Advanced Programming Exercises

# Challenge 1: Use a state machine to create a basic autonomous program using a looping opmode. This program can do anything, but rather than using a linear opmode, use a standard looping opmode.

# Challenge 2: Create a class hierarchy for a robot using the four principles of object-oriented programming (abstraction, encapsulation, inheritance, and polymorphism). Example classes can include a drivetrain, a sensor system, a scoring system, and so on.

# Challenge 3: Using a gyroscope, create a program that turns 90 degrees. Then, using encoders, drive forward approximately 3 feet.

# Challenge 4: Use PWM control on a motor controller to explore how motors can be synchronized.

# Challenge 5: Appropriately use access modifiers on existing variables and understand their function/purpose.

# Challenge 6: Brainstorm and implement a new sensor into your existing robot.

# Challenge 7: Document existing code using flowcharts and/or UML diagrams.

# What other ideas do you have?

# Feel free to explore Java and the FTC SDK on your own!

Key Terms/Points

A class is a collection of objects of a similar type.

An object is an instance of a class.

A package allows a developer to group related classes (and interfaces) together.

Object-oriented programming is based on the concept of “objects”. Objects contain data in the form of fields/attributes. Objects contain code in the form of procedures/methods. An object's methods can access and often modify the attributes of the object.

Abstraction is the implementation of an object that contains the same essential properties and actions we can find in the original object we are representing.

Encapsulation is the internal representation of an object is generally hidden from view outside of the object’s definition.

Inheritance refers to how classes can inherit attributes and behavior from pre-existing classes called base classes, also known as superclasses.

Polymorphism is the ability of an object to take on many forms.