**Wave Software Goals and Proficiencies**

Level 1 – Basic Concepts

* Can compile code and fix (or at least know where to find) errors
* Can download code to the robot
* Functions: Write a function that may or may not take parameters, modifies a piece of data or reads a piece of data, then returns a value related to that data.
* Classes: Understand the concept of classes and be able to retrieve and set information from the RoboRio (read joystick axis, set motor outputs, etc).
* PID: Knows basically how the value of the input will affect the value of the output (direction and relative speed based on position). Can place maximum power limits on the outputs.

Level 2 – Intermediate Concepts

* Functions: Write a function that may or may not take parameters but modifies a member variable that can be read by another function.
* Classes: Can write a class interface for a robot subsystem.
* Classes: Can write a basic Command (ex. turn motor on until sensor is tripped) and tie it to a joystick button action or include it in a Sequence.
* Debugging: Able to debug code written by other Wave members
* Can save code to repository (ex. GitHub)
* Can Image a RoboRio
* Can check and set the CAN ID’s of controller components
* Basic understanding of LabView (can program a simple drive, turn motors on/off, read sensor inputs)

Level 3 – Advanced Concepts

* Classes: Can write a complex Command (ex. non-default constructors, completes a variable sequence) and tie it to a non-button action (ex. joystick axis).
* PID: Can tune a PID. Can set appropriate target tolerances.
* Debugging: Able to debug code written by other teams (think: outreach)
* Can break a robot down into logical subsystems (ie. System Design).
* Can create a robot program using Robot Builder.

**Student Lead Requirements**

* At least at proficiency level 2.
* Can work independently (without a software mentor present) towards completing a software task and can report on progress that’s been made, or report on the different methods used to try to complete the task.
* Works with, and can mentor, the junior software members to complete tasks.
* Works with other subteam Student Leads to learn what the robot needs to do, and provides input to the other subteam Student Leads to make sure that it is being designed in a way that can be controlled by software.