



=



Imagineering

+



Programming

+



Mechanisms

+



Structures

1

## Lab Certified

+11 hours  
+8 projects  
+10 wizchips



**Brainstorming** : The practice of divergent thinking to generate lots of ideas.

**Planning** : Learn to break down a goal into actions to achieve it.

**Prototyping** : Test a design concept on a smaller model of the project.

**Setting Goals** : Learn to work towards a specific objective.

**A-Frame** : Use LEGO materials such as straight beams, angled beams, friction pegs and connector pegs to build a triangular structure. An A-Frame is a strong foundation for many designs.

**Cantilever** : A projecting beam supported at only one end and well-anchored on the other end.

**Cube** : Use LEGO materials such as straight beams to build a basic cube structure.

**External Bracing** : Techniques to build strong frames using rigid beams and other LEGO materials.

**Rigid Beam** : Use overlapping beams to create a long rigid structure.

**Differential Drive Vehicle** : A simple 2 wheeled robot chassis design that is easy to build and very maneuverable. It turns by running motors at different speeds and in different directions. It incorporates an elemental machine that uses wheels, axles, and bushings.

**Motor Control** : Commands for controlling motors: wbRunMotors, wbMoveSeconds, WbMoveRotations, wbMoveDegrees.

**NXT Button Control** : Reading and reacting to NXT button input: isButton, wbMoveUntil.

**Tool Chain - 1** : Create and Play a simple program on the Robot; Turn on and off the Brick and interrupt a running program.

**Touch Sensor Control** : Reading and reacting to Touch Sensor input: isTouchDown, isTouchUp, isTouchPressed, wbMoveUntil.

**Wait Commands** : Commands for making the robot wait for input or for time to pass: wbWait, wbWaitSeconds, wbWaitUntil.

**Ready To Teach - Lab Certified** : Understand how to teach all knowledge nuggets at this level

2

## Explorer

+13 hours  
+12 projects  
+10 wizchips



**PVC Pipe Structure** : Learn to integrate PVC pipe with LEGO to create bigger structures.

**Wizbits** : Students can use end caps and brackets to combine LEGO and PVC pipe into a single robot structure.

**Wizbrick** : A Wizbrick is a plastic housing specially designed to make it easier to attach a wide range of materials to an NXT brick, and to make it easier to plug and unplug the NXT brick so work in progress can be easily saved without dis-assembly.

**Zoob Blocks** : Able to use all Zoob block types and connect them with wizbits to create ridiculously cool robots.

**Display Control** : wbClear, wbPrint, Image Import, Image Editor, wbDraw.

**Flow Control - 1** : Simplest introduction to loops and logic: while(forever), if- else.

**LED Wand** : Use an LED Wand to create light effects for the robot: wbSetWand.

**Parallel Programming** : Using more than one sequence to accomplish multiple robot behaviors at the same time.

**Play Station Control - 1** : Digital control using isButtonDown.

**Speaker Control** : wbPlayFile, Sound Recorder, Sound Import, wbPlayTone.

**Tool Chain - 2** : Check connectivity, check battery life, check available space, download and delete files.

**Understand Explorer Nuggets** : Assisting mentor must understand and be able to teach all nuggets at the Explorer level.

This is a nugget checked off by a lead mentor or Timur.

**Video Upload** : Able to correctly upload videos captured from the lab. There is a quick example of uploading a video.

**Mentors must also know to name projects by the approved project's name in order to upload. The format is**

**Project\_Name.SKU#.Name.Name** Tip: if names are excluded from the file name, it is attached to all students in the lab e.g.

**Project\_Name.SKU#** if you need to differentiate between the same named people then just add the first letter of the last name, and if that doesn't then add the next letter and keep going until it does. Example: Project\_Name.3454.Johnch.Johnch

3

## Apprentice

+17 hours  
+12 projects  
+15 wizchips



**Troubleshooting** : Learn to systematically diagnose and fix flaws in a robot.

**Color Sensor Control** : Reading and reacting to color sensor input: wbRead, isInRange, isNotInRange, wbSetFloodLight.

**Distance Sensor Control** : Reading and reacting to distance sensor input: wbRead, isInRange, isNotInRange.

**Flow Control - 2** : while(condition), if-then-else, nested if statements, and random numbers.

**Light Sensor Control** : Reading and reacting to light sensor input: wbRead, isInRange, isNotInRange, wbSetFloodLight.

**Sound Sensor Control** : Reading and reacting to sound sensor input: wbRead, isInRange, isNotInRange.

**Projects by level** : Mentors must know to name approved projects done at different levels differently. The level identifier must be included- e.g. World's Tallest Robot2..They are 1 for Lab Certified, 2 for Explorer, 3 for Apprentice, 4 for Wizard, and 5 for Master. Approved projects will be named with the identifier in the curriculum documents and database.

**Promotions** : Able to promote students through the web site. Quick tutorial on how to promote a student or students to the next level.

**Tracking Absences** : How to mark students absent in your program. Mentors must also know to email Lisa regarding any absences on that day.

4

## Maker

+19 hours  
+15 projects  
+15 wizchips



**Gearing down** : Understand and use gear ratios to create a slower rotating mechanism with higher torque.

**Gearing up** : Understand and use gear ratios to create a faster rotating mechanism.

**Pulley** : An elemental machine with a rope and wheel. Be able to integrate and apply a pulley wheel.

**Screw Gear** : An elemental machine that converts rotational motion into linear motion.

**Line Following Algorithm** : Integrate what has been learned in a line following algorithm.

5

## Imagineer

+23 hours  
+15 projects  
+15 wizchips



**Ackerman Steering** : A front wheel steering mechanism for a car using a linkage design.

**Four Bar Linkage** : A linkage structure with 4 links. A foundational design pattern.

**Gripper** : Create a gripper using a 4-bar linkage.

**Slider Crank** : A common linkage to translate rotation into lateral motion.

**Arithmetic Operators** : Using common math operators (+, -, \*, /, %) to do arithmetic with variables.

**Arithmetic Relations** : Advanced logic with variables: ==, <, <=, >, >=.

**Boolean Operators** : Methods for writing more sophisticated programming logic: and, or, not.

**Compass Sensor Control** : Controlling Robot Navigation using a compass sensor.

**Magnetic Sensor Control** : Use a Magnetic Field Sensor to control the robot and find objects.

**Play Station Control - 2** : Reading and reacting to analog controller inputs.

**Rotation Sensor Control** : Reading and reacting to rotation sensor input.

**Tool Chain - 3** : Log data to a file and upload files from the robot. Useful for debugging programs.

**Variables** : variable declaration, assignment, modification and use; basic data types: boolean, int, string, double

6

## Wizard

+37 hours  
+15 projects  
+15 wizchips



**Accelerometer Sensor Control** : Reading and reacting to accelerometer sensor input: acceleration in 3D.

**Data Structures** : Learn how to design and use arrays and lists of primitive data types.

**Feedback Control** : A control loop that achieves a setpoint by minimizing the distance between the current and target values.

**File Input and Output** : Learn to use the Java File Input and Output libraries.

**Flow Control - 3** : break, continue, for loops, exception handling.

**Functions** : Creating user defined functions to package up code for easy re-use.

**Graphics** : 2D Animation and Simulation using java Graphics library.

**Gyroscope Sensor Control** : Reading and reacting to gyroscope sensor input: velocity.

**Math Functions** : Use built in math libraries for common mathematical functions: random, abs, min, max, avg, sum,....

**String Manipulation** : Learn to parse and compose strings using the java string library.

7

## Master

+43 hours  
+15 projects  
+15 wizchips



**Behaviors** : Learn behavior-based control strategies, a common paradigm for high level robot control.

**Navigation - 1** : Using odometry to estimate and track robot position in 2D.

**Object-Oriented Programming** : Learn how to design and use objects in your programs.

**PID Control** : One of the most important low level control paradigms in robotics.