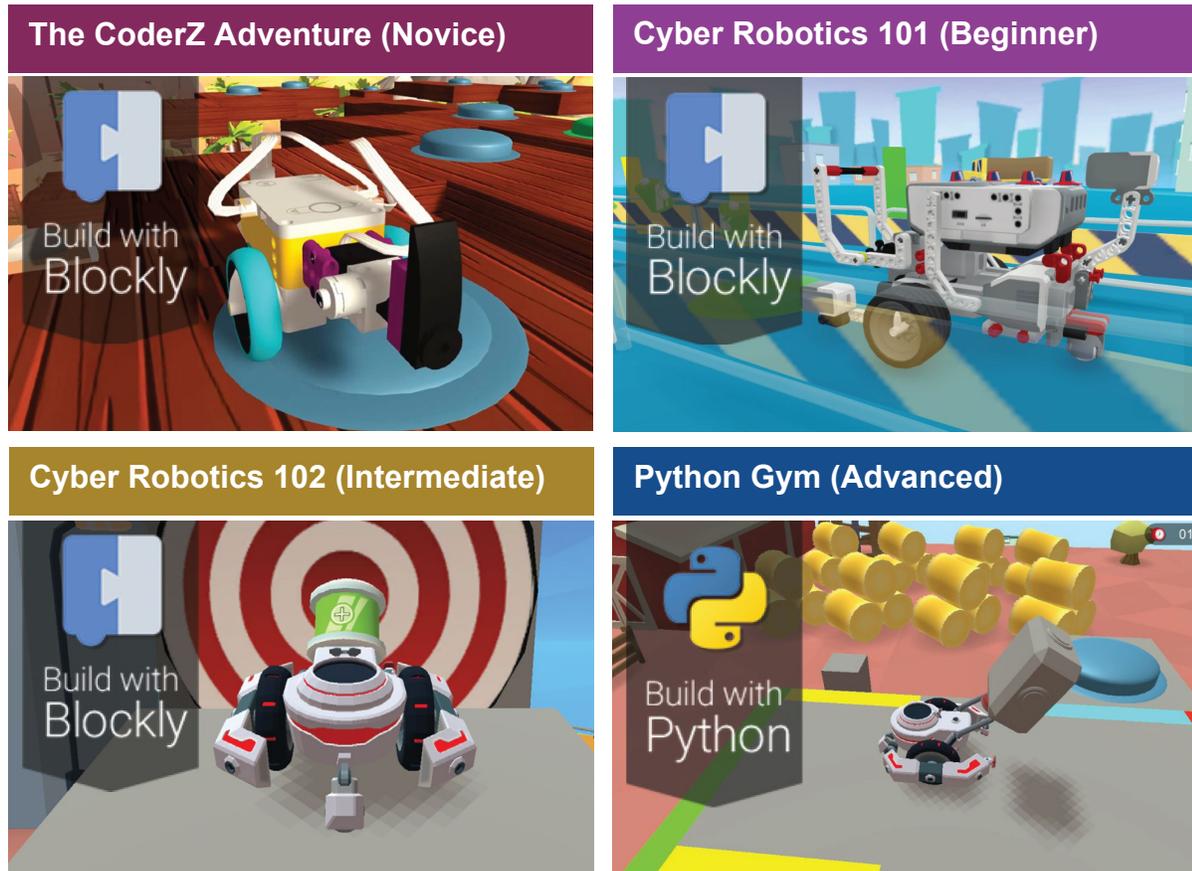


CoderZ™ Content Framework

CoderZ is an engaging self-paced virtual learning platform that provides interactive lessons and activities designed to teach students principles of coding within the exciting and challenging context of robotics.

There are four courses available from which schools/districts may select to implement:

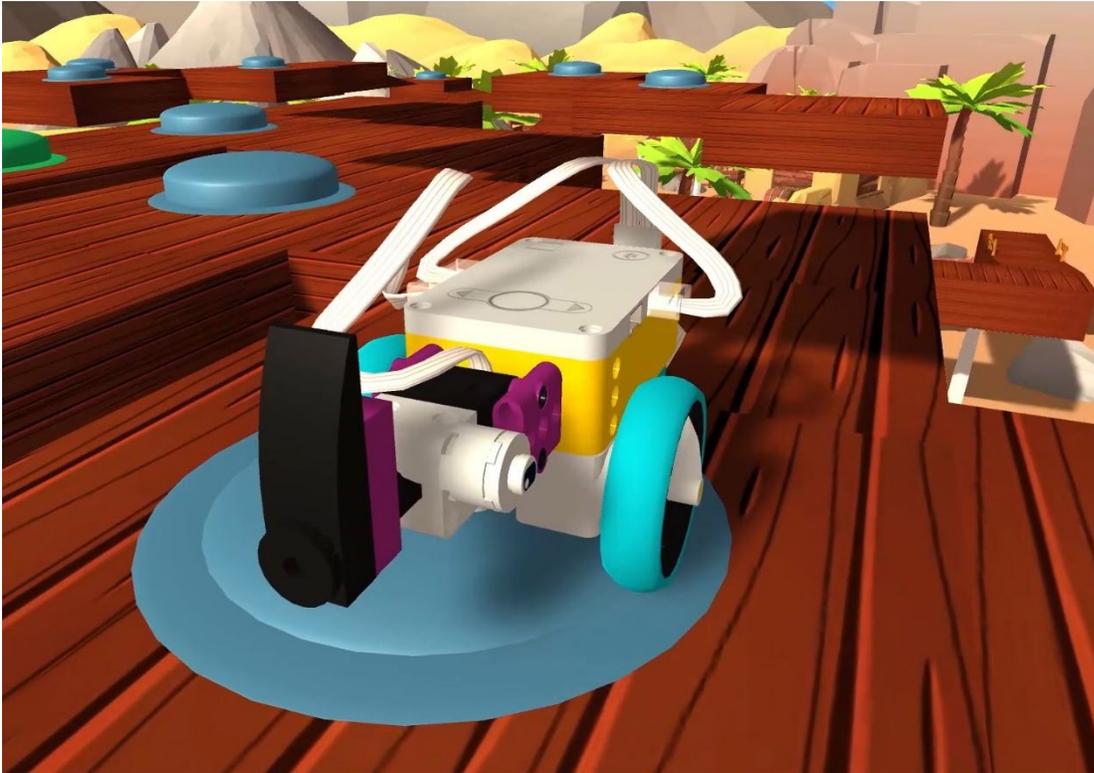


The CoderZ Content Framework provides an overview of the standards, skills, and concepts taught and supported within the learning platform, by the available CoderZ teacher resources - including the teacher/course guides, and with the integration of teacher lead lessons and activities as supported by CoderZ's professional development. Each course's content is individually framed for efficient referencing.

TABLE OF CONTENT

CODERZ CONTENT FRAMEWORK -----	1
The CoderZ Adventure -----	3
Cyber Robotics 101-----	7
Cyber Robotics 102-----	13
Python Gym-----	19

The CoderZ Adventure



In this introductory coding course students are introduced to the basic principles of coding within the context of robotics. While participating in a series of increasingly complex tasks students learn about the relationship between hardware and software, the role of mathematical modeling and data in computational problem solving, how basic programming constructs can be combined to create complex algorithms, as well as develop a foundation in key programming practices.

Standards, Skills, and Concepts

CSTA

Computer Systems

- 1B-CS-01 Describe how internal and external parts of computing devices function to form a system.
- 1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.

Data and Analysis

- 1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predicate outcomes, or communicate an idea.

Algorithms and Programming

- 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate
- 1B-AP-10 Create programs that include sequences, events, loops, and conditionals.
- 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
- 1B-AP-13 Use an interactive process to plan the development of a program by including others' perspectives and considering user preferences.
- 1B-AP-15 Test and debug (identify and fix errors) in a program or algorithm to ensure it runs as intended.

Impacts of Computing

- 1B-IC-18 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

NGSS

- 3-5-ETS1-1 Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.
- 3-5-ETS1-2 Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.
- 3-5-ETS1-37 Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.

K-12 Computer Science Framework

Computing Systems

- Troubleshooting - Check connections and power to resolve common issues. Explain and demonstrate how rebooting a machine is commonly an effective strategy.

Algorithms and Programming

- Algorithms - Compare algorithms and select the one most appropriate for a specific context/task.
- Control - Recognize patterns and effectively use loops to enhance the efficiency of writing code.
- Program Development - Use an iterative process involving design, implementation, and review when developing code.

Impacts of Computing

- Social Interaction - Analyze how computing technology's facilitation of communication and innovations influences social institutions such as family, education, and the economy.

CC Math

Mathematical Practice

- MP1 Make sense of problems and persevere in solving them.
- MP2 Reason abstractly and quantitatively.
- MP3 Construct viable arguments and critique the reasoning of others.
- MP4 Model with mathematics.
- MP5 Use appropriate tools strategically.
- MP6 Attend to precision.
- MP7 Look for and make use of structure.
- MP8 Look for and express regularity in repeated reasoning.

21st Century Skills

- Creativity and Innovation
- Critical Thinking and Problem Solving
- Communication
- Collaboration

Computational Thinking Skills

- Decomposition
- Pattern Recognition
- Algorithm Design

Robotics Concepts and Skills

- Motion Planning - Direction

Cyber Robotics 101



In this beginner's coding course students are introduced to the principles of coding within the context of robotics. While engaged in a series of activities with increasing complexity students learn the essential principles of robotics including motion planning, motion planning with sensors, the relationship between hardware and software, the role of mathematical modeling and data in computational problem solving, the importance of quality planning, as well as develop the ability to execute good programming practices. CoderZ and supporting integrated content is sequenced to meet STEAM objectives and is designed to provide students authentic learning opportunities to increase interest and engagement while moving from acquisition of knowledge to transference of learning.

Standards, Skills, and Concepts

CSTA

Computer Systems

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

Algorithms and Programming

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to use.

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

Impacts of Computing

- 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies.
- 2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.

NGSS

- MS-ETS1-2 Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
- MS-ETS1-3 Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

K-12 Computer Science Framework

Computing Systems

- Devices - Examine the interaction between humans and computing devices. Evaluate and analyze the advantages, disadvantages, and recognize the unintended consequences.
- Troubleshooting - Deploy a structured process to troubleshoot problems within a system to ensure potential solutions are not overlooked and simple issues are resolved.

Algorithms and Programming

- Algorithms - Design, test, and debug algorithms that are readable and easy to follow.
- Variables - Understand and use variables to represent and process data to produce varying outputs.
- Control - Use loops, conditions, and other control structures to create more complex programs.
- Modularity - Use functions to make code easier to reuse and read.

Networks and the Internet

- Cybersecurity - Explain the importance of protecting information sent and received across networks from unauthorized access and modification (encryption and Hypertext Transfer Protocol Secure - HTTPS).

Data and Analysis

- Collection - Discuss how using sensors with a robot to collect data regarding its environment and make decisions about the next steps to take is similar to the human process and how robots and humans interact.

CC Math

Mathematical Practice

- MP1 Make sense of problems and persevere in solving them.
- MP2 Reason abstractly and quantitatively.
- MP3 Construct viable arguments and critique the reasoning of others.
- MP4 Model with mathematics.
- MP5 Use appropriate tools strategically.
- MP6 Attend to precision.
- MP7 Look for and make use of structure.
- MP8 Look for and express regularity in repeated reasoning.

21st Century Skills

Learning and Innovation Skills

- Creativity and Innovation
- Critical Thinking and Problem Solving
- Communication and Collaboration

Information, Media and Technology Skills

- Information Literacy
- Information and Communications Technology (ICT)

Life and Career Skills

- Flexibility and Adaptability
- Initiative and Self-direction
- Social and Cross-Cultural Skills
- Productivity and Accountability
- Leadership and Responsibility

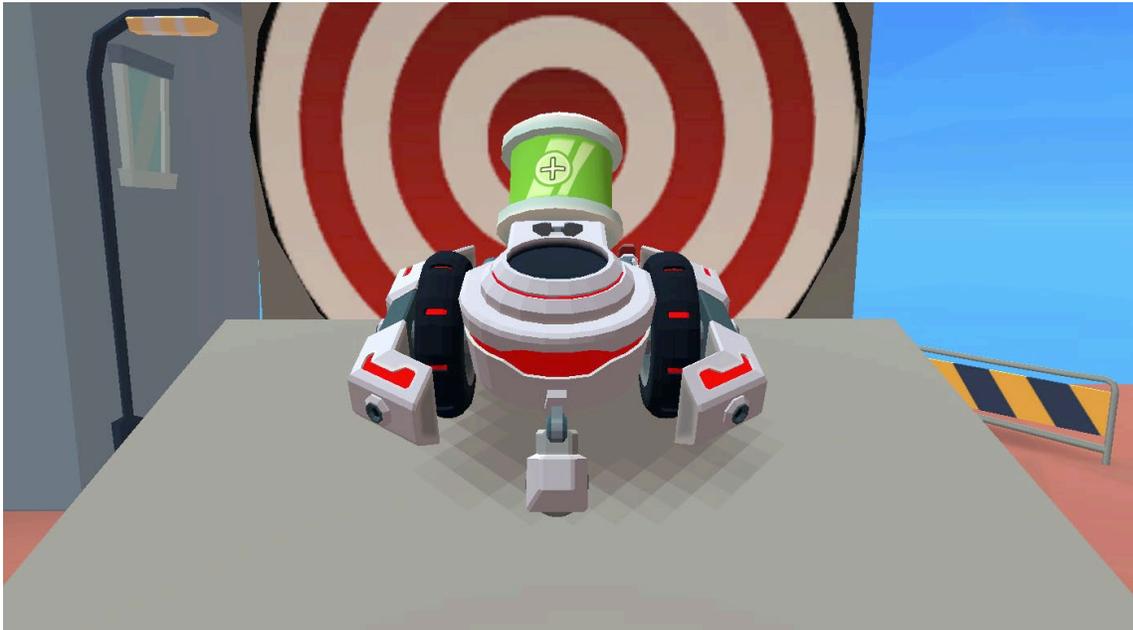
Computational Thinking Skills

- Decomposition
- Pattern Recognition
- Algorithm Design
- Abstraction

Robotics Concepts and Skills

- Motion Planning
 - Direction
 - Power/Speed
 - Distance
 - Duration
 - Arm
- Sensor-based Motion Planning
- Motors and Optical Encoders
- Sensors and Controllers
 - Touch - collision
 - Gyroscope - rotation
 - Ultrasonic - distance
 - Light/Color - surface color/brightness

Cyber Robotics 102



Cyber Robotics 102 (CR102) is a continuation of the Cyber Robotics 101 course. This course teaches STEM and coding topics using gamification in a physics-based environment. Through a series of activities with increasing complexity CR102 introduces autonomous systems, teaches scanning and mapping an environment, error correction methods, and different system control algorithms. Cyber Robotics 102 provides a deeper coding experience with more opportunities to use conditionals, variables, functions, etc. CoderZ and supporting integrated content is sequenced to meet STEAM objectives and is designed to provide students authentic learning opportunities to increase interest and engagement while moving from acquisition of knowledge to transference of learning. By the end of the course students will understand the forces of physics acting on robots and the influence those forces have on a robot's performance. Students will effectively control and program a robot to interact with objects around it and safely navigate through changing environments.

Standards, Skills, and Concepts

CSTA

Computer Systems

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

Algorithms and Programming

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to use.

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

Impacts of Computing

- 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies.
- 2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.

NGSS

- MS-ETS1-2 Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
- MS-ETS1-3 Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

K-12 Computer Science Framework

Computing Systems

- Devices - Examine the interaction between humans and computing devices. Evaluate and analyze the advantages, disadvantages, and recognize the unintended consequences.
- Troubleshooting - Deploy a structured process to troubleshoot problems within the system to ensure potential solutions are not overlooked and simple issues are resolved.
- Algorithms and Programming
- Algorithms - Design, test, and debug algorithms that are readable and easy to follow.
- Variables - Understand and use variables to represent and process data to produce varying outputs.
- Control - Use loops, conditions, and other control structures to create more complex programs.
- Modularity - Use functions to make code easier to reuse and read.

Networks and the Internet

- Cybersecurity - Explain the importance of protecting information sent and received across networks from unauthorized access and modification (encryption and Hypertext Transfer Protocol Secure - HTTPS).

Data and Analysis

- Collection - Discuss how using sensors with a robot to collect data regarding its environment and make decisions about the next steps to take is similar to the human process and how robots and humans interact.

CC Math

Mathematical Practice

- MP1 Make sense of problems and persevere in solving them.
- MP2 Reason abstractly and quantitatively.
- MP3 Construct viable arguments and critique the reasoning of others.
- MP4 Model with mathematics.
- MP5 Use appropriate tools strategically.
- MP6 Attend to precision.
- MP7 Look for and make use of structure.
- MP8 Look for and express regularity in repeated reasoning.

21st Century Skills

Learning and Innovation Skills

- Creativity and Innovation
- Critical Thinking and Problem Solving
- Communication and Collaboration

Information, Media and Technology Skills

- Information Literacy
- Information and Communications Technology (ICT)

Life and Career Skills

- Flexibility and Adaptability
- Initiative and Self-direction
- Social and Cross-Cultural Skills
- Productivity and Accountability
- Leadership and Responsibility

Computational Thinking Skills

- Decomposition
- Pattern Recognition
- Algorithm Design
- Abstraction

Robotics Concepts and Skills

- Motion Planning
 - Direction
 - Power/Speed
 - Distance
 - Duration
 - Brake
 - Arms
- Sensor-based Motion Planning
- Motors and Optical Encoders
- Sensors and Controllers
 - Touch - collision
 - Gyroscope - rotation and tilts (x, y, z)
 - Ultrasonic - distance
 - Light/Color - surface color/brightness

Python Gym



Python Gym is an advanced course providing teachers and students the opportunity to code in Python, a text-based computer language, using the CoderZ physically-based environment introduced in Cyber Robotics 102. While engaged in a series of activities with increasing complexity students learn the essential principles of text-based coding including syntax and debugging and continue to develop best programming practices for robotics including; motion planning, motion planning with sensors, physics, computational problem solving, and the importance of quality planning. Python Gym provides opportunities to use conditionals, variables, functions, modules and more.

CoderZ and supporting integrated content is sequenced to meet STEAM objectives and is designed to provide students authentic learning opportunities to increase interest and engagement while moving from acquisition of knowledge to transference of learning.

Standards, Skills, and Concepts

CSTA

Computer Systems

- 2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
- 3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

Algorithms and Programming

- 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.
- 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
- 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-14 Create procedures with parameters to organize code and make it easier to use.
- 2-AP-17 Systematically test and refine programs using a range of test cases.
- 2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 2-AP-19 Document programs in order to make them easier to follow, test, and debug.

- 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
- 3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.
- 3B-AP-23 Evaluate key qualities of a program through a process such as code review.

Impacts of Computing

- 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies.
- 2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
- 3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
- 3B-IC-27 Predict how computational innovations that have revolutionized aspects of our culture might evolve.

NGSS

- MS-ETS1-2 Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
- MS-ETS1-3 Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

K-12 Computer Science Framework

Computing Systems

- Devices – Examine
- e the interaction between humans and computing devices. Evaluate and analyze the advantages, disadvantages, and recognize the unintended consequences.
- Troubleshooting - Deploy a structured process to troubleshoot problems within the system to ensure potential solutions are not overlooked and simple issues are resolved.
- Hardware and Software - Explore the interactions between hardware, software and user of a computing system and evaluate the flow of information.

Algorithms and Programming

- Algorithms - Design, test, and debug algorithms that are readable and easy to follow. Evaluate and select algorithms based on performance, reusability, and ease of implementation.
- Variables - Understand and use variables to represent and process data to produce varying outputs.
Use data structures to effectively manage program complexity (based on functionality, storage, and performance tradeoffs).
- Control - Use loops, conditions, and other control structures to create more complex programs.
Evaluate control structures and control structure combinations considering the tradeoffs related to implementation, readability, and program performance.
- Modularity - Use functions to make code easier to reuse and read.
Use modules to develop and manage complex tasks/programs.

Networks and the Internet

- Cybersecurity - Explain the importance of protecting information sent and received across networks from unauthorized access and modification (encryption and Hypertext Transfer Protocol Secure - HTTPS).

Data and Analysis

- Collection - Discuss how using sensors with a robot to collect data regarding its environment and make decisions about the next steps to take is similar to the human process and how robots and humans interact.

CC Math

Mathematical Practice

- MP1 Make sense of problems and persevere in solving them.
- MP2 Reason abstractly and quantitatively.
- MP3 Construct viable arguments and critique the reasoning of others.
- MP4 Model with mathematics.
- MP5 Use appropriate tools strategically.
- MP6 Attend to precision.
- MP7 Look for and make use of structure.
- MP8 Look for and express regularity in repeated reasoning.

21st Century Skills

Learning and Innovation Skills

- Creativity and Innovation
- Critical Thinking and Problem Solving
- Communication and Collaboration

Information, Media and Technology Skills

- Information Literacy
- Information and Communications Technology (ICT)

Life and Career Skills

- Flexibility and Adaptability
- Initiative and Self-direction
- Social and Cross-Cultural Skills
- Productivity and Accountability
- Leadership and Responsibility

Computational Thinking Skills

- Decomposition
- Pattern Recognition
- Algorithm Design
- Abstraction

Robotics Concepts and Skills

- Motion Planning
 - Direction
 - Power/Speed
 - Distance
 - Duration
 - Brake
 - Arms
- Sensor-based Motion Planning
- Motors and Optical Encoders
- Sensors and Controllers
 - Touch - collision
 - Gyroscope - rotation and tilts (x, y, z)
 - Ultrasonic - distance
 - Light/Color - surface color/brightness