COOL NAME PENDING

FIRST Tech Challenge (FTC)

16-Week Elite Deep Curriculum Guide

#coolnamepending

Week 1: Game Analysis & Strategic Planning

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 2: Chassis Engineering & Design Principles

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 3: Drivetrain Systems (Tank vs Mecanum)

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 4: Gear Ratios, Torque & Mechanical Advantage

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 5: CAD Design Foundations (Onshape/Fusion)

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 6: Java Programming Fundamentals (FTC SDK)

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 7: Motor Control & Encoder Systems

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 8: PID Control & Motion Precision

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 9: Sensor Integration (IMU, Distance, Color)

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 10: Autonomous Path Planning & State Machines

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 11: Mechanism Engineering (Slides, Arms, Linkages)

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 12: TeleOp Optimization & Driver Training

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 13: Data Logging, Debugging & Performance Metrics

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 14: Engineering Notebook & Judging Preparation

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 15: Full Robot Integration & Match Simulation

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.

Week 16: Final Optimization & Championship Readiness

Learning Objectives

Students will understand theoretical foundations, apply hands-on engineering design, and demonstrate measurable improvement in robot performance. Each week integrates mechanical, programming, and strategic thinking.

Technical Theory Session

Mentor conducts a structured whiteboard lecture covering formulas, engineering trade-offs, and FTC-specific applications. Students calculate drivetrain speed, torque requirements, center of gravity, or programming flow logic depending on the weekly theme.

Hands-On Lab Implementation

- Design or refine subsystem according to weekly focus.
- Perform controlled test trials (minimum 8–10 repetitions).
- Measure and record performance data.
- Analyze failures and redesign accordingly.

Programming Integration

Students implement structured Java modules including hardware mapping, modular methods, sensor feedback loops, and debugging strategies. Code must follow clean architecture and reusability principles.

Engineering Documentation Requirement

Each student contributes to the engineering notebook with diagrams, calculations, testing results, and improvement analysis. Documentation must justify design decisions using data.

Performance Metrics & Evaluation

Subsystem improvements are evaluated based on repeatability, speed, accuracy, durability, and strategic effectiveness. Weekly review meeting determines next iteration priorities.