pathfinding code has to work before it can go and pick up the object.

- If you haven't completed the code from Lab 6, please make sure to do so.

<u>Part B (5 points):</u>

- Re-build the driving base you had from the previous lab including the sensors.
- Use the concepts presented in the **Pre-lab 7 Claw Design Principles section** to build a claw that can be operated to grab and pick up object.
    - First decide how you'd like to mount the motor onto your base. This will affect the positioning of gears.
    - Look at the **Pre- lab 5 Introduction to Gears section** file to figure out the best gear arrangement for your claw.
    - **<u>For fun</u>**, you might consider building a claw that grabs **<u>AND</u>** lifts. Realize that this would require motion in two perpendicular directions.
      Pro Tip: It is possible to use one motor and build a claw that grabs and lifts object. Imagine what gear arrangement can accomplish this task. *<u>This is not required as part of the grading rubric.</u>*

<u>Part C (5 points):</u>

- The grabbing portion of the task will require some motion-planning where you program the robot to perform certain steps to accomplish a task *(similar to the forklift lab).*
- After building the claw, open a new program.
    - Write a program to play around with the motor to see how far it has to rotate to open and close the claw.

    - Write down the algorithm to keep track of the necessary steps.
- Figure out the algorithm necessary for the robot to move towards the object, grab and pick it up and write a program.
- Incorporate your path-finding code from the previous lab to your current program.
    - **Things to Consider**
        - Think how you may have to position your robot so that the object is within it's best grasping range.
        - The speed of pinching action will depend on the weight of the object. If it grasps too fast, it might push the object out of the grasp.

- Test and debug your code until your robot performs the task successfully.

Things to submit at the end of the recitation:

- **Lab Handout** to the LAs
- **Lego Mindstorms EV3 code** (Submit this on Canvas) . Use the naming convention **"Lab7_SXX_TXX_.ev3"**
- **Video files:** Please submit the videos related to Lab #7 to the following link and be sure to use the naming convention for your file: "**Lab7_SXX_TXX_.vid**". (or other common video format extension)

  https://www.dropbox.com/request/kB8kZqwK1qXbx6S9JnKw

Evaluation Criteria:

- Claw design: **(5 points)**

Criteria	**Description**	**Points**
Performance	The design can be easily operated and programmed to successfully grab object.	3
Reliability	The grasp is successful at least ⅔ times	2

- Robot successfully implements path-finding algorithm to detect and move towards object. **(2 points)**
- Robot successfully implements motion-planning algorithm to stop and grab object. **(3 points)**

LAB 8: PILL HOPPER DESIGN

Development of first draft of 3D printed pill hopper design for final project

BACKGROUND: In order to produce the fully-autonomous pill sorting system outlined in the MIME101/CS160 final project document, teams must design, print, and mount pill hoppers to final designs. The purpose of this laboratory is to help students create their first draft of a functional final pill hopper design.

Part A (1 point):

First, as a team, review the final pill sorter document in Canvas as well as the hopper grading criteria listed on the final page of this document. Following the review, as a team, create a pill hopper design requirements table based on final project requirements. Include **at least** three distinct **quantifiable** design criteria.

Table 1. Hopper Design Requirements

Design Criterion	**Criterion Description**	**Criterion Importance**

HOPPER DESIGN BACKGROUND:

Hoppers have been used to store and sort materials as far back as humans have harvested and stored crops. Prior to the 1960s, hoppers and storage vessels were designed mostly by guessing [1]. However, Andrew W. Jenike developed material flow theories and methods to apply hopper design theories.

Hoppers may be designed in a variety of unique shapes. The figures below shows some of the most common shapes for mass flow hoppers and funnel flow hoppers. The primary difference between mass flow and funnel flow is that in the case of **mass flow** all of the material in the bin is in motion. Whereas, in the case of **funnel flow**, only the material in the center above the hopper outlet is in motion.

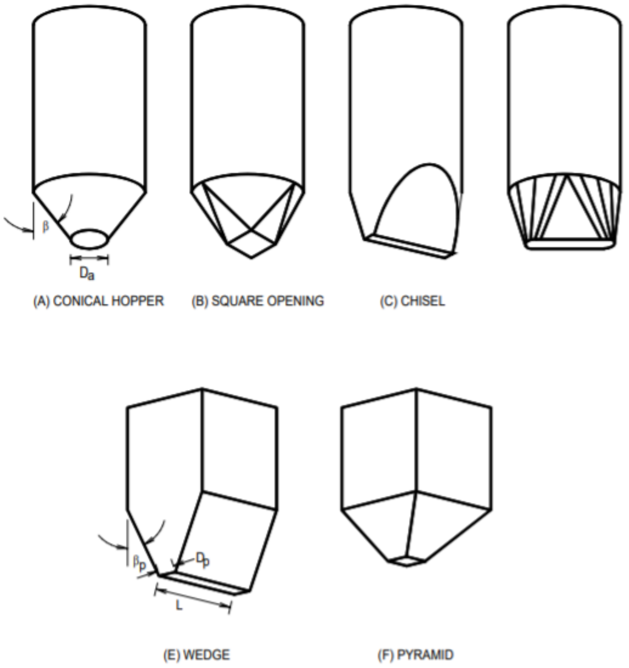


Figure 1. Common Mass Flow Hopper Concepts

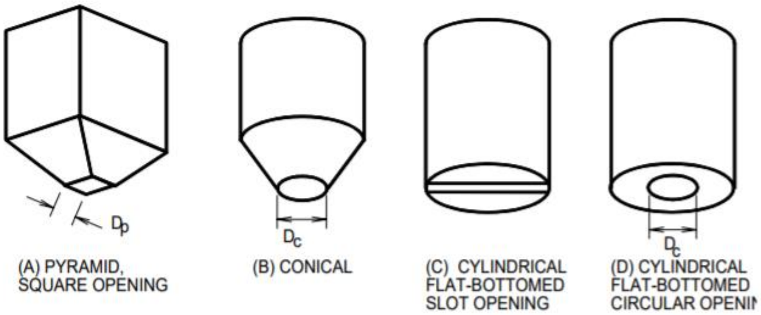

Dp
Dc
Dc
(A) PYRAMID, SQUARE OPENING
(B) CONICAL
(C) CYLINDRICAL FLAT-BOTTOMED SLOT OPENING
(D) CYLINDRICAL FLAT-BOTTOMED CIRCULAR OPENIN

Figure 2. Common Funnel Flow Hopper Concepts

The most prevalent hopper design problems cause inadequate flow through the hopper outlet. Listed below are the most noteworthy problems which should be addressed by design teams:

1.) *Bridging:* material is cohesive enough that particles are able to form arch bridges across the hopper span. This error can cause flow to cease completely.

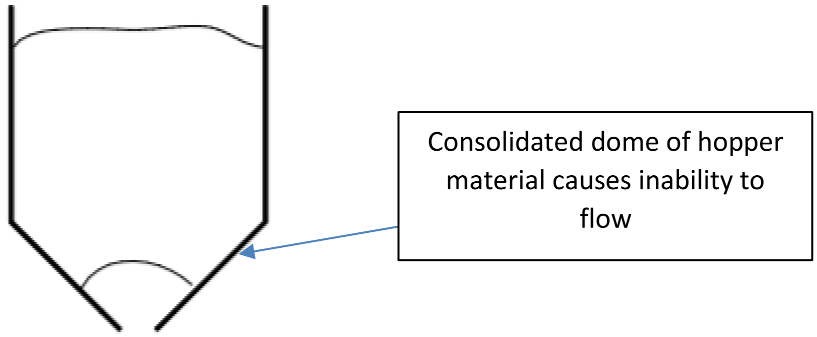
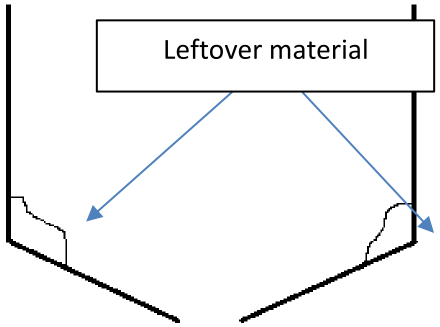

Consolidated dome of hopper material causes inability to flow

Figure 3. Material Bridging

2.) *Incomplete Emptying:* Dead spaces within the hopper bin can prevent a complete discharge of material.



3.) *Time Consolidation:* When particles are allowed to sit in a hopper over a long period of time, they are allowed to rearrange themselves to become more packed together. The consolidated materials tend to cause bridging.

Part B (3 points): Hopper Design Concept Sketches

Each member of the team is required to come up with at least three distinct hopper design concepts, and provide sketches in the space provided below. For now, keep the sketches fairly general, and not quantified with dimensions. Within the sketches, indicate where pills would be stored, and how the pills would ideally flow. Additionally, give a general description or sketch of how the hopper may be mount to the parent assembly (the final pill sorter). Lastly, provide a description of how the hopper, when integrated into the final design, will successfully dispense one pill at a time.

1.)

2.)

3.)

Now, come together as a team and present and discuss the hopper design concepts. As a team, come up with the **one** hopper design that is most widely recognized as simple and satisfactory with regard to the design criteria. Additionally, if there is a clear runner up, note the second place design because the hopper chosen is **not** necessarily the final hopper design to be incorporated to the final assembly as there will be an opportunity to redesign the parts if they do not work.

Here are some key guiding questions to consider when discussing the design concepts. It is not necessary to write down answers to these questions. Note, these are not in any order of importance:

1.) Is the design concept able to be created in Tinkercad with relative ease?
2.) Can the design hold 20 pills, as required by the hopper grading criteria?
3.) How will the hopper attach to the parent assembly of the final design? Will it require many more Lego elements?
4.) What does the pill dispensing mechanism look like? Does the hopper allow for simple single pill dispensing process?
5.) Will the pill color be read while in the hopper or following the dispense step?
6.) How much material is required to build the hopper? Consider doing a quick and rough volume calculation, as design economics are a significant portion of the grading criteria.

Part C (5 points): Detailed Design Sketch

Once the team decides on a design concept, individually sketch the design below. In this section, there must be inclusion of how the design will be mounted to the final parent assembly; this can be a brief written discussion to supplement a sketch (sketch is required). Additionally, this sketch must include how the pills will flow, and must include dimensions and calculations that prove the design will allow (1) pills to fit through the outlet, and (2) 20 pills to be stored.

Now, as a team, open Tinkercad and design your part!

Once finished, export as an .STL file, and upload to canvas; one design should be submitted per team. No more than one design may be submitted per team.

Part D (1 Point): Post Laboratory Questions and Discussion:

Answer the following question clearly and concisely. You are encouraged to use sketches to supplement your descriptions.

1.) How does your hopper design address the most prevalent hopper design failure modes discussed in the *Hopper Design Background* section of the laboratory?

Each student submit a complete laboratory document to class LAs

INITIALS__________________

FINAL DESIGN WORK PERIOD

This laboratory session was designed to give design teams time to work on pressing areas of their final design projects. Additionally, this laboratory session serves as a time for students to ask specific questions to LA staff. The first portion of this laboratory will involve setting action items for the work session and discussing with LA staff; this portion should take no more than 30 minutes of the section. The remainder of the laboratory will be open work time where teams work on determined action items.

Part A (4 Points): Gantt Chart Review

As referenced in the pre-lab document, at least one team member is required to bring an updated version of the team Gantt chart to the lab session. As a team, review your Gantt chart and find the most pressing outstanding design objectives. In the space provided, please list these design requirements below (at least three are required). Be prepared to discuss with LAs.

INITIALS__________________

Part B (4 Points): Next Steps and LA interview

In the space provided below, list specific action items, corresponding to the previous incomplete design requirements, that can be worked on during the remainder of the class. Be prepared to share your outstanding requirements and how you will go about working on them with your LA staff during a short interview. Be as specific as possible when planning work flow for the remainder of class. It is required to list team member names for specific tasks, and specify how action items relate to completion of corresponding design requirements. At least one LA will sign off on this document to ensure a successful plan is made.

LA Signature ________________________

Now, get to work! Don’t forget to complete the short reflection below.

Part C (2 Points): Reflection Questions

1.) Which of the outstanding design requirements were you able to make progress on and/or complete during lab today?

2.) How will the team satisfy the incomplete design requirements by competition day?

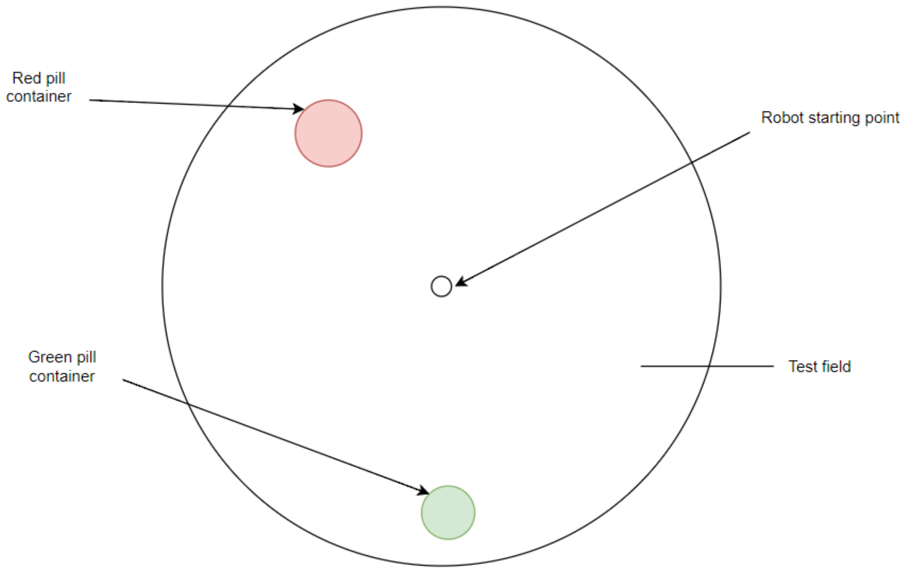
3.) What has been the most challenging portion of the final design project thus far? How would the team combat this challenge differently if they had the opportunity to retry?

Final Design: Pill Sorter

Source

Lab Background and Goal: Local non-profit, AID Design, has requested MIME101/CS160 students to design a pill sorting mechanism for visually impaired patients surrounding the Oregon State University campus.

Final Evaluation Procedure: MIME101/CS160 TAs will randomly select five pills of either red or green. Two colored pill containers (red and green) will be placed at random distances and angles from center within a 360-degree test field. Design teams must create a fully-autonomous system for sorting colored pills into correct corresponding containers. Robots must begin operation at the center of the test field. Students may use Cozmo, Lego Mindstorm, or a combination of both. Pill container and test field dimensions are standardized and can be seen below.



Test field dimensions:

5-foot diameter

Pill Container Dimensions:

-Height: 3 inches
-Container Diameter: 6 Inches

Free Materials:

- Single Lego Mindstorm® Education Core Set
- Cozmo hardware and accompanying software
- Any 3D Printed Material

Purchasable Materials:

- Additional Lego Sensors (1 Point / sensor)
- Additional Lego Elements (1 Point / pound)
- Masking tape (0.25 Point / inch)
- Card Board (0.25 Point / square inch)

Constraints and conditions:

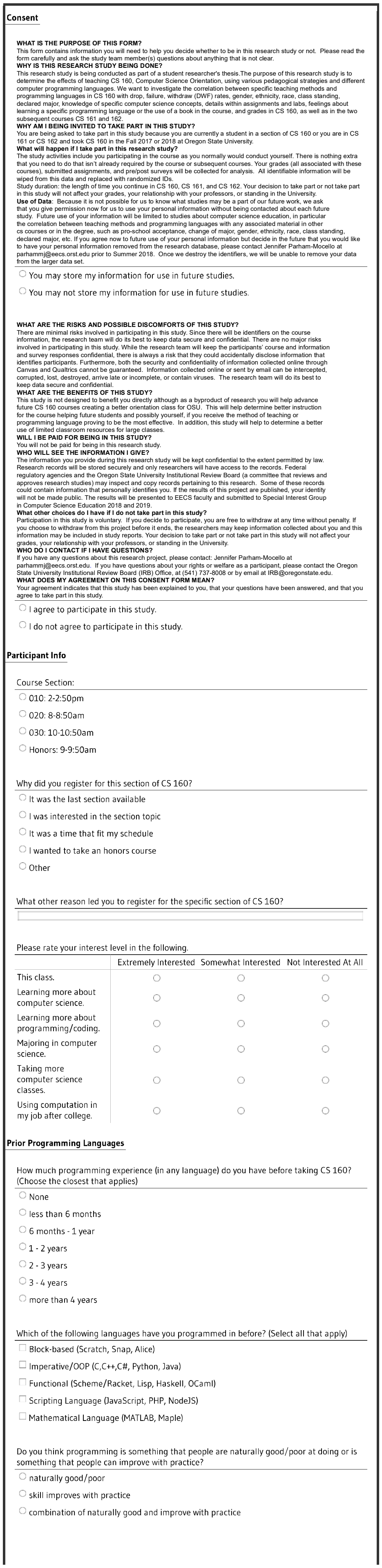
- Budget for Purchasable Materials = 20 points

Evaluation:

Students will be evaluated in seven areas of the design with designated weights:

Grading Criteria	Description	Total Points of Category
Accuracy	The capability of students' designs to accurately sort pills will be graded as a percentage of the five random pills given to the teams by the TA's on test day (4 points per pill)	20
Time	Designs must sort pills in a timely fashion. Students will be given two minutes to sort all five pills. Each twenty second interval over the two minute limit will be a one point penalty.	20
Economics	Economic viability of students' designs will be evaluated based on materials used. Students will have full credit to begin design, and points will be taken away based on outside materials used as outlined in the laboratory assignment.	20
Student Testing	Design teams must create original design testing experiments to validate their designs. Test data must be recorded and submitted to TA's for full credit.	10
Social Justice Design Criteria	Design teams must submit brief write ups for required social justice design criteria to TA's for full credit.	10
3D Printed Component	Students must incorporate a 3D printed component into the final design. The component will be evaluated based on its utility (i.e the component’s usefulness and benefit).	10
Midterm Design Proposal	Teams' midterm design proposals will be graded based on writing quality (see engineering technical writing guidelines), presentation, and accuracy with regard to final designs.	10

Appendix B



Consent

WHAT IS THE PURPOSE OF THIS FORM?
This form contains information you will need to help you decide whether to be in this research study or not. Please read the form carefully and ask the study team member(s) questions about anything that is not clear.

WHY IS THIS RESEARCH STUDY BEING DONE?
This research study is being conducted as part of a student researcher's thesis.The purpose of this research study is to determine the effects of teaching CS 160, Computer Science Orientation, using various pedagogical strategies and different computer programming languages. We want to investigate the correlation between specific teaching methods and programming languages in CS 160 with drop, failure, withdraw (DWF) rates, gender, ethnicity, race, class standing, declared major, knowledge of specific computer science concepts, details within assignments and labs, feelings about learning a specific programming language or the use of a book in the course, and grades in CS 160, as well as in the two subsequent courses CS 161 and 162.

WHY AM I BEING INVITED TO TAKE PART IN THIS STUDY?
You are being asked to take part in this study because you are currently a student in a section of CS 160 or you are in CS 161 or CS 162 and took CS 160 in the Fall 2017 or 2018 at Oregon State University.

What will happen if I take part in this research study?
The study activities include you participating in the course as you normally would conduct yourself. There is nothing extra that you need to do that isn't already required by the course or subsequent courses. Your grades (all associated with these courses), submitted assignments, and pre/post surveys will be collected for analysis. All identifiable information will be wiped from this data and replaced with randomized IDs.
Study duration: the length of time you continue in CS 160, CS 161, and CS 162. Your decision to take part or not take part in this study will not affect your grades, your relationship with your professors, or standing in the University.
Use of Data: Because it is not possible for us to know what studies may be a part of our future work, we ask that you give permission now for us to use your personal information without being contacted about each future study. Future use of your information will be limited to studies about computer science education, in particular the correlation between teaching methods and programming languages with any associated material in other cs courses or in the degree, such as pro-school acceptance, change of major, gender, ethnicity, race, class standing, declared major, etc. If you agree now to future use of your personal information but decide in the future that you would like to have your personal information removed from the research database, please contact Jennifer Parham-Mocello at parhammj@eecs.orst.edu prior to Summer 2018. Once we destroy the identifiers, we will be unable to remove your data from the larger data set.

○ You may store my information for use in future studies.

○ You may not store my information for use in future studies.

WHAT ARE THE RISKS AND POSSIBLE DISCOMFORTS OF THIS STUDY?
There are minimal risks involved in participating in this study. Since there will be identifiers on the course information, the research team will do its best to keep data secure and confidential. There are no major risks involved in participating in this study. While the research team will keep the participants' course and information and survey responses confidential, there is always a risk that they could accidentally disclose information that identifies participants. Furthermore, both the security and confidentiality of information collected online through Canvas and Qualtrics cannot be guaranteed. Information collected online or sent by email can be intercepted, corrupted, lost, destroyed, arrive late or incomplete, or contain viruses. The research team will do its best to keep data secure and confidential.

WHAT ARE THE BENEFITS OF THIS STUDY?
This study is not designed to benefit you directly although as a byproduct of research you will help advance future CS 160 courses creating a better orientation class for OSU. This will help determine better instruction for the course helping future students and possibly yourself, if you receive the method of teaching or programming language proving to be the most effective. In addition, this study will help to determine a better use of limited classroom resources for large classes.

WILL I BE PAID FOR BEING IN THIS STUDY?
You will not be paid for being in this research study.

WHO WILL SEE THE INFORMATION I GIVE?
The information you provide during this research study will be kept confidential to the extent permitted by law. Research records will be stored securely and only researchers will have access to the records. Federal regulatory agencies and the Oregon State University Institutional Review Board (a committee that reviews and approves research studies) may inspect and copy records pertaining to this research. Some of these records could contain information that personally identifies you. If the results of this project are published, your identity will not be made public. The results will be presented to EECS faculty and submitted to Special Interest Group in Computer Science Education 2018 and 2019.

What other choices do I have if I do not take part in this study?
Participation in this study is voluntary. If you decide to participate, you are free to withdraw at any time without penalty. If you choose to withdraw from this project before it ends, the researchers may keep information collected about you and this information may be included in study reports. Your decision to take part or not take part in this study will not affect your grades, your relationship with your professors, or standing in the University.

WHO DO I CONTACT IF I HAVE QUESTIONS?
If you have any questions about this research project, please contact: Jennifer Parham-Mocello at parhammj@eecs.orst.edu. If you have questions about your rights or welfare as a participant, please contact the Oregon State University Institutional Review Board (IRB) Office, at (541) 737-8008 or by email at IRB@oregonstate.edu.

WHAT DOES MY AGREEMENT ON THIS CONSENT FORM MEAN?
Your agreement indicates that this study has been explained to you, that your questions have been answered, and that you agree to take part in this study.

○ I agree to participate in this study.

○ I do not agree to participate in this study.

Participant Info

Course Section:

○ 010: 2-2:50pm

○ 020: 8-8:50am

○ 030: 10-10:50am

○ Honors: 9-9:50am

Why did you register for this section of CS 160?

○ It was the last section available

○ I was interested in the section topic

○ It was a time that fit my schedule

○ I wanted to take an honors course

○ Other

What other reason led you to register for the specific section of CS 160?

Please rate your interest level in the following.

	Extremely Interested	Somewhat Interested	Not Interested At All
This class.	○	○	○
Learning more about computer science.	○	○	○
Learning more about programming/coding.	○	○	○
Majoring in computer science.	○	○	○
Taking more computer science classes.	○	○	○
Using computation in my job after college.	○	○	○

Prior Programming Languages

How much programming experience (in any language) do you have before taking CS 160? (Choose the closest that applies)

○ None

○ less than 6 months

○ 6 months - 1 year

○ 1 - 2 years

○ 2 - 3 years

○ 3 - 4 years

○ more than 4 years

Which of the following languages have you programmed in before? (Select all that apply)

□ Block-based (Scratch, Snap, Alice)

□ Imperative/OOP (C,C++,C#, Python, Java)

□ Functional (Scheme/Racket, Lisp, Haskell, OCaml)

□ Scripting Language (JavaScript, PHP, NodeJS)

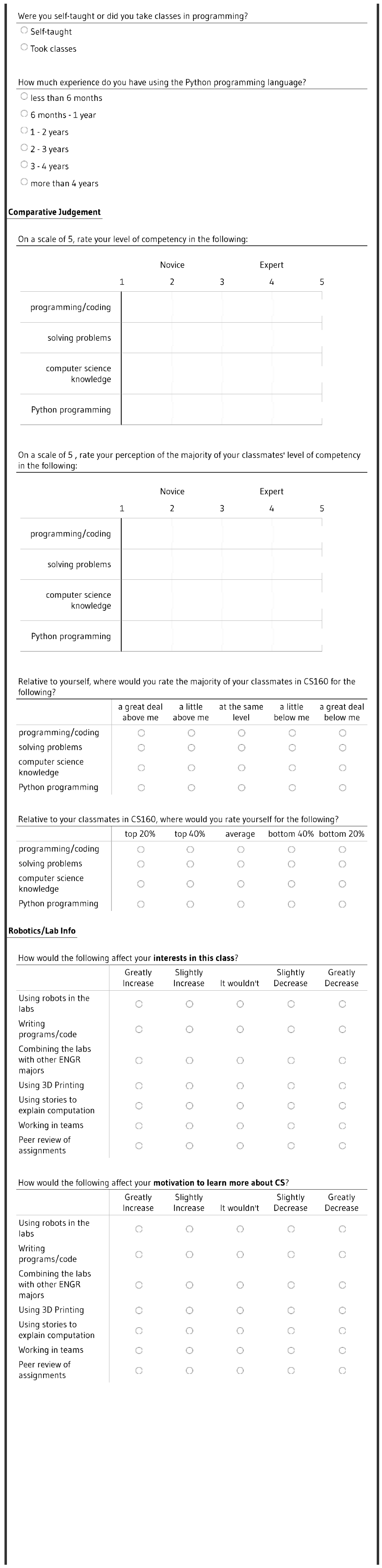
□ Mathematical Language (MATLAB, Maple)

Do you think programming is something that people are naturally good/poor at doing or is something that people can improve with practice?

○ naturally good/poor

○ skill improves with practice

○ combination of naturally good and improve with practice



Were you self-taught or did you take classes in programming?

○ Self-taught

○ Took classes

How much experience do you have using the Python programming language?

○ less than 6 months

○ 6 months - 1 year

○ 1 - 2 years

○ 2 - 3 years

○ 3 - 4 years

○ more than 4 years

Comparative Judgement

On a scale of 5, rate your level of competency in the following:

		Novice		Expert	
	1	2	3	4	5
programming/coding					
solving problems					
computer science knowledge					
Python programming					

On a scale of 5 , rate your perception of the majority of your classmates' level of competency in the following:

		Novice		Expert	
	1	2	3	4	5
programming/coding					
solving problems					
computer science knowledge					
Python programming					

Relative to yourself, where would you rate the majority of your classmates in CS160 for the following?

	a great deal above me	a little above me	at the same level	a little below me	a great deal below me
programming/coding	○	○	○	○	○
solving problems	○	○	○	○	○
computer science knowledge	○	○	○	○	○
Python programming	○	○	○	○	○

Relative to your classmates in CS160, where would you rate yourself for the following?

	top 20%	top 40%	average	bottom 40%	bottom 20%
programming/coding	○	○	○	○	○
solving problems	○	○	○	○	○
computer science knowledge	○	○	○	○	○
Python programming	○	○	○	○	○

Robotics/Lab Info

How would the following affect your **interests in this class**?

	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease
Using robots in the labs	○	○	○	○	○
Writing programs/code	○	○	○	○	○
Combining the labs with other ENGR majors	○	○	○	○	○
Using 3D Printing	○	○	○	○	○
Using stories to explain computation	○	○	○	○	○
Working in teams	○	○	○	○	○
Peer review of assignments	○	○	○	○	○

How would the following affect your **motivation to learn more about CS**?

	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease
Using robots in the labs	○	○	○	○	○
Writing programs/code	○	○	○	○	○
Combining the labs with other ENGR majors	○	○	○	○	○
Using 3D Printing	○	○	○	○	○
Using stories to explain computation	○	○	○	○	○
Working in teams	○	○	○	○	○
Peer review of assignments	○	○	○	○	○

How would the following affect your **motivation to learn more about programming/coding**?

	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease
Using robots in the labs	○	○	○	○	○
Writing programs/code	○	○	○	○	○
Combining the labs with other ENGR majors	○	○	○	○	○
Using 3D Printing	○	○	○	○	○
Using stories to explain computation	○	○	○	○	○
Working in teams	○	○	○	○	○
Peer review of assignments	○	○	○	○	○

What size team would you prefer to work with in the following environments?

	Individually	2 people	3 people	4 people	more than 4 people
Labs	○	○	○	○	○
Assignments	○	○	○	○	○
Class Exercises	○	○	○	○	○
Term Project	○	○	○	○	○

Your lab and project teams will have students from MIME 101, the Mechanical, Industrial, and Manufacturing Engineering orientation course. Do you think a more diverse group of majors will add to your ability to learn more in this class?

○ Yes

○ Maybe

○ No

Have you ever ...

	Yes	No
Used a Cozmo robot	○	○
Used a Lego Mindstorm robot	○	○
Designed anything for 3D printing	○	○

Algorithms

An algorithm ...

	What is your answer?			How confident are you in your answer?			
	Always	Sometimes	Never	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
... is given in a language.	○	○	○	○	○	○	○
... solves a problem.	○	○	○	○	○	○	○
... terminates.	○	○	○	○	○	○	○
... takes time to execute.	○	○	○	○	○	○	○

What are the properties of an algorithm?

	What is your answer?		How confident are you in your answer?			
	must	should	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
All instructions in an algorithm __________ be effective.	○	○	○	○	○	○
An algorithm __________ have good runtime complexity.	○	○	○	○	○	○

How many algorithms are there for a problem?

	What is your answer?			How confident are you in your answer?			
	exactly 1	1 or more	0 or more	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
For every problem there is/are __________ algorithms to solve it.	○	○	○	○	○	○	○

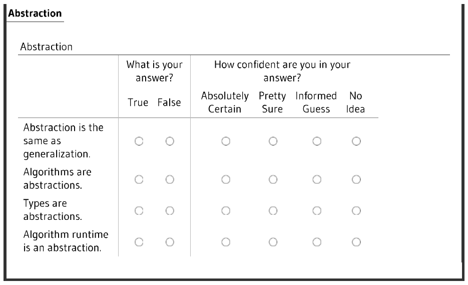
How is the efficiency of an algorithm measured?

	What is your answer?			How confident are you in your answer?			
	in seconds	in number of steps	as a growth rate	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
The efficiency of an algorithm is measured __________.	○	○	○	○	○	○	○

Representation

Representation

	What is your answer?		How confident are you in your answer?			
	True	False	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
A data type defines how to store data.	○	○	○	○	○	○
Different data structures can implement one data type.	○	○	○	○	○	○
A data type can be used to implement a data structure.	○	○	○	○	○	○
Recursion is more general than loops.	○	○	○	○	○	○



Abstraction

Abstraction

	What is your answer?		How confident are you in your answer?			
	True	False	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
Abstraction is the same as generalization.	○	○	○	○	○	○
Algorithms are abstractions.	○	○	○	○	○	○
Types are abstractions.	○	○	○	○	○	○
Algorithm runtime is an abstraction.	○	○	○	○	○	○

Appendix C

Participant Info

First and Last Name:

____

Course Section:

○ 010: 2-2:50pm

○ 020: 8-8:50am

○ 030: 10-10:50am

○ Honors: 9-9:50am

Please rate your interest level in the following.

	Extremely Interested	Somewhat Interested	Not Interested At All
This class.	○	○	○
Learning more about computer science.	○	○	○
Learning more about programming/coding.	○	○	○
Majoring in computer science.	○	○	○
Taking more computer science classes.	○	○	○
Using computation in my job after college.	○	○	○

If you were interested in majoring in computer science and you are not as interested anymore, why did you change?

____

Assessment of Teaching Method and Programming Language

How confident are you at using Python?

○ Extremely confident

○ Somewhat confident

○ Not confident at all

How much did you enjoy programming in Python in CS 160?

○ A great deal

○ Somewhat

○ Not at all

How much did you enjoy programming in Scratch in CS 160?

○ A great deal

○ Somewhat

○ Not at all

Did you have prior coding experience before taking CS 160?

○ Yes

○ No

How do you think your prior programming experience helped or hurt you in the class?

○ Helped a lot

○ Helped a little

○ Hurt a little

○ Hurt a lot

How do you think your lack of prior programming experience helped or hurt you in the class?

○ Helped a lot

○ Helped a little

○ Hurt a little

○ Hurt a lot

How do you feel about the approach used to teach your section of CS 160?

○ Like a great deal

○ Like somewhat

○ Dislike somewhat

○ Dislike a great deal

Rate your level of agreement on the following statements.

	Strongly agree	Agree	Disagree	Strongly disagree
The computational activities without the use of a computer helped me learn.	○	○	○	○
The computational activities without the use of a computer motivated me to learn more about computer science.	○	○	○	○
The computational activities without the use of a computer were engaging.	○	○	○	○
The coding activities on the computer helped me learn.	○	○	○	○
The coding activities on the computer motivated me to learn more about computer science.	○	○	○	○
The coding activities on the computer were engaging.	○	○	○	○

Did you find the textbook, Once Upon an Algorithm, useful to your learning?

○ Extremely useful

○ Very useful

○ Moderately useful

○ Not at all useful

Did you find the textbook, Once Upon an Algorithm, interesting?

○ Extremely interesting

○ Very interesting

○ Moderately interesting

○ Not interesting at all

What was your favorite chapter in the textbook, Once Upon an Algorithm?

○ Chapter 1: A Path to Understanding Computation
○ Chapter 2: Walk the Walk: When Computation Really Happens
○ Chapter 3: The Mystery of Signs
○ Chapter 4: Detective's Notebook: Accessory after the Fact
○ Chapter 5: The Search for the Perfect Data Structure
○ Chapter 6: Sorting out Sorting
○ Chapter 7: Mission Intractable
○ Chapter 8: The Prism of Language
○ Chapter 9: Finding the Right Tone: Sound Meaning
○ Chapter 10: Weather, Rinse, Repeat
○ Chapter 11: Happy Ending Not Guaranteed
○ Chapter 12: A Stitch in Time Computes Fine
○ Chapter 14: The Magical Type
○ Chapter 15: A Bird's Eye View: Abstracting from Details
○ N/A or didn't read

What was your LEAST favorite chapter in the textbook, Once Upon an Algorithm?

○ Chapter 1: A Path to Understanding Computation
○ Chapter 2: Walk the Walk: When Computation Really Happens
○ Chapter 3: The Mystery of Signs
○ Chapter 4: Detective's Notebook: Accessory after the Fact
○ Chapter 5: The Search for the Perfect Data Structure
○ Chapter 6: Sorting out Sorting
○ Chapter 7: Mission Intractable
○ Chapter 8: The Prism of Language
○ Chapter 9: Finding the Right Tone: Sound Meaning
○ Chapter 10: Weather, Rinse, Repeat
○ Chapter 11: Happy Ending Not Guaranteed
○ Chapter 12: A Stitch in Time Computes Fine
○ Chapter 14: The Magical Type
○ Chapter 15: A Bird's Eye View: Abstracting from Details
○ N/A or didn't read

How did the following affect your...

	engagement in this class?					motivation to learn more about CS?					motivation to learn more about programming/coding?				
	Greatly Increase	Slightly Increase	It didn't	Slightly Decrease	Greatly Decrease	Greatly Increase	Slightly Increase	It didn't	Slightly Decrease	Greatly Decrease	Greatly Increase	Slightly Increase	It didn't	Slightly Decrease	Greatly Decrease
Using robots in the labs	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Writing programs/code	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Combining the labs with MIME majors	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Using 3D Printing	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Using stories to explain computation	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Working in teams	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Peer review of assignments	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

How would the following affect your...

	engagement in this class?					motivation to learn more about CS?					motivation to learn more about programming/coding?				
	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease	Greatly Increase	Slightly Increase	It wouldn't	Slightly Decrease	Greatly Decrease
Using robots in the labs	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Writing programs/code	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Combining the labs with other ENGR majors	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Using 3D Printing	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Using stories to explain computation	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Working in teams	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Peer review of assignments	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Do you think there is a difference between Computer Science and coding?

○ Yes
○ No

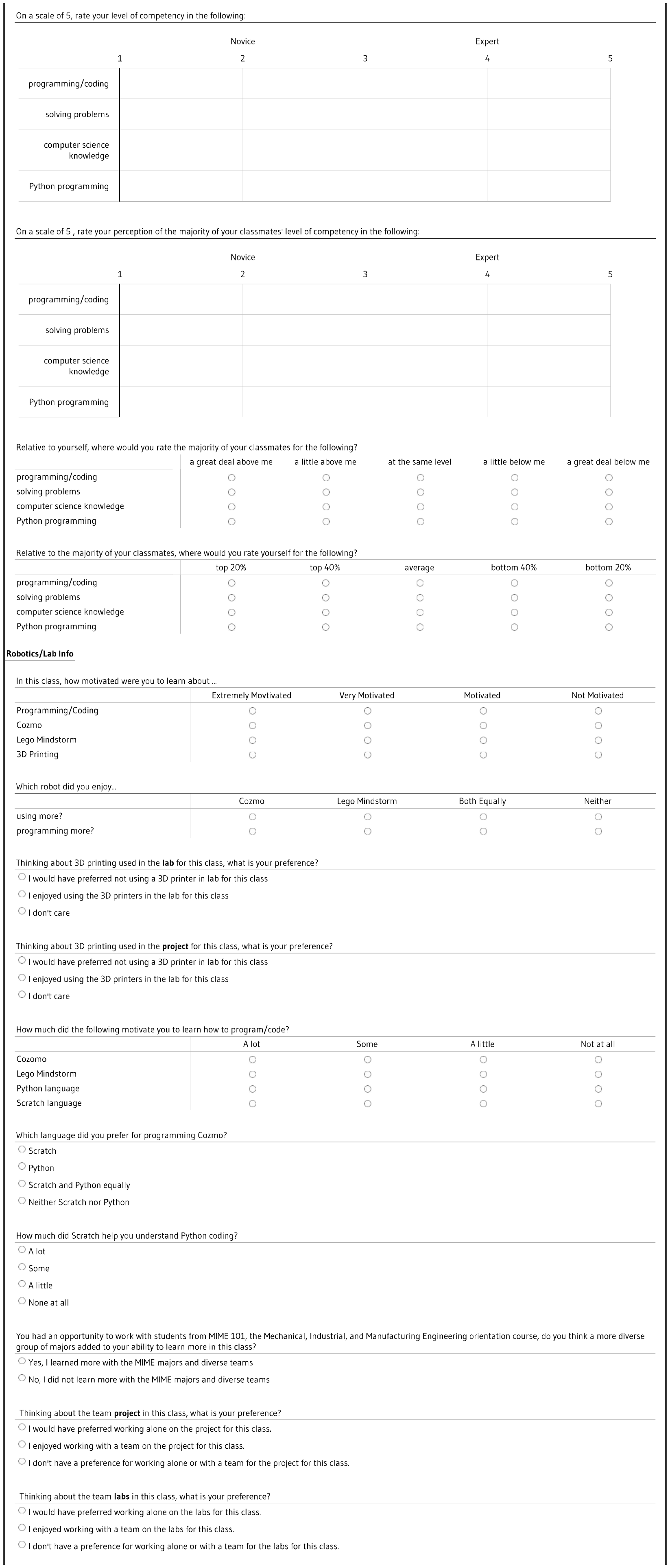
As a result of taking this class, how well do you understand the following computer science concepts?

	Extremely well	Very well	Moderately well	Slightly well	Not well at all
Representation	○	○	○	○	○
Conditional Execution	○	○	○	○	○
Repetition	○	○	○	○	○
Decomposition	○	○	○	○	○
Data Structures	○	○	○	○	○

Overall, how much did you enjoy the class?

○ A great deal
○ A lot
○ A moderate amount
○ A little
○ None at all

Comparative Judgement



On a scale of 5, rate your level of competency in the following:

		Novice		Expert	
	1	2	3	4	5
programming/coding					
solving problems					
computer science knowledge					
Python programming					

On a scale of 5 , rate your perception of the majority of your classmates' level of competency in the following:

		Novice		Expert	
	1	2	3	4	5
programming/coding					
solving problems					
computer science knowledge					
Python programming					

Relative to yourself, where would you rate the majority of your classmates for the following?

	a great deal above me	a little above me	at the same level	a little below me	a great deal below me
programming/coding	○	○	○	○	○
solving problems	○	○	○	○	○
computer science knowledge	○	○	○	○	○
Python programming	○	○	○	○	○

Relative to the majority of your classmates, where would you rate yourself for the following?

	top 20%	top 40%	average	bottom 40%	bottom 20%
programming/coding	○	○	○	○	○
solving problems	○	○	○	○	○
computer science knowledge	○	○	○	○	○
Python programming	○	○	○	○	○

Robotics/Lab Info

In this class, how motivated were you to learn about ...

	Extremely Movtivated	Very Motivated	Motivated	Not Motivated
Programming/Coding	○	○	○	○
Cozmo	○	○	○	○
Lego Mindstorm	○	○	○	○
3D Printing	○	○	○	○

Which robot did you enjoy...

	Cozmo	Lego Mindstorm	Both Equally	Neither
using more?	○	○	○	○
programming more?	○	○	○	○

Thinking about 3D printing used in the **lab** for this class, what is your preference?

○ I would have preferred not using a 3D printer in lab for this class

○ I enjoyed using the 3D printers in the lab for this class

○ I don't care

Thinking about 3D printing used in the **project** for this class, what is your preference?

○ I would have preferred not using a 3D printer in lab for this class

○ I enjoyed using the 3D printers in the lab for this class

○ I don't care

How much did the following motivate you to learn how to program/code?

	A lot	Some	A little	Not at all
Cozomo	○	○	○	○
Lego Mindstorm	○	○	○	○
Python language	○	○	○	○
Scratch language	○	○	○	○

Which language did you prefer for programming Cozmo?

○ Scratch

○ Python

○ Scratch and Python equally

○ Neither Scratch nor Python

How much did Scratch help you understand Python coding?

○ A lot

○ Some

○ A little

○ None at all

You had an opportunity to work with students from MIME 101, the Mechanical, Industrial, and Manufacturing Engineering orientation course, do you think a more diverse group of majors added to your ability to learn more in this class?

○ Yes, I learned more with the MIME majors and diverse teams

○ No, I did not learn more with the MIME majors and diverse teams

Thinking about the team **project** in this class, what is your preference?

○ I would have preferred working alone on the project for this class.

○ I enjoyed working with a team on the project for this class.

○ I don't have a preference for working alone or with a team for the project for this class.

Thinking about the team **labs** in this class, what is your preference?

○ I would have preferred working alone on the labs for this class.

○ I enjoyed working with a team on the labs for this class.

○ I don't have a preference for working alone or with a team for the labs for this class.

After working in teams in this class, what size team do you prefer for the following environments?

	No team	2 people	3 people	4 people	more than 4 people
Labs	○	○	○	○	○
Assignments	○	○	○	○	○
Class Exercises	○	○	○	○	○
Term Project	○	○	○	○	○

Algorithms

An algorithm ...

	What is your answer?			How confident are you in your answer?			
	Always	Sometimes	Never	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
... is given in a language.	○	○	○	○	○	○	○
... solves a problem.	○	○	○	○	○	○	○
... terminates.	○	○	○	○	○	○	○
... takes time to execute.	○	○	○	○	○	○	○

What are the properties of an algorithm?

	What is your answer?		How confident are you in your answer?			
	must	should	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
All instructions in an algorithm ________ be effective.	○	○	○	○	○	○
An algorithm ________ have good runtime complexity.	○	○	○	○	○	○

How many algorithms are there for a problem?

	What is your answer?			How confident are you in your answer?			
	exactly 1	1 or more	0 or more	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
For every problem there is/are ________ algorithms to solve it.	○	○	○	○	○	○	○

How is the efficiency of an algorithm measured?

	What is your answer?			How confident are you in your answer?			
	in seconds	in number of steps	as a growth rate	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
The efficiency of an algorithm is measured ________.	○	○	○	○	○	○	○

Representation

Representation

	What is your answer?		How confident are you in your answer?			
	True	False	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
A data type defines how to store data.	○	○	○	○	○	○
Different data structures can implement one data type.	○	○	○	○	○	○
A data type can be used to implement a data structure.	○	○	○	○	○	○
Recursion is more general than loops.	○	○	○	○	○	○

Abstraction

Abstraction

	What is your answer?		How confident are you in your answer?			
	True	False	Absolutely Certain	Pretty Sure	Informed Guess	No Idea
Abstraction is the same as generalization.	○	○	○	○	○	○
Algorithms are abstractions.	○	○	○	○	○	○
Types are abstractions.	○	○	○	○	○	○
Algorithm runtime is an abstraction.	○	○	○	○	○	○