# Java Programming for FTC Robotics

COURSE OUTLINE

## Course Description

This class aims to help students transition from Blocks to custom Java coding for FTC robots. The course covers Java fundamentals—syntax, data types, variables, operators, control flow, and object-oriented programming—along with the advantages of coding in Java over Blocks. Students will also learn FTC-specific concepts like the FTC SDK, opmodes, hardware mapping, and telemetry, gaining hands-on experience coding opmodes in Android Studio.

**01** Introduction to Java for FTC

- Why move from Blocks to Java?
- Overview of Android Studio and the FTC SDK
- Setting up a project and understanding the workspace

**02** Java Fundamentals

- Java syntax and structure (classes, methods, main concepts)
- Data types, variables, and constants
- Operators and expressions
- Control flow: conditionals (if, else, switch)
- Loops (for, while) and their use in robotics

**03** Object-Oriented Programming (OOP) Basics

- What is OOP and why it matters in FTC programming
- Classes and objects
- Methods and parameters
- Encapsulation and modular code for robotics

**04** FTC SDK and OpModes

- Understanding the FTC SDK structure

- Difference between TeleOp and Autonomous OpModes
- Creating and running an OpMode
- Using telemetry
- Hands-on: writing a simple TeleOp program in Java

**05** Hardware Mapping and Device Control
- Hardware mapping in code
- Controlling DC motors, servos, and sensors
- Hands-on: program a robot to drive using Java

**06** Autonomous Programming
- Writing autonomous routines with Java
- Using timers and encoders for movement control
- Introduction to sensor-based programming (touch, distance, IMU, etc.)
- Hands-on: write an autonomous program to complete a simple task

**07** Best Practices and Next Steps
- Structuring reusable code (methods and helper classes)
- Organizing code for readability and maintainability
- Advanced programming: command-based, pathing, custom interfaces

**08** Questions & Comments