

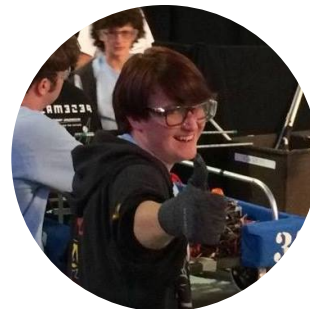
Intro to FIRST Programming

Kepler Sticka-Jones and Mark Van der Merwe

Introductions

- **Kepler Sticka-Jones**

- Computer Science Major at the University of Utah
- Graduated from Judge Memorial Catholic High School in Spring 2016
- Former Lead Programmer for JudgeMent Call Robotics - Team 5933 (2016 Season Rookie Team and Utah FRC Finalist)



- **Mark Van der Merwe**

- Computer Science Major at the University of Utah
- Graduated from Academy for Math, Engineering, and Science in Spring 2016
- Former Programming Team Lead for AMES Robotics - Team 3243



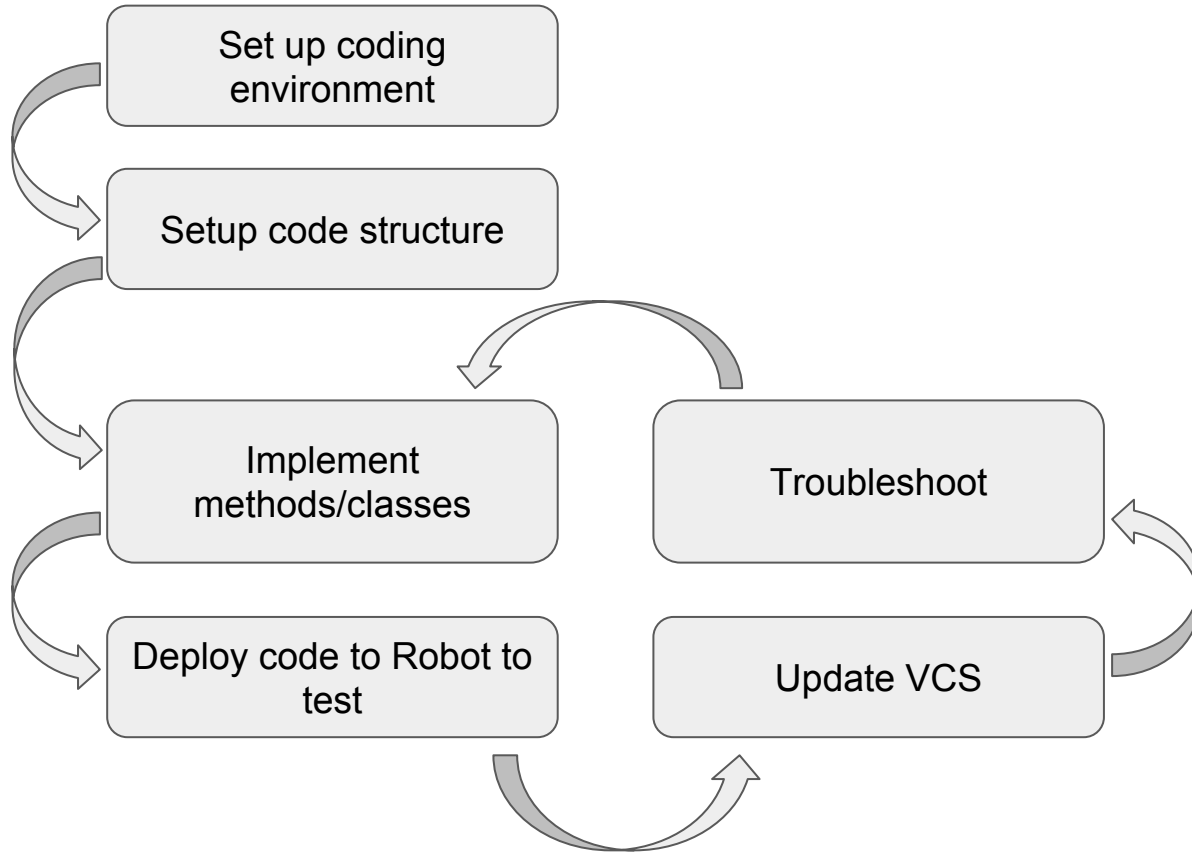
Agenda

- Present the basics of FRC programming.
- Introduce Code Lifecycle
 - Starting development
 - IDE's, programming languages, Code Structures.
 - Object-oriented Programming basics
 - Classes, methods, variables
 - Implementing with WPI Libraries
 - Deploying code to the robot
 - DriverStation
 - Github (VCS)
 - Troubleshooting
- Answer your questions!

What is programming?

- Hardware and Software
- Software tells the hardware what to do.
- Think like a computer.
 - Programming Language syntax v. semantics
- For FRC, we use programming to read signals from the robot, read input from the operator to send instruction to the robot to drive and move game pieces, and to run the robot in autonomous mode.





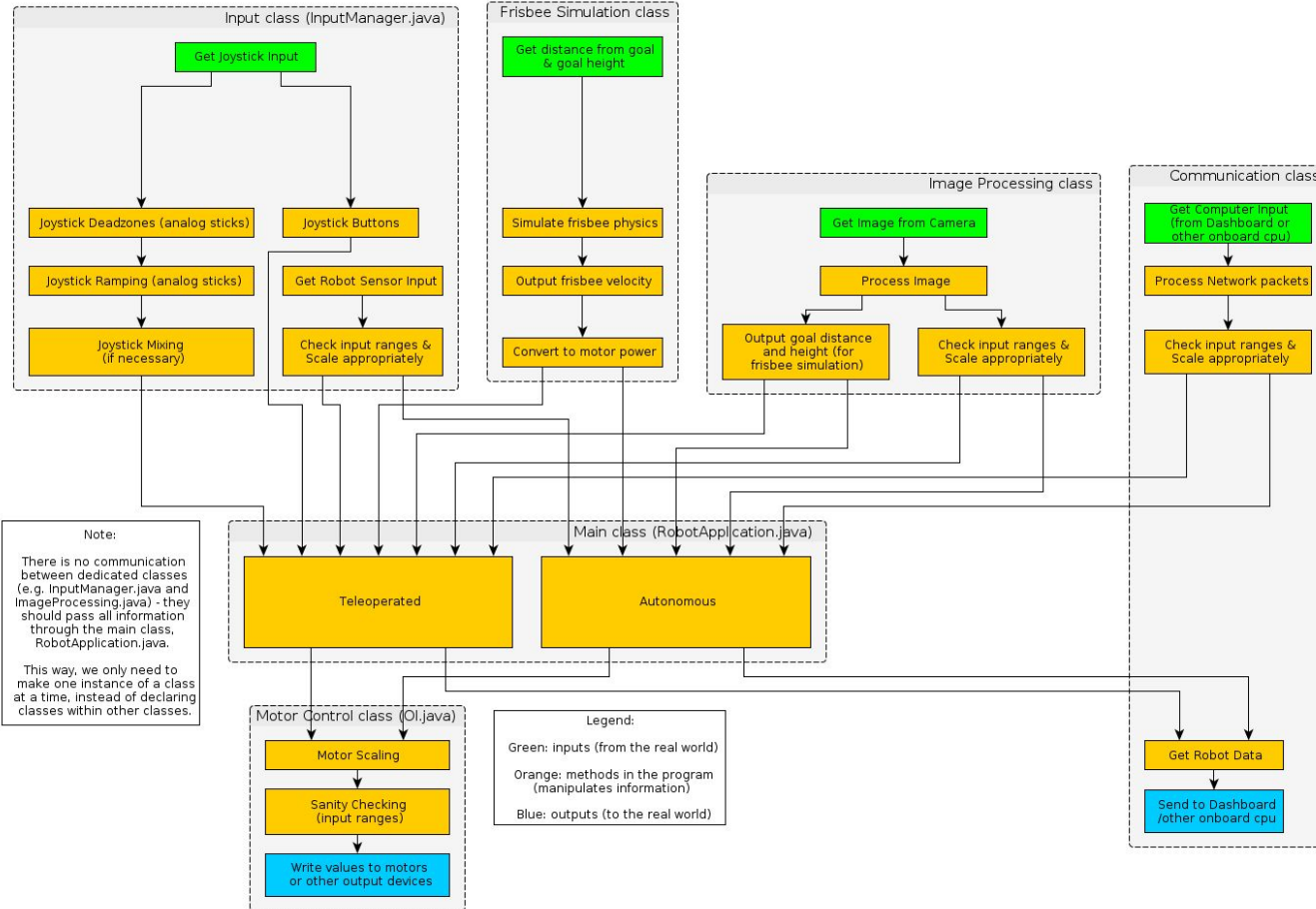
Setting Up Coding Environment

- Integrated Development Environment (IDE)
 - Easy place to code.
 - Integrates development with building, running, storing code.
- Eclipse
 - Robotics plugins
 - RobotBuilder
 - WPILib
 - Java or C++
- LabView
 - Graphical UI

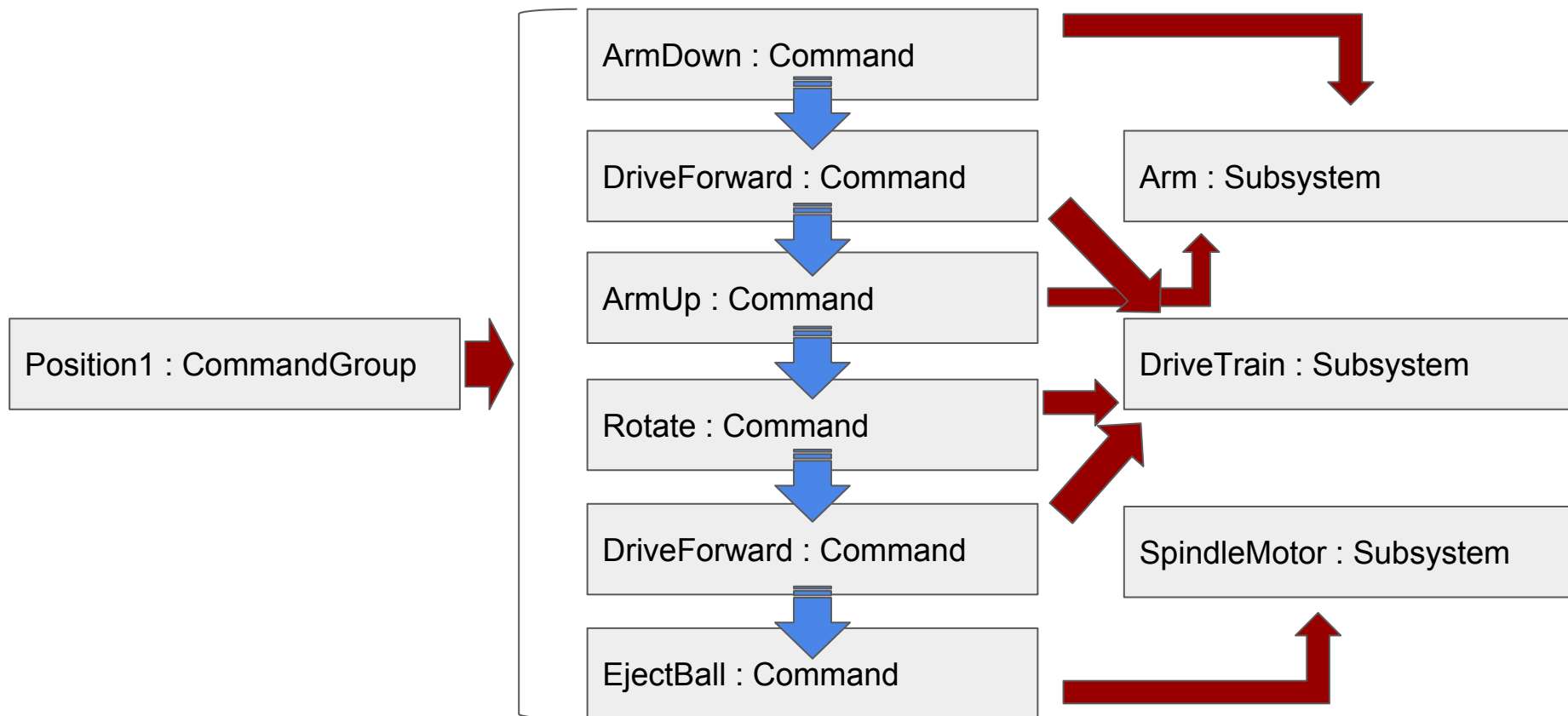
Object Oriented Programming

- Bicycle analogy
 - Break up into logical components, describe attributes and functions of components
 - Some things describe every bike / every bike can do some things but others are specific to types of bikes (inheritance)
 - Yet others specific to individual bikes
- Classes and Objects
- Properties and Methods
- Inheritance, Constructors, Prototypes, Overloading, Polymorphism, Virtuals, Reflection, Implementation / Interfacing / Abstracts, etc., etc.





Command-Based Programming



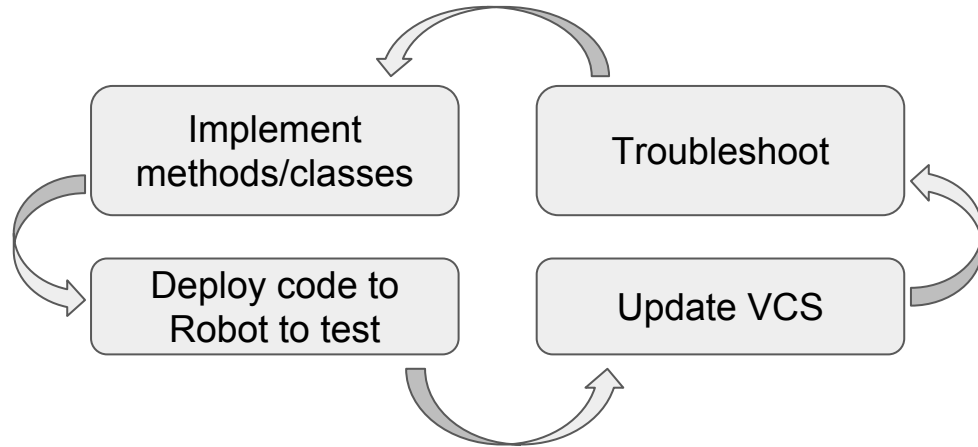
Setting up Code Structure

- Separate your Intentions (Object-Oriented Programming)
- Command-Based Programming
 - Split robot into subsystems
 - Develop commands using those subsystems.
 - RobotBuilder
- Iterative Robot
 - Split up robot components on your own.
- Set up the hierarchy of your robot. Create boilerplate code (classes, methods).

RobotBuilder for Boiler Plate

- Part of WPILib Eclipse Plugins
- Helps you to plan your robot codebase and create boilerplate
 - Drag-and-drop UI
 - Create Subsystems for the different mechanisms on the robot
 - Create Commands
 - Create Operator Interfaces (Joysticks)
- Exports to a Java project with all the files needed to start implementing the robot

The Programming Cycle



Implementing Code

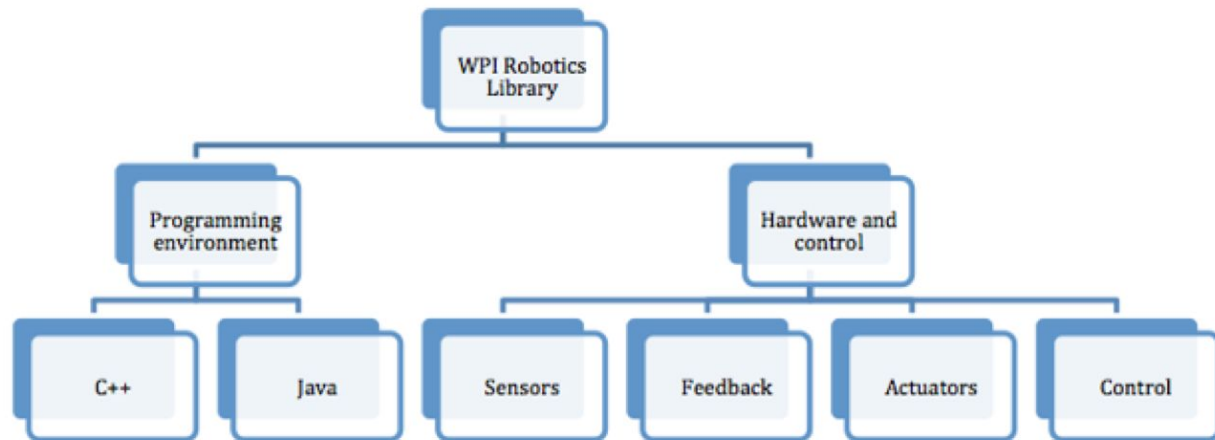
- Put code into the methods and classes you've just created.
- Use Java logic paired with WPI library objects and methods.



Relevant xkcd

WPI Libraries

- FRC creates libraries that you can add as plugins into Eclipse. This gains you access to a multitude of information for your benefit.
 - Provides you with the functions you need to run the robot.
 - Provides example code.
 - Provides a structure for you to develop your code off of.
 - Plugins

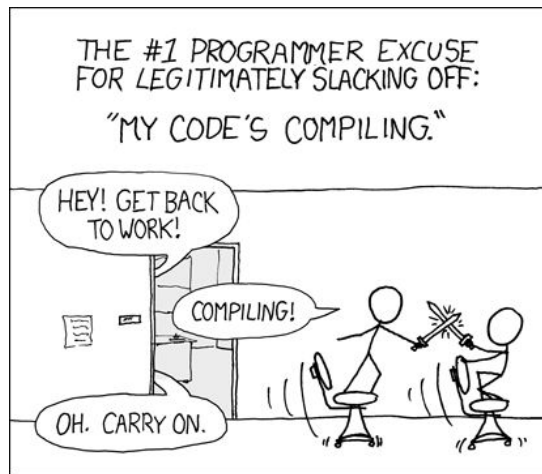


WPI Libraries

- Tutorials
 - See tutorials and get information on all of the WPI Library resources
 - <https://wpilib.screenstepslive.com/s/4485>
- Documentation
 - <http://first.wpi.edu/FRC/roborio/release/docs/java/>

Deploying Code

- Deploy Code to the RoboRio within Eclipse
 - Connect by wifi, ethernet, or USB
- Connect to the RoboRio with the Driver Station
 - Use the Console to view print lines. Let's you troubleshoot!
 - Use Riolog from a connected laptop.
- Run from the Driver Station
 - Figure out what's working and what's not.



Relevant xkcd

Version Control System - Git

- Use Git
 - This gives you the opportunity to visualize how your code base grew and changed from build season to competition
 - Allows you to explain why code was changed without adding comments to code
 - Allows teammates to create changes independently of you
- Use GitHub
 - Host code for free
 - Share code with team mates
 - Share code with FIRST community
- Use GitHub Flow
 - Make branches for any group of changes
 - Make commits to the branch
 - When finished open a Pull Request on repo
 - Use Pull Request to discuss changes with team
 - Merge into master branch after discussion

Troubleshooting

- Stack Overflow
 - Stick your problem in Google, you never know...
- Chief Delphi and other Forums
- FRC Documentation
 - Tutorials
 - Java docs
- Logging
 - Put prints in your statement
- Debugging