



Curriculum & Instruction

SUMMIT PUBLIC SCHOOLS

Content Area: Mathematics/Computer Science
Course: Robotics (H560)
Grade(s): 9-12

Course Overview

Science, Technology, Engineering and Mathematics (STEM) education provides students with the opportunity to participate in many themed hands-on activities, including those with connections to sports and games. Competitions involving robotics help students learn math, physics, and engineering by engaging in activities that are useful and enjoyable while providing them hands-on experiences in those key areas.

STEM Labs follow a sequence of learning experiences. The learner is asked to do the following:

- Create a build or an artifact
- Explore the build or the artifact and speculate as to its possible application in the real world
- Learn by doing
- Make changes to a design or a build to improve and enhance it
- Assess knowledge

1. Seek

Each engineering-focused STEM Lab begins with instructions for creating a working build or guidelines for an original design. If building from a set of given instructions, learners should be given ample time to follow the step-by-step instructions for creating the build whether individually or in small groups. After creating a design or build, learners are asked to test what it does. Learners are told to experiment with the build and answer various questions, such as what it does, how it might be used, what if any mechanical advantage it affords, and how to explain the build using engineering terms. Learners are directed to answer these questions within their engineering notebooks because the intent is to have their answers be available for review and feedback.

2. Play

The play section within a STEM Lab begins with a brief reading that provides a context for the concepts or skills within the activity. A short procedure may follow to explore the new skill or concept introduced. Most often, learners will return to testing their builds to recognize some feature of its design but there are many different types of activities that

they may complete. Some STEM Labs will focus on a single activity by having only one reading and one procedural activity, but most have more than one activity so that multiple concepts or skills can be introduced and more complicated concepts can be explored.

3. Apply

This section is where learners are provided with examples of how the concepts they are being introduced apply to their daily lives. They are also offered a look into the application of those skills in the competitive aspect of robotics and engineering.

4. Rethink

After learning about the concepts within the build, learners are given the opportunity to explore further with their build through a challenge. After creating the environment for the build, students may need to modify their build for success. Most of the challenges are competitive in nature and foster hands-on learning. Learners are directed to record and justify the changes they make in their engineering notebooks. Questions are posed to ask about both the design and testing phase.

5. Know

At the end of the STEM Lab, learners are asked questions about the concepts taught throughout the lab. Questions can be printed if they are to be completed by hand and turned in for a grade. Most questions are multiple-choice or true-false. Whether working in teams, groups, or classroom settings, the answers to the questions can be discussed to make sure that all learners recognize the correct answers and why they are correct.

Prerequisite: None

New Jersey State Learning Standards

Summit Public Schools curriculum aligns with the [New Jersey Student Learning Standards](#).

Computer Science and Design Thinking Standards:

- 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and
- identify trade-offs to justify the choice.
- 8.1.12.AP.4: Design and iteratively develop computational artifacts for
- practical intent, personal expression, or to address a societal issue by combining code snippets or remedies.
- 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
- 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
- 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

Date Revised: September 2026

- 8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
- 8.2.12.ED.1: Use and maintain technological products and systems using advanced fabrication tools (e.g., CAD software, 3D printing, CNC machining).
- 8.2.12.ED.2: Design and create a prototype to solve a real-world problem using a structured design process, identifying trade-offs and presenting for peer review.

NJSLS - Science: Engineering, Technology, and Applications of Science

- **NJSLS-S.HS-ETSI-1:** Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants. expectation directly related to climate, climate change, or the effects of a changing climate on environmental systems.
- **NJSLS-S.HS.ETSI-2:** Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- **NJSLS-S.HS.ETSI-3:** Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts. expectation directly related to climate, climate change, or the effects of a changing climate on environmental systems.
- **NJSLS-S.HS.ETSI-4:** Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem. expectation directly related to climate, climate change, or the effects of a changing climate on environmental systems.

NJSLS-S - Science: Physical Science

- **NJSLS-S: HS-PS2-1:** Analyze data to support the claim that Newton's second law of motion describes the mathematical relationship among the net force on a macroscopic object, its mass, and its acceleration.
- **NJSLS-S: HS-PS2-2:** Use mathematical representations to support the claim that the total momentum of a system of objects is conserved when there is no net force on the system.
- **NJSLS-S: HS-PS2-3:** Apply scientific and engineering ideas to design, evaluate, and refine a device that minimizes the force on a macroscopic object during a collision.
- **NJSLS-S: HS-PS3-3:** Design, build, and refine a device that works within given constraints to convert one form of energy into another form of energy.

Interdisciplinary Standards:

- **CCSS.6.RPA.1:** Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities.
- **CCSS.6.RPA.3.D:** Use ratio reasoning to convert measurement units; manipulate and transform units appropriately when multiplying or dividing quantities.
- **CCSS.ELA-LITERACY.L.9-10.6:** Acquire and use accurately general academic and domain-specific words and phrases, sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.

- **CCSS.ELA-LITERACY.SL.9-10.1:** Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grades 9-10 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
- **CCSS.ELA-LITERACY.SL.9-10.1.D:** Respond thoughtfully to diverse perspectives, summarize points of agreement and disagreement, and, when warranted, qualify or justify their own views and understanding and make new connections in light of the evidence and reasoning presented.
- **CCSS.ELA-LITERACY.SL.9-10.1B:** Work with peers to set rules for collegial discussions and decision-making (e.g., informal consensus, taking votes on key issues, presentation of alternate views), clear goals and deadlines, and individual roles as needed.
- **CCSS.ELA-LITERACY.SL.9-10.1C:** Propel conversations by posing and responding to questions that relate the current discussion to broader themes or larger ideas; actively incorporate others into the discussion; and clarify, verify, or challenge ideas and conclusions.
- **CCSS.ELA-LITERACY.SL.9-10.4:** Present information, findings, and supporting evidence clearly, concisely, and logically such that listeners can follow the line of reasoning and the organization, development, substance, and style are appropriate to purpose, audience, and task.
- **CCSS.ELA-LITERACY.SL.11-12.1:** Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grades 11-12 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
- **CCSS.ELA-LITERACY.SL.11-12.1B:** Work with peers to promote civil, democratic discussions and decision-making, set clear goals and deadlines, and establish individual roles as needed.
- **CCSS.ELA-LITERACY.SL.11-12.1C:** Propel conversations by posing and responding to questions that probe reasoning and evidence; ensure a hearing for a full range of positions on a topic or issue; clarify, verify, or challenge ideas and conclusions; and promote divergent and creative perspectives.
- **CCSS.ELA-LITERACY.SL.11-12.4:** Present information, findings, and supporting evidence, conveying a clear and distinct perspective, such that listeners can follow the line of reasoning, alternative or opposing perspectives are addressed, and the organization, development, substance, and style are appropriate to purpose, audience, and a range of formal and informal tasks.
- **RI.MF.9–10.6.** Analyze, integrate, and evaluate multiple interpretations (e.g., charts, graphs, diagrams, videos) of a single text or text/s presented in different formats (visually, quantitatively) as well as in words in order to address a question or solve a problem.
- **RST.9-10.7:** Translate quantitative or technical information expressed in words in a text into visual form (e.g., a table or chart) and translate information expressed visually or mathematically (e.g., in an equation) into words.
- **HSA.CED.A.1:** Create equations and inequalities in one variable and use them to solve problems.
- **HSN.Q.A.1:** Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs and data displays.
- **HSA.CED.A.4:** Rearrange formulas to highlight a quantity of interest, using the same reasoning as in solving equations.

- **HSG.SRT.C.8:** Use trigonometric ratios and the Pythagorean Theorem to solve right triangles in applied problems.
- **CCSS.MATH.PRACTICE.MP3:** Construct viable arguments and critique the reasoning of others.
- **CCSS.MATH.PRACTICE.MP5:** Use appropriate tools strategically.
- **CCSS.MATH.PRACTICE.MP6:** Attend to precision.
- **CCSS.MP.1:** Make sense of problems and persevere in solving them.
- **CCSS.MP.4:** Model with mathematics.
- **CCSS.MP.5:** Use appropriate tools strategically.
- **CCSS.MP.6/CCSS.MP6:** Attend to precision.
- **CCSS.RST.9-10.2/CCSS.RST.9-10.2:** Determine the central ideas or conclusions of a text; trace the text's explanation or depiction of a complex process, phenomenon, or concept; provide an accurate summary of the text.
- **CCSS.RST.9-10.3/CCSS.ELA-LITERACY.RST.9-10.3:** Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
- **CCSS.RST.11-12.3:** Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
- **CSTA.1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **CSTA.1B-IC-18:** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.
- **CSTA.2-AP-10:** Use flowcharts and/or pseudocode to address complex problems as algorithms.
- **CSTA.2-AP-12:** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
- **CSTA.2-AP-13:** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- **CSTA.2-AP-19:** Document programs in order to make them easier to follow, test, and debug.
- **CSTA.3A-AP-13:** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- **CSTA.3A-AP-16:** Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
- **CSTA.3A-AP-17:** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- **CSTA.3A-AP-21:** Evaluate and refine computational artifacts to make them more usable and accessible.
- **CSTA.3A-AP-22:** Design and develop computational artifacts working in team roles using collaborative tools.
- **CSTA.3A-AP-23:** Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
- **CSTA.3A-IC-24:** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- **CSTA.3B-AP-08:** Describe how artificial intelligence drives many software and physical systems.
- **CSTA.MP.5:** Use appropriate tools strategically.
- **CSTA.MP.6:** Attend to precision.

- **CSTA.RST.9-10.3:** Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
- **CSTA.RST.11-12.3:** Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
- **CSTA.RST.11-12.9:** Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.

Career Readiness, Life Literacies and Key Skills Standards:

- **9.4.8.CI.3:** Examine challenges that may exist in the adoption of new ideas
- **9.4.8.CT.2:** Develop multiple solutions to a problem and evaluate short- and long-term effects to determine the most plausible option
- **9.4.8.DC.7:** Collaborate within a digital community to create a digital artifact using strategies such as crowdsourcing or digital surveys.
- **9.4.12.CI.1:** Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
- **9.4.12.CT.1:** Identify problem-solving strategies used in the development of an innovative product or practice
- **9.4.12.CT.2:** Explain the potential benefits of collaborating to enhance critical thinking and problem solving
- **9.4.12.CT.4:** Participate in online strategy and planning sessions for course-based, school-based, or other projects and determine the strategies that contribute to effective outcomes.
- **9.4.12.TL.4:** Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Career Readiness Practices:

- **CRP 1.** Act as a responsible and contributing citizen and employee.
- **CRP 2.** Apply appropriate academic and technical skills.
- **CRP 3.** Attend to personal health and financial well-being.
- **CRP 4.** Communicate clearly and effectively and with reason.
- **CRP 5.** Consider the environmental, social and economic impacts of decisions.
- **CRP 6.** Demonstrate creativity and innovation.
- **CRP 7.** Employ valid and reliable research strategies.
- **CRP 8.** Utilize critical thinking to make sense of problems and persevere in solving them.
- **CRP 9.** Model integrity, ethical leadership and effective management.
- **CRP 10.** Plan education and career paths aligned to personal goals.
- **CRP 11.** Use technology to enhance productivity.
- **CRP 12.** Work productively in teams while using cultural global competence.

New Jersey Curriculum Mandates:

This curriculum integrates lessons that address the following interdisciplinary New Jersey Curriculum Mandates:

- [Diversity and Inclusion](#) (Grades K - 12)
 - C.18A:35-4.36a: Curriculum to include instruction on diversity and inclusion.
 - (3) encourage safe, welcoming, and inclusive environments for all students regardless of race or ethnicity, sexual and gender identities, mental and physical disabilities, and religious beliefs.

Learning Objectives

Learning objectives are clear, specific statements that define what students should know, understand, and be able to do by the end of a lesson or course.

Students will be able to...

- Plan a race course using scaled drawings.
- Analyze relationships between scaled measurements.
- Convert units of measurement across different scales.
- Apply proportional reasoning and scale factors to solve engineering problems.
- Adapt the Speedbot design for specialized tasks.
- Integrate a Bumper Switch to detect physical obstacles.
- Monitor real-time robot updates and telemetry using the Brain Screen.
- Interpret game rules to form valid competition strategies.
- Modify the Speedbot mechanism to serve medical field applications.
- Program the Speedbot to execute custom driving patterns.
- Automate robot actions through custom code structures.
- Explain how moving robots transfer kinetic energy to obstacles upon impact.
- Calculate how mass and velocity change an object's overall momentum.
- Construct a passive plow mechanism to push field objects.
- Program robots to emulate creative expression and artistic processes.
- Document engineering workflows, reflections, and design iterations in a standardized notebook.
- Execute precise motion sequences to ensure accurate multi-step robot travel.
- Explain the function of the single-capture Vision Sensor snapshot block.
- Implement continuous background execution using forever loops.
- Calibrate and tune hardware components for optimal performance.
- Design specialized end effectors for handling hazardous or delicate items.
- Configure motor timeouts to prevent mechanical strain within limited motion ranges.
- Evaluate the efficiency and accuracy gains of replacing human warehouse labor with autonomous systems.
- Structure controller loops to continuously poll button and joystick inputs.
- Define how loop structures continuously repeat robotic behaviors.
- Compare the repetitive execution consistency of robots against human operation.
- Differentiate fully autonomous code execution from driver-controlled modes.
- Decompose complex autonomous paths into small, codeable behaviors.
- Utilize Distance and Optical sensors to measure object proximity.
- Develop precise algorithms to solve specific operational problems.
- Construct algorithms using sequence, selection, and loop structures.
- Trace code block execution order within a continuous loop structure.
- Evaluate boolean conditions in if-then-else blocks to control execution flow.

- Implement conditional logic to make program decisions based on dynamic boolean inputs.
- Apply conditional branching to route code execution paths based on environmental state.
- Define User Experience (UX) relative to user interface (UI) efficacy.
- Analyze user feedback data to optimize UI and improve overall user experience.
- Predict the execution path of continuous forever loops.
- Configure count-controlled repeat loops to execute code for a specified number of iterations.
- Map controller inputs to trigger event-driven programming functions.
- Apply motor "hold" settings to maintain arm and claw positioning under load.
- Assess robot design features to leverage mechanical advantages while accommodating physical limitations.
- Incorporate peer and design feedback to iterate on mechanical solutions.
- Apply competitive sports strategies to solve complex robotic engineering challenges.
- Design specialized manipulators to interact with and move field elements.
- Distinguish between motor-powered active manipulators and unpowered passive manipulators.
- Integrate passive mechanisms to manipulate objects without using extra motors.
- Configure dual-motor groups to function as unified mechanical drives.
- Select appropriate claw configurations to securely grip and transport diverse shapes.
- Conduct strategic scouting of opponent teams to make data-driven strategy decisions.
- Compare the mechanical movement of robotic arms to human biomechanics.
- Analyze empirical performance data to drive design and tactical decisions.
- Program autonomous routines to navigate fields without driver intervention.
- Balance manual driver control speed against the precision of programmed movements.
- Formulate path-planning maps to structure autonomous software routines.
- Deploy Optical sensors to automate visual target identification and sorting.
- Combine conditional statements with sensor loops for responsive feedback control.
- Collaborate within a team to systematically test and refine competitive strategies.
- Predict center-of-gravity shifts between static rigid structures and multi-jointed mechanisms.
- Evaluate dynamic robot stability based on speed, acceleration, and center-of-gravity position.
- Mitigate driver transition challenges by standardizing joystick layouts and control sensitivity.
- Design gear trains to optimize either output speed or mechanical advantage (torque).
- Calculate total system gear ratios across multi-gear chains.
- Determine input and output gear configurations required to produce high torque vs. high speed.
- Draft a comprehensive Request for Proposal (RFP) outlining problem constraints and project scope.
- Deconstruct the structural elements of an RFP, including executive summaries and scope criteria.
- Outline detailed project timelines, organizational rubrics, and cost estimates within an RFP.
- Construct a friction-based pinching claw to grasp game objects.
- Deploy a passive side-sweeping plow mechanism to direct objects during travel.
- Design an under-object scoop mechanism to lift and transport payload.
- Apply scientific principles and iterative design methods to solve complex engineering problems.

- Select appropriate active or passive manipulators based on operational constraints.
- Integrate vertical lift mechanisms with end effectors to manipulate elevated objects.
- Refine strategy through iterative testing and collaborative problem-solving.
- Manage project scope under tight deadlines using structured team planning.
- Program the AI Vision sensor to identify, track, and categorize dynamic field objects.
- Demonstrate persistence and productive problem-solving through iterative engineering failures.

Units of Study

Units of study are organized collections of lessons and learning experiences focused on a specific topic and set of skills within a course. Each unit is guided by a set of essential questions.

Lab 1: Robo Rally [~ 1 week]

- How do mechanical trade-offs between speed and power affect a robot's overall performance?
- How do sensors and closed-loop feedback make a robot's movement predictable and repeatable?
- What strategies allow programmers to efficiently isolate, debug, and fix unexpected robot behaviors?
- How do engineers use test data to balance competing constraints when refining a design?

Lab 2: Team Freeze Tag [~2 weeks]

- How do mechanical choices (such as wheel type or gear selection) impact a robot's agility, speed, and maneuverability during active gameplay?
- How can programmers use sensor logic and conditional statements to create automated responses (like freezing when a bumper switch is triggered)?
- How does custom controller mapping affect a driver's reaction time and efficiency in high-pressure scenarios?
- How do teams analyze field spatial awareness and game constraints to develop effective offensive and defensive strategies?

Lab 3: Medbot [~2 weeks]

- How do clear design constraints and requirements guide engineers when developing robots for specialized environments like healthcare?
- How can sensors be programmed to help a robot navigate complex pathways and reliably deliver payloads?
- In what ways do mechanical mechanisms (such as manipulators or claws) ensure precise object handling without dropping or damaging materials?
- How do engineers analyze test results and system failures to make a robotic delivery system more accurate and repeatable?

Lab 4: Momentum Alley [~1 week]

- How do forces and mechanical energy transfer affect the motion and impact of a robotic bowling arm?

- How can programmers use variable control and sensor timing to achieve precise, repeatable launches?
- In what ways does physical weight distribution and structure placement alter momentum and stability during an automated action?
- How do engineers collect and analyze trial data to make calculated adjustments to a robot's power and trajectory?

Lab 5: It's a Draw! [~1 week]

- How do mathematical coordinates and geometric paths translate into physical motor movements for a drawing robot?
- How can precise motor control and sensor feedback ensure consistency and accuracy when executing detailed patterns?
- In what ways do pen attachment designs and downforce pressures affect line quality and drawing precision?
- How do programmers break down complex shapes and designs into sequential algorithmic steps?

Lab 6: Vision Sensor [~1 week]

- How do computer vision sensors capture and process optical data to help a robot recognize colors and objects?
- How do programmers use object detection data and conditional statements to trigger specific autonomous robot actions?
- In what ways do environmental factors (such as ambient lighting and surface glare) affect a vision sensor's accuracy and reliability?
- How do engineers calibrate physical camera positioning and sensor signatures to ensure precise object tracking?

Lab 7: Speedy Delivery [~2 weeks]

- How do drivetrain modifications and gear ratios influence a delivery robot's speed without sacrificing steering control?
- How can programmed sensors be used to navigate complex layouts and detect obstacles along a delivery route?
- In what ways do physical container designs and cargo placement impact a robot's balance and cargo security during rapid transit?
- How do engineers evaluate trade-offs between delivery speed, accuracy, and payload safety when optimizing an automated system?

Lab 8: Loop, There it Is! [~2 weeks]

- How do repeat loops and conditional logic simplify complex, repetitive programming sequences for an autonomous robot?
- How can sensor feedback be integrated with loop structures to continuously monitor and respond to changing surroundings?
- In what ways do modular code structures make debugging and optimizing a robot's path execution more efficient?
- How do engineers determine when to use fixed-count loops versus condition-based loops to achieve consistent performance?

Lab 9: Castle Crasher [~2 weeks]

- How do mechanical advantages, such as torque and leverage, help a robot generate enough force to clear heavy obstacles?
- How can distance sensors and optical feedback be programmed to detect, track, and clear target objects off a field?
- In what ways do structural design and center of mass impact a robot's stability when pushing or lifting objects?
- How do engineers use iterative testing to refine autonomous strategies for maximizing field clearance within a time limit?

Lab 10: To Do, or Not to Do [~2 weeks]

- How do decision-making control structures (such as if-then-else statements) allow a robot to evaluate dynamic environmental conditions?
- How can multiple sensor inputs be prioritized within a program to prevent conflicting robotic actions?
- In what ways does physical sensor placement alter the timing and accuracy of conditional triggers?
- How do programmers systematically test edge cases to ensure conditional logic executes reliably under varied circumstances?

Lab 11: Clawbot with Controller [~1 week]

- How does human-machine interface design influence a driver's control accuracy and responsiveness during complex robot maneuvers?
- How can custom controller button mapping and joystick axis scaling be programmed to optimize precise mechanical movements?
- In what ways do mechanical mechanisms (such as arm joints and claw grippers) translate driver inputs into controlled physical force?
- How do teams evaluate real-world driver feedback to refine code and mechanical configurations for peak operational efficiency?

Lab 12: Robosoccer [~2 weeks]

- How do mechanical drivetrain designs and wheel configurations influence a robot's speed, traction, and agility during fast-paced field play?
- How can custom controller configurations and event-driven code be optimized to improve driver reaction time and ball handling?
- In what ways do physical manipulators or intake mechanisms maximize control over a moving object without violating game rules?
- How do teams analyze field dynamics and physical trade-offs to develop effective offensive, defensive, and cooperative match strategies?

Lab 13: Robot Soccer [~2 weeks]

- How do mechanical drivetrain choices and wheel selections impact a robot's agility, speed, and ball control on a soccer field?
- How can custom controller configurations be programmed to optimize driver responsiveness and precise manipulator actions?
- In what ways do physical ball-handling mechanisms ensure continuous control while protecting against mechanical jamming or defense?

- How do teams analyze field spatial awareness and opponent tactics to execute coordinated offensive and defensive strategies?

Lab 14: Up and Over [~3 weeks]

- How do mechanical advantages, such as gear ratios and linkages, allow a robotic arm or lift to lift objects efficiently?
- How can motor encoders and sensor limits be programmed to control precise vertical movement and position holding?
- In what ways do center of mass, structural bracing, and counterweights maintain robot stability when lifting high payloads?
- How do engineers evaluate trade-offs between lifting speed, motor strain, and structural stability when refining an elevation system?

Lab 15: Ring Leader [~2 weeks]

- How do mechanical manipulators and intake designs affect a robot's ability to reliably align, scoop, and secure ring-shaped objects?
- How can sensors and motor encoders be programmed to achieve precise arm positioning and automated ring scoring routines?
- In what ways do object geometry, center of gravity, and friction influence how rings travel through a mechanical delivery system?
- How do teams analyze test data to optimize the trade-offs between ring intake speed, scoring accuracy, and mechanism reliability?

Lab 16: Treasure Hunt [~2 weeks]

- How do optical and color sensors process visual feedback to help a robot distinguish target objects from field obstacles?
- How can condition-based logic and continuous sensor monitoring be programmed to execute autonomous searching and retrieving routines?
- In what ways do collector and gripper mechanisms adapt to variation in object shape, orientation, and surface friction?
- How do engineers refine navigation paths and scanning algorithms to minimize time spent locating objects in an unstructured field?

Lab 17: Gravity Rush [~1 week]

- How do gravitational forces and ramp angles influence a robot's speed, acceleration, and energy consumption?
- How can sensors and motor encoders be programmed to control braking and maintain stability when navigating steep inclines?
- In what ways do wheel traction, weight distribution, and chassis design prevent slipping or tipping under severe gravitational loads?
- How do engineers analyze energy loss from friction and gravity to optimize battery usage and mechanical performance on slopes?

Lab 18: Mechanical Advantage [~1 week]

- How do different gear configurations and mechanical advantages alter the balance between output speed and rotational torque?

- How can programmers adjust motor power outputs to complement physical gear reductions and prevent mechanical straining?
- In what ways do compound gear trains and lever arms amplify mechanical force when lifting heavy structural loads?
- How do engineers collect mechanical performance data to evaluate power transfer efficiency and reduce system friction?

Lab 19: Design by Request [~2 weeks]

- How do engineering design requirements and client specifications shape the initial criteria and constraints for a robotic solution?
- How can motor control programs be structured to satisfy complex functional demands while maintaining code clarity and safety?
- In what ways do modular physical builds allow teams to rapidly prototype, test, and iterate mechanisms against project goals?
- How do engineers document and present design trade-offs to justify their technical choices during design reviews?

Lab 20: Platform Placer [~3 weeks]

- How do different robotic lift designs (such as scissor lifts or linear slides) provide the vertical reach and stability needed to place objects onto elevated platforms?
- How can motor encoders and feedback loops be programmed to stop a lift mechanism at precise heights for accurate object placement?
- In what ways do the center of gravity and leverage shift as a robot raises a payload to an elevated platform, and how can physical design counteract tipping?
- How do teams evaluate strategy and driver controls to balance speed, precision, and stability during platform scoring challenges?

Lab 21: Clean Water Mission [~6 weeks]

- How can AI Vision Sensors or optical sensors be used to distinguish between clean water elements and pollutants on the field?
- How do programmers combine object detection data with autonomous navigation loops to efficiently collect and sort field game pieces?
- In what ways do custom intake mechanisms and conveyer systems securely transport water game elements without dropping them?
- How do teams evaluate mission constraints and scoring zones to optimize path planning and maximize game point strategy within time limits?

Instructional Resources and Materials

Instructional resources are the information sources teachers utilize to guide their teaching and assessing. Instructional materials are the texts students read or review. The instructional resources and materials may vary in order to meet the learning needs of students.

Recommended:

- V5 Vex Robotics Curriculum Pacing Guide
- Vex Robotics YouTube Channel

Date Revised: September 2026

- Vex for Education

Teacher Resources:

- VEX Teacher Resources
- V5 Vex Robotics Curriculum Pacing Guide

Assessment Strategies

Assessment measures a student's mastery of a course standard and learning objective. Assessment can be used for both instructional purposes and for evaluative purposes. Teachers may choose one or more of the following ideas to assess learners. They may also modify these ideas to make the assessment more responsive to students' interests and needs.

Formative Assessments:

- Engineering Notebook Rubric
- Entrance/Exit Tickets
- Short Quizzes

Summative Assessments:

- Rubric Assessment on STEM Lab

Benchmark Assessments:

- Pre-Assessment
- Post Assessment

Alternative Assessments:

- Student Reflection
- Student Goal Setting

Suggested Strategies for Modification & Differentiation

[Modifications Chart](#)