Lesson Plan:

Tell:

- Object—A piece of hardware (ex. motor controller, solenoid, servo, controller, etc.)
- Variable—A piece of stored information (ex. Integer (int), Decimal (double), Word (String), Object)
- Logic Flow:
  - Methods—Perform an action, such as setting a motor's speed.
    - Parameter—A variable for the method.
  - Commands—Call a method, can be called by an Xbox button.
  - Button bindings—When I press <Button>, <Command> happens.
- Syntax:
  - Semicolons at the ends of lines.
  - Scope resolution operator = period
  -

Do:

MVP:

- Ctrl + Shift + P -> WPILib: Create a new project.
- Settings (Language: Java, Name: Something, Team Number: 4028, etc.)
- Navigate to ExampleSubsystem.java
- Declare motor (ex. CANSparkMax m_motor)
- Initialize motor in constructor.
- Make new method that takes in a double "speed" param.
- Motor.set(speed);
- Make a command that calls the method.
- Navigate to RobotContainer.configureButtonBindings()
- Reference the button with m_driverController.<button>().onTrue();
- Pass in the previously created command (m_driverController.<button>().onTrue(ExampleSubsystem.<command>(<parameter>)))
- Lamba

Advanced:

- Declare encoder (ex. RelativeEncoder m_encoder)
- Initialize the encoder with m_motor.getEncoder();
- Create a new command method that uses the set method in the factory run() method along with command decorators to use encoder feedback to rotate to a specific position.
  - Store getEncoder() into a variable in a runOnce() with an andThen() decorator at the end to run the run().
  - Run motor.set() in the run() method
  - Use the .until() decorator to limit the encoder position to the relativePosition + the desired amount of rots.
- Bind this new command to a new button in RobotContainer.configureButtonBindings()

Level 2:

- Casing best practices
    - camelCase for variables
    - CAPITAL_SNAKE_CASE for constants.
    - TitleCase for Classes + Enums
- Formatting/Organization
- Enums + State Machines
- Subsystem Planning
    - Be Extensible!
- Basic PID – position control
- Stringing Commands together
    - Parallel & Sequential Groups
    - Multi-Subsystem Commands
- Current Limiting & Sensing
- Phoenix Tuner, REV HW Client
- RoboRIO Imaging & Maintenance

Remember: git

Level 3:

- Advanced PID
    - Tuning
    - Velocity
    - Motion Magic & NO bad
    - Profiled PID
- Robot Architecture
- Conditional & State-Sensitive Commands
- SysID & Feedforward
- Accessing External Data
    - Coprocessors
    - NT – LL, PV
    - Color Sensor
    - TOF sensor

Level weee:

- Coprocessors
- Interfaces & Abstract Classes
- Rust ()
-