



# Software Engineering 101

This course is a foundational course that is taught with all of the courses in Karma Crafters



# Module: Mastering Git and GitHub Workflow

Equip yourself with essential version control skills and collaboration techniques in this module, where you'll master the concept of Git and GitHub to streamline your development process.

- **Git Fundamentals:** Delve into the basics of Git, covering essential operations like adding, committing, and branching. Learn how to effectively manage your codebase's history.
- **GitHub Essentials:** Create repositories, and master the concept of pulling and pushing changes using GitHub. Seamlessly synchronize your local codebase with the cloud.
- **Hands-On Assignment and Branching:** Apply your knowledge through a practical assignment while exploring the power of branching to manage different lines of development.
- **Merge, Rebase, and Conflict Resolution:** Learn essential techniques for merging and rebasing branches, while gaining insights into resolving merge conflicts.
- **Group Collaboration Dynamics:** Dive into the dynamics of collaborative work, experiencing the power of Git and GitHub in enabling smooth group-based projects.

## Outcome

By the end of this module, you'll be well-versed in using Git and GitHub as your trusty development companions. Armed with the ability to navigate version control, seamlessly collaborate, and resolve conflicts, you'll be prepared to excel in team projects and contribute effectively to the world of software development.



# Module: Essential Programming Fundamentals

Lay a strong foundation in programming with this module, encompassing vital concepts that form the backbone of coding proficiency.

- **Variables and Data Structures:** Dive into the core of programming by understanding variables and exploring fundamental data structures like strings, numbers, arrays, and objects.
- **Operators and Control Flow:** Master the concept of using operators to manipulate data and navigate through control flow structures, setting the stage for efficient coding.
- **Looping Mechanisms:** Discover how to iteratively process tasks using loops, gaining insights into creating repetitive actions in your programs.
- **Functions and Modularization:** Unlock the power of functions, essential for writing organized and reusable code, setting the stage for efficient problem-solving.
- **Introduction to Object-Oriented Programming:** Step into the world of object-oriented programming by learning about classes, encapsulation, and the principles that enable scalable and maintainable code.

## Outcome

By the end of this module, you'll have a robust understanding of programming basics, empowering you to craft efficient and organized code. Armed with these essential skills, you'll be ready to embark on more complex programming challenges and explore diverse coding languages with confidence.



# Module: Effective Coding Practices

Elevate your coding proficiency with this module, where you'll master essential practices that enhance code readability, maintainability, and efficiency.

- **Comprehensive Documentation with JSDocs:** Dive into the concept of writing comprehensive JSDocs that provide clarity and context to your codebase, making collaboration smoother.
- **Optimal Variable Naming Conventions:** Learn the significance of well-chosen variable names, ensuring your code communicates its intent effectively to both you and fellow developers.
- **Code Formatting and Indentation:** Discover the impact of consistent code formatting and indentation on readability, and grasp techniques to maintain a clean and organized codebase.
- **Principles for Robust Design: SOLID, YAGNI, and DRY:** Familiarize yourself with foundational principles like SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion), YAGNI (You Aren't Gonna Need It), and DRY (Don't Repeat Yourself). Uncover their role in crafting robust and efficient code.

## Outcome

By the end of this module, you'll possess a toolkit of effective coding practices that transform your code into a well-structured, readable, and maintainable masterpiece. Armed with these skills, you'll be poised to create efficient, collaborative, and future-proofed codebases.



# Module: Comprehensive Test Cases

Elevate your development process by mastering the concept of creating comprehensive test cases that ensure the reliability and robustness of your code.

- **The Importance of Test Cases:** Understand the significance of test cases in ensuring code quality, identifying bugs, and validating the accuracy of your applications.
- **Types of Test Cases:** Explore different types of test cases, including unit tests, integration tests, and end-to-end tests, to cover various aspects of your application.
- **Test Case Design:** Dive into the intricacies of designing effective test cases, encompassing different scenarios, edge cases, and potential vulnerabilities.

## Outcome

By the end of this module, you'll be equipped with the skills to create comprehensive test cases that ensure the reliability and robustness of your codebase. Armed with these abilities, you'll contribute to creating software solutions that meet high-quality standards, enhancing your role as a proficient developer.



# Continuous Practice

**Note: These concepts will be applied to your everyday work.**

- These modules will just be an introduction during your first week of classes. But will be recalled and practiced almost every class during your project development.
- Also, these are some of the most important module that the industry will use to vet you during your interview and hiring. So make sure you understand them and make them your habit.