	Course Title: Robotics **Course Description** Robotics is a course designed to introduce students to the construction and programming of simple robots using Lego Mindstorms. Students will learn to program a robot to operate autonomously using LabVIEW for Lego Mindstorms. To accomplish this task this course teaches students how to construct flow diagrams that break complex tasks into smaller and simpler tasks; basic Boolean Algebra and Logic to control data from multiple data sources simultaneously; and to design executable programs that complete a series of increasingly difficult tasks. Students will be introduced into the basic function and algorithm design as well as how to construct readable programs. The culmination of the programming challenges leads students to an elimination style competition with robots designed by other teams. **Length of Course 1 semester**	
Essential Questions	• How can an autonomously programed robot be designed to perform specific tasks using a variety of sensors that acquire information about the world external to the robot? • How can autonomous robots be designed and used to perform manual and repetitive tasks safely? In the workforce? In the home? And is it good or bad that these automated machines are replacing human labor? • Do robots have an important place in our world? Or will they one day take over like in apocalyptic movies?	
Standards	• No state defined standard exists for this course.	
Concepts and Skills	**Programming**: Students will develop programming skills by modifying simple programs that run autonomously on a robot referred to as an NXT. Programs will become increasingly complex and students will progress from one programs to the next only through the mastery of each program in sequential order. The overarching goal of each student is to create programs that are capable of a high level of autonomous control over the NXT using data sensors that collect data on the environment external to the NXT. **Practices** •Ask questions that arise from careful observations of phenomena. •Use and develop multiple types of models to represent and support explanations of phenomena and solve problems. (teach) •Design an investigation individually and collaboratively and test designs to support explanations for phenomena, or test solutions to problems and refine the design accordingly. (Teach) •Conduct an investigation individually and collaboratively, taking accurate data to produce reliable measurements and consider limitations on the precision of the data (teach) •Select appropriate tools to collect, record, analyze, and evaluate data. (teach) •Manipulate dependent and independent variables and collect data about a proposed process or system. (teach) •Apply concepts of statistics and probability (including determining function fits to data, slope, intercept, and correlation coefficient for	**CCR Reading Standards** 1. Cite specific textual evidence to support analysis of science and technical texts, attending to the precise details of explanations or descriptions. 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. 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. 4. Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to *grades 9–10 texts and topics*. 5. Analyze the structure of the relationships among concepts in a text, including relationships among key terms Analyze the author’s purpose in providing an explanation, describing a procedure, or discussing an experiment in a text, defining the question the author seeks to address. 6. 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. 7. Read and comprehend science/technical texts in the grades 9–10 text complexity band independently and proficiently. **CCR Writing Standards**

	linear fits) to analyze data. (teach) •Compare and contrast various types of data sets to examine consistency of measurements and observations. (reteach) •Analyze data to identify design features or characteristics of the components of a proposed process or system to optimize it relative to criteria for success. (teach) •Use mathematical or algorithmic representations of phenomena or design solutions to describe explanations and create computational models or simulations. (reteach) •Apply techniques of algebra and functions to represent and solve scientific and engineering problems. (reteach) •Make quantitative and qualitative claims regarding the relationship between dependent and independent variables. (reteach) •Engage in arguments using scientific and empirical evidence from investigations. •Engage in argument to critique solutions proposed by peers by citing relevant evidence •Redefine argument based on evidence from multiple sources (peers, text, etc.) •Accountable talk strategies; turn-and-talk; think-write-pair-share •Apply scientific knowledge, reasoning, and empirical evidence from investigations to support claims, explain phenomena, and solve problems (reteach) •Design, evaluate, and refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence. (reteach) •Critically read scientific literature adapted for classroom use to determine the central ideas or conclusions of a text; summarize complex concepts, or processes by paraphrasing them in simpler but still accurate terms. (reteach) •Synthesize, communicate, and evaluate the validity and reliability of claims, methods, and designs that appear in scientific and technical texts or media reports, verifying the data when possible. (reteach) •Produce scientific and/or technical writing and/or oral presentations that communicate scientific ideas and/or design and performance of a process •Compare, integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) in order to address a scientific question or solve a problem.	1. Write arguments focused on *discipline-specific content*. 2. Write informative/explanatory texts, including scientific procedures/ experiments, or technical processes. 3. Write precise enough descriptions of the step-by-step procedures they use in their investigations or technical work that others can replicate them and (possibly) reach the same results. 4. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience. 5. Use technology, including the Internet, to produce, publish, and update individual or shared writing products, taking advantage of technology's capacity to link to other information and to display information flexibly and dynamically. 6. Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation. 7. Draw evidence from informational texts to support analysis, reflection, and research.
Content Objective	• Identify and explain the steps of the engineering design process: identify the problem, develop possible solutions, select the best possible solution(s), construct prototypes, test and evaluate prototypes, and redesign if necessary. • Understand that the engineering design process is used in the solution of problems and the advancement of society. Identify examples of technologies, objects, and processes that have been modified to advance society, and explain why and how they were modified.	

	• Develop flow diagrams that identify the logical steps in a program operation, before using a computer to program. • Develop computational algorithms that solve specific problems. • Use a variety of sensors (ultrasonic, sound, touch, and light) to interpret data from the world external to the robot, with the ability of that data to modify the robots behavior. Students should also understand the limitations of sensors and how they acquire information from the external world. • Modify previously existing programs to perform increasingly complex challenges and operations without reinventing the wheel. • Use logic gates and operations (Boolean Algebra) to develop hierarchical program flow and command. • Develop an understanding of how gears (and gear ratios, and gear trains) can be used to increase motor power or speed and the tradeoffs between using one over the other. • Identify basic physical limitations to robot design (center of mass and torque).
Assessments/ Products	**Investigations:** Programming Lesson 1: **Drive forward two feet and stop, and reverse two feet** – Your First LabVIEW program. Programming Challenge 2: **Drive forward two feet and stop, and reverse two feet** Programming Challenge 3: **Turning the NXT** – types of turns. Programming Challenge 4: **Driver forward two feet, stop for one second, reverse power turn, stop for one second, drive forward to the starting point and stop** Programming Lesson 5: **While Loops and Boolean Values** – an Introduction to While Loops and repeating code. Programming Challenge 6: **Stopping on a Black Line** Programming Challenge 7: **Touch-Touch-Stop** Programming Challenge 8: **Drive forward two feet, reverse power turn inside of a While Loop, return to starting point**. Programming Lesson 9: **Case Structures –**Making Choices about Sensor Input Programming Challenge 10: **Looking for Silver** –Adding Cases to Case Structures Programming Challenge 11: **Waiting for a Clap AND Looking for Silver** –Using Or Gates to Stop a While Loop Programming Challenge 12: **Navigating a Maze** Programming Challenge 13: **Wait for a Clap to Begin** – Introducing Program Order Programming Challenge 14: **Stop When Close, Stay in a Box, and Stop for Loud Noise –** Precedence Programming Challenge 15: **Making it Across a Black Line** - Stringing Loops Together and eliminating wait functions. Programming Challenge 16: **Wait before Driving Across a Black Line** Programming Challenge 17: **Crossing a Black Line but Stop anywhere for a Clap** Programming Challenge 18: **Reverse Power Turn from one line to the Next** Programming Lesson 19: **Stay Back** – An Introduction to Proportionality Control Programming Challenge 20: **The Line Follower** Programming Challenge 21: **Follow Line Stop if Close Reverse Follow Line** Programming Challenge 22: **Wait to Follow a Line** Programming Challenge 23: **Stay in Box without Reversing Out –** Final Challenge before Sumo Bot Challenge Programming Challenge 24: **Sumo Bot Competition** **Student self-assess** Student print outs of each program including written summaries explaining the purpose of the program and explanations of each subprogram within the overall program. **Notebooks:**

	➢ **Content Notes (every day or close to it):** Students will identify topics; identify the main ideas and most important details and examples associated with each topic; include summaries as well as student-generated follow-up questions and answers, reflections, visualizations, and responses to the content, using higher order thinking skills (e.g., predict, connect, infer, analyze, evaluate, categorize, synthesize). ➢ **Vocabulary**: Students will highlight additional, key vocabulary in their notebooks; they will build an understanding of the vocabulary using vocabulary-development exercises (e.g., word webs, Frayer Model), as well as use the vocabulary in their daily work and conversations. ➢ **Narrative and Explanatory Essay (in response to one or more Essential and Guiding Questions)/Investigation Reports:** Student work will include evidence of planning: graphic organizers, brainstorming lists; editing of language, vocabulary, grammar, structure; organized and developed ideas utilizing precise and domain specific language; student sharing, student and teacher feedback, and revisions based on these conversations. Argumentative essays/investigation reports will include an explicit claim, scientific evidence in support of the claim (from reports, data, observations, etc.), and an explanation of how the evidence connects to and verifies the claim. ➢ **Other Sample Products:** KWL Charts. Venn Diagrams, Concept Maps, H.O.T. Boxes, Others?
Textbooks, Materials, and Resources	Teacher generated Programming Lessons and Challenges.