ft)			
AA batteries (100 pcs)	https://www.amazon.com/dp/B004SCA15K/_encoding=UTF8?coliid=IMHHSED3OLLFK&colid=3954J1GQRI8GD&th=1	$ 19.08	$ 0.3816
Resistors, 51 Ohm (100 pcs)	https://www.amazon.com/dp/B0185FHLH6/_encoding=UTF8?coliid=I3SZQFOCT8CAMC&colid=3954J1GQRI8GD	$ 6.16	$ 0.1232
OPTIONAL Phototransistors (20 pcs)	https://www.amazon.com/AMBIENT-Ambient-Sensors-IR-BLOCKING-Phototransistor/dp/B01K563B22/ref=sr_1_1?ie=UTF8&qid=1499351089&sr=8-1&keywords=phototransistor	$ 7.99	
		Approximate cost per bot	$5.42

Distribution A.

4.3.2. CHALLENGE PROBLEM: PSEUDOCODE & FLOWCHARTS

4.3.2.1. OBJECTIVES

- Understand and apply how pseudocode and flowcharting can be used for planning code or programming.

4.3.2.2. KEY VOCABULARY

- Pseudocode
- Flow chart
- Autonomous
- Basic behaviors
- Simple behaviors
- Complex behaviors

4.3.2.3. PRELIMINARY INVESTIGATION

- Introduction to Programming: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/index_fundamentals.html
  - Section 1. Programmer and Machine
  - Section 2. Planning and Behaviors
    - Reference sheets at the bottom of the page
      - Behaviors
      - Pseudocode & Flowcharts
      - Thinking about Programming
    - Additional reference videos
      - Iterative Design
      - Flowcharts
  - Programmer and Machine video: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/1-1IntroductiontoProgramming1.html
  - Planning and Behaviors: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/1-1IntroductiontoProgramming2.html
- Additional References
  - From Illinois Institute of Technology Computer Science: http://www.cs.iit.edu/~cs100/ProblemSolving.pdf

4.3.2.4. CHALLENGE

- Create a flow chart and pseudocode for an autonomous robot that avoids walls in a maze.
  - Use the Autobot from the last challenge as a starting point then extend to coding.

4.3.2.5. EXTENSIONS

- Use flowchart software
    - Examples of free or free-trial software:
        - https://www.draw.io/
        - https://raptorflowchart.en.softonic.com/
        - https://cacoo.com/
- If students have coding experience, they can write the code or they can work backwards—write the code, pseudocode, and flow chart.
- Engineer-defined—discuss with teacher.

4.3.3. CHALLENGE PROBLEM: LABYRINTH

4.3.3.1. OBJECTIVES

- Modify motor commands to adjust movement speed, direction, power, etc.
- Understand and use logic to develop simple robotic programs.
- Become proficient in ROBOTC or other programming language and syntax.
- Properly add comments in coding.
- Understand how encoders function and use encoders to better control speed and position.

4.3.3.2. KEY VOCABULARY

- Swing turn
- Point turn
- Wait state
- Shaft encoder
- While loop
- Boolean Logic
- if/else statements
- Variable
- Global variable
- Debug
- Proportional-Integral-Derivative (PID) controller
- Motor Controller

4.3.3.3. PRELIMINARY INVESTIGATION

Record all notes, drawings, example calculations, coding notes, and references in your notebook.
Iterate as often as needed.
Verify final investigation runs with instructor.
***Save all programs—several will be used and modified in later investigations.**

- SparkFun Rotary Encoder tutorial: https://www.sparkfun.com/news/2438
- For reference: videos and references in the MOVEMENT module of CMU Curriculum: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/index_movement.html
    - Sumo Bot: Rotational Torque vs. Power Level Investigation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/sumo_bot.pdf
        - Adapt to use force sensors instead of cans to quantify force
    - Wait States Investigation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/wait_states_power_level.pdf
        - Will establish relationship between motor wait time and distance traveled.
        - Graph results to determine relationship between time and distance to better predict distance.

    - Is it linear? What is the equation of the line?
- Simulated Acceleration Investigation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/simulated_acceleration.pdf
  - Program robot to accelerate under controlled parameters
- Power Levels Investigation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/power_levels_investigation.pdf
  - Will establish relationship between motor power levels and distance traveled.
- Turning Investigation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/turning.pdf
  - Determine the necessary calculations to make turns of various angles.
- Sentry Simulation: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/sentry_one.pdf
  - Proof of concept for autonomous travel
- Driving Straight: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/driving_straight.pdf
  - Tests whether robot drives straight—may need to adjust motor power levels due to robot weight distribution, friction, etc.
- Basketball Drill: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/basketball_drills.pdf
  - Challenge to practice coding with encoders
- Power Level with Encoders: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/encoder_powerlevel_investigation.pdf
  - Normalizing robot movement with the encoders, determines whether the relationship between power and speed is proportional
- Turning with Encoders: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/turning_w_encoders.pdf
  - Complete more accurate and consistent turns using encoders

- Driving Straight with Encoders: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/driving_straight_2.pdf
    - Use encoders to maintain a straight path
- Seeing the Difference: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/seeing_difference.pdf
    - Determine whether motors at the same power turn differently.
- Robot Acceleration/Deceleration: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/robot_acceleration.pdf
    - Programming challenge to write a program in which the robot accelerates constantly.
- CHOOSE ONE
    - Basketball Drills with Encoders: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/basketball_drills.pdf
        - Use encoders and PID to more accurately run the Basketball Drill
    - Sentry Simulation with Encoders: http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/sentry_one.pdf
        - Use encoders and PID to more accurately run the Sentry Simulation

4.3.3.4. CHALLENGE

- Design, build, and program a robot that can autonomously navigate through a maze from a starting point to a goal area using movement behaviors.

4.3.3.5. EXTENSIONS

- Use the encoders to collect data on distance to establish programming conditions.
- To accentuate the role of encoders, collect data with and without encoders driving on an uneven dirt surface.
- Require more precise moves, goal area, etc.
- Use Functions, Parameters, or other advanced coding to optimize programming.
- Use robots other than VEX (e.g. robot chassis with Arduino microcontroller and other components).
- Engineer-defined—discuss with teacher.

4.3.3.6. TEACHER NOTES

- As much or as little of the Preliminary Investigation can be used depending on the depth of understanding desired (i.e. a simple autonomous robot can be built just from iterative programming. The investigations will afford a more precisely controlled autonomous robot.)
- Maze can be created from a variety of materials: tape, PVC, 2x4s, etc.
- The maze should remain setup during the course of the challenge or be easily setup during each class.

4.3.4. CHALLENGE PROBLEM: SENSORS

4.3.4.1. OBJECTIVES

- Identify and compare sensors and applications.
- Develop challenges and solutions to demonstrate knowledge and application of sensors.

4.3.4.2. KEY VOCABULARY

- Proprioceptive
- Exteroceptive
- Active sensor
- Passive sensor
- Sensor
- Digital
- Analog
- Dynamic
- Resolution
- Sensitivity
- Error
- Calibration

4.3.4.3. PRELIMINARY INVESTIGATION

Record all notes, drawings, example calculations, coding notes, and references in your notebook.

Iterate as often as needed.

- For about 5-10 minutes, have students identify types of sensors on the robots in the *Robot Catalog* or from online research.
- Group activity: on whiteboard, have students list and categorize types of sensors.
- In team, research a specific category of sensors (e.g. position) and present findings to group, including:
    - Types of sensors
    - Classifications
        - (e.g. but not limited to: proprioceptive/exteroceptive; active/passive; digital/analog)
    - How the sensors work
    - If applicable: comparison of sensors
        - E.g. why use an ultrasonic vs PIR sensor
    - Examples of authentic applications

- Videos and resources contain within the Sensing module: http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/

4.3.4.4. CHALLENGE

- Develop challenge(s) and solution(s) to demonstrate knowledge and application of sensors.

4.3.4.5. REQUIREMENTS & CONSTRAINTS

- Mastery of at least three different sensors must be demonstrated.
- May develop one or multiple challenges to demonstrate mastery.
- Code must be properly commented explaining parameters of sensors.
- Challenges with authentic application are encouraged.

4.3.4.6. TEACHER NOTES

To save time or for a more directed learning experience, a combination of VEX Challenges can be implemented instead of student-designed challenges (these can also be used with non-Vex robotics—the sensor utilized is in parentheses):

- Wall Follower (limit switch; could also be adapted for other sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/operator_assist/docs/wall_follower.pdf
- Robotic Mouse (limit switch and bumper switch; could also be adapted for other sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/operator_assist/docs/robotic_mouse.pdf
- Robocci (limit switch): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/operator_assist/docs/robocci.pdf
- Quick Tap (bumper or limit switch; could also be adapted for other sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/operator_assist/docs/quick_tap.pdf
- Counter (limit switch or bumper switch; could also be adapted for other sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/operator_assist/docs/add_sub.pdf
- Table Bot (a number of sensors: ultrasonic, light sensors, etc.): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/forward_till_near/docs/table_bot.pdf
- Speed of Sound (ultrasonic sensor; could be adapted for light sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/forward_till_near/docs/speed_sound.pdf
- Sonic Scanner(ultrasonic sensor; could be adapted for light sensors): http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vex/sensing/forward_till_near/docs/sonic_scanner.pdf

4.4. CAPSTONE PROJECT

4.4.1. CHALLENGE PROBLEM: CAPSTONE

4.4.1.1. OBJECTIVES

- Utilize knowledge and skills to identify a problem and develop a solution using robotics.
- Develop creative and innovative solutions, using failure as an opportunity to learn.
- Use critical thinking and problem solving to analyze and synthesize information and solutions; troubleshoot problems that arise.
- Demonstrate the ability to effectively collaborate with a team, exercising flexibility and shared responsibility.
- Effectively communicate ideas, procedures, reflection, results, etc. through a variety of modalities—oral, written, nonverbal, etc.
- Develop and practice executive functioning skills.

4.4.1.2. KEY VOCABULARY

- Systems design

4.4.1.3. PRELIMINARY INVESTIGATION

Record all notes, drawings, example calculations, coding notes, and references in your notebook.

Iterate as often as needed.

- Self-directed

4.4.1.4. CHALLENGE

- In teams or individually, identify a problem and develop and build a solution using robotics.

<u>**Opportunities**</u>

- Each team will have access to tools and technology available in the classroom, including:
    - Computers and modeling software
    - 3D printers
    - Laser cutter
    - VEX robotic components (must be shared)
    - Arduino, Raspberry Pi, other microcontrollers/ microcomputers
    - Tablets
    - Motors, LEDs, actuators, and other components
    - Tools

- Teams may bring in supplies or tools
- Teams may request specific supplies
- Team structure is flexible, but a variety of “experts” is recommended (each person can be an expert in multiple areas)
    - Programming expert
    - Mechanics expert
    - Management expert
    - Documentation/ communication expert

Requirements

- Progress should be monitored via student engineering notebooks.
- Weekly or biweekly team meetings
    - Each team briefly presents progress, problems, or future directions.
    - Rotate presenter for each team so that everyone gets a chance to present.
    - Foster a constructive, positive atmosphere in which everyone should be willing to defend their ideas and accept constructive criticism.
    - This may have to be modeled or reinforced with students.
    - Practice running efficient meetings (enforce time limits, avoid distractions and side conversations, etc.).

- Possible Deliverables/Milestones:
    - Weekly website/ blog update
        - Pictures
        - Progress
        - Examples of troubleshooting
    - Develop technical business plan/proposal
    - Video tutorials
    - “Instructables”—technical instructions for the project
    - Develop a Kickstarter-like page
    - Final project and presentation
        - Throughout the process, students should be involved in developing a rubric for their project.

4.4.1.5. TEACHER NOTES

- An example of a college, robotics capstone course (University of Washington, CSE481C: Capstone Software - Robotics): https://sites.google.com/view/cse481sp17

- Fidelity and depth of project should reflect allotted time for capstone. Ideally, the capstone project will require one quarter to one semester (more advanced).

- Requirements or deliverables can be customized or developed by individuals or teams. For example, those interested in business should develop a full business proposal; students interested in engineering could create a technical engineering report; other deliverables could capitalize on student talents or other interests—website design, YouTube videos, etc.

- Identifying a problem will likely be the biggest hurdle for most students. Students should capitalize on their own experiences and interests. To get started, students could:
    - Think about the major uses of robotics—automation, dangerous work, etc.
    - Reflect on students' lives, jobs, etc.—where could robotics be used?
    - Research global issues that can be addressed with robotics.

- Hot topics in robotics
    - Ethics
    - Autonomy (autonomous vehicles)
    - Artificial intelligence
    - Soft robotics
    - Humanoids
    - Human interaction
        - Trust/predictability
        - Control vs. interpretation
        - Therapy robots
    - Bioengineering
    - Wearable robotics
    - Remote sensing
    - Biomimetics

- Alternatively, there are many robotics contests in which students can directly compete as a capstone or to use challenges as a springboard for developing their robot.
- For a fun, possibly shorter capstone or "break" from their big project, students can create a robotic Rube Goldberg Machine

5. RESOURCES

This section contains links to resources that may be used during the course.

5.1. STANDARDS

- Vex Robotics Curriculum mapped to standards: http://curriculum.vexrobotics.com/teacher-materials/standards-matching-and-accreditation
- ROBOTC Vex Trainer mapped to standards (p70-79): http://education.rec.ri.cmu.edu/wp-content/uploads/2015/02/vex_cortex_product_overview.pdf
- ISTE (International Society for Technology in Education) Standards: https://www.iste.org/standards/standards/for-students

5.2. ADDITIONAL RESOURCES AND LINKS TO INFORMATION

Table 2: Additional Resources

Title	URL	Brief Description
Modkit for Vex	https://www.vexrobotics.com/iq-modkit-for-vex-g.html	Graphical programming environment for VEX
Vex EDR Forum	https://www.vexforum.com/	Forum for VEX
VEX Reference Index	http://www.education.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/index_reference.html	Resources for all things VEX
ROBOTC-- Lego Mindstorms EV3	http://education.rec.ri.cmu.edu/previews/ev3_products/ev3_intermediate/	Programming curriculum for Lego Mindstorms
Robot Shop	http://www.robotshop.com/	Parts, ideas, resources for robotics
SparkFun	https://www.sparkfun.com/	Source for sensors, robotic and electronics parts. Educational resources
Adafruit	https://www.adafruit.com/	Source for sensors,

		robotic and electronics parts. Educational resources and tutorials
Instructables	http://www.instructables.com/	Resource for project ideas, contests, and tutorials. Writing an Instructable could be a great technical writing project.
GitHub	https://github.com/	Software development platform. Code repository-- can find pertinent coding for projects