STALWART LEARNING

commit today!

1000+ Industry Relevant Courses || 500+ Industry Veteran Trainers

ML, AI & Gen AI || Cloud (AWS, Azure, Google Cloud) || Cyber Security || Data Engineering || DevOps || Programming Languages || Java/J2EE || Microsoft Technologies || Mobile & Web Technologies || Agile || Project Management || Business Analysis

---

Course Title:

AI-Augmented Software Development with GitHub Copilot and Beyond

Category:

GitHub Copilot Specialization

Total Duration:

24 Hours (3 Hours/Day)

Course Overview:

This specialization empowers software developers, engineers, and technology leaders to harness the full potential of **AI-assisted software development** using **GitHub Copilot** and related tools. The course blends hands-on coding practice with strategic understanding of how AI is transforming software engineering — from code creation to testing, documentation, and DevOps automation.

Participants will learn to **leverage GitHub Copilot and complementary AI tools** across the Software Development Lifecycle (SDLC), implement **responsible AI practices**, and prepare for the **next generation of AI-augmented development environments**.

Course Objectives:

Upon completion, participants will be able to:

- Apply AI-assisted coding, debugging, and refactoring techniques using GitHub Copilot.
- Integrate AI tools into daily workflows for faster, cleaner, and more maintainable code.

- Automate documentation, testing, and DevOps with AI-enabled extensions.
- Understand and apply governance, security, and ethical frameworks for AI-generated code.
- Evaluate and deploy AI augmentation strategies within enterprise software ecosystems.

Pre-requisites:

- Basic programming experience in any modern language (e.g., Python, JavaScript, C#, Java).
- Familiarity with GitHub repositories, version control, and IDEs (VS Code preferred).
- Optional: Exposure to Agile or DevOps workflows.

Detailed Curriculum

Module 1: Foundations of AI-Augmented Development (Day 01)

- The evolution of AI in software engineering
- AI's role in accelerating developer productivity
- Introduction to Gen AI, LLM, core concepts
- Overview of GitHub Copilot within GitHub and GitHub Enterprise environments
- Setting up Copilot-ready environments (VS Code, extensions, configuration)
- Model selection in AI coding assistants
- **Industry insights:** AI adoption examples from BFSI, healthcare, manufacturing, and retail

Hands-on:
Configure GitHub Copilot settings in VS Code, validate authentication and repository access, and generate initial AI-assisted code suggestions for a simple business use case

Outcome:
Participants understand how AI coding assistants enhance developer productivity, how GitHub Copilot fits within GitHub Enterprise environments, and are fully prepared for deeper hands-on usage in subsequent modules.

Module 2: Mastering GitHub Copilot Fundamentals (Day 02)

- How GitHub Copilot understands developer intent, prompts, and context windows
- Writing effective prompts for intelligent code completion using GitHub Copilot
- Working across languages and frameworks with GitHub Copilot (Python, JavaScript, Java, C#)
- Real-time code suggestions, debugging, and optimization assistance from GitHub Copilot
- **Copilot Prompt Engineering Techniques:**
    - Ask-oriented, context-rich, and multi-line prompting
    - Role-based and goal-driven prompt design
    - Combining code and natural language for better context alignment
    - **Prompt iteration and evaluation:** maintaining prompt histories and comparing outputs for consistency and accuracy

Hands-on:
Build a simple end-to-end project using GitHub Copilot–assisted development, focusing on prompt refinement and contextual guidance.

Outcome:
Participants learn effective prompt-writing strategies and understand how GitHub Copilot's contextual awareness influences code suggestions and output quality, while retaining human control over design and correctness.

Module 3: Advanced Coding with AI Assistance (Day 03)

- Intelligent debugging, code refactoring, and test automation using GitHub Copilot
- AI-assisted error detection, dependency awareness, and logic optimization with GitHub Copilot
- Pair programming with GitHub Copilot for large codebases
- Integrating GitHub Copilot with unit testing frameworks (PyTest, JUnit)
- **Identifying and Managing Hallucinations in AI-Generated Code:**
    - Understanding AI-generated inaccuracies and incomplete logic
    - Techniques for manual validation and test-driven verification
    - Safe fallback strategies and mandatory code review workflows
- **Prompt evaluation:** comparing response consistency and correctness across iterations

Hands-on:
Refactor a legacy codebase using GitHub Copilot recommendations and validate outputs using unit tests and structured code review.

Outcome:
Developers can confidently debug, refactor, and test code using GitHub Copilot, while mitigating risks through verification, testing, and human oversight.

Module 4: Beyond Copilot — The AI Developer Toolkit (Day 04)

- Comparison of GitHub Copilot, Cline, Tabnine, and Amazon CodeWhisperer
- AI Native IDEs Overview (Cursor, Windsurf, Anti Gravity)
- AI-assisted documentation and code explanation
- **Exploring MCP for multi-tool workflows:**
    - Conceptual overview of MCP and how it allows GitHub Copilot to interact with other AI tools or internal repositories
    - Governance, policy, and security considerations for MCP usage in enterprise settings

Hands-on:

- Generate documentation and deployment scripts using GitHub Copilot and MCP if the environment supports.
- Explore repetitive task automation using AI plugins/extensions: generating boilerplate, creating test scaffolds, or documenting code automatically.

Outcome:

- Participants gain exposure to multiple AI development tools and extensions.
- Understand how to design **practical, human-coordinated AI-assisted workflows**.
- Recognize MCP's advanced capabilities and limitations while learning to use AI safely and effectively.

Module 5: Integrating AI into the Software Development Lifecycle (Day 05)

- Embedding AI tools (GitHub Copilot) into Agile and DevOps workflows
- Using AI for requirement gathering, sprint planning, and task estimation
- AI-assisted code reviews, test coverage analysis, and QA automation
- A/B testing of AI prompts within SDLC to evaluate productivity gains and output accuracy
- Measuring AI impact on developer performance and team efficiency
- **Business Use Cases Across Industries:**
    - Banking → compliance script generation
    - Telecom → API integration automation
    - Retail → chatbot enhancement for customer orders
    - Education → auto-grading and content generation

Hands-on:
Create an AI-integrated SDLC workflow using GitHub Projects, Issues, and GitHub Copilot collaboration, including AI-assisted code reviews and test coverage improvement.

Outcome:
Participants can embed Copilot across the software lifecycle and measure its impact with prompt-driven metrics.

Module 6: Enterprise-Scale Deployment and Governance (Day 06)

- Managing AI coding assistants (including GitHub Copilot) in enterprise projects
- Integration with internal repositories, APIs, and enterprise access policies
- Policy, compliance, and organizational alignment (SOC 2, GDPR, HIPAA considerations)
- **Prompt governance**: defining rules for prompt usage, logging, and auditability
- **Best practices for enterprise enablement:**
    - User roles and access control
    - Repository and branch protection rules
    - Version control and review workflows
    - Secure data flow for AI-assisted development

Case Study:
Enterprise GitHub Copilot enablement for a product engineering team using internal repositories and standard SDLC controls.

Outcome:
Participants learn how to implement GitHub Copilot securely within enterprise environments using governance policies, repository controls, and compliance-aligned SDLC practices.

Module 7: Responsible AI, Ethics, and Security in Development (Day 07)

- Data privacy, intellectual property, and code ownership
- Bias detection and risk management in AI-assisted code
- Responsible AI principles for enterprise software development
- Secure development lifecycle for AI-assisted coding
- Governance considerations for AI-assisted development environments
- **Hallucination recognition:** identifying red flags in AI-generated code and validating logic consistency
- **Prompt validation practices:** peer-reviewed and test-driven evaluation of AI outputs

Hands-on:
Develop a Responsible AI and Secure Coding policy tailored for your organization.

Outcome:
Participants can define and implement responsible, secure, and compliant AI-assisted coding practices.

Module 8: Best Practices, Capstone, and Future of AI Development (Day 08)

- **Best Practices for Copilot and AI-Augmented Coding:**
    - Structured prompt writing and iterative refinement
    - Maintaining a prompt repository and versioning system
    - Reviewing AI-generated code through automated testing
    - Integrating hallucination checks in CI/CD pipelines
- **Emerging trends:** Multi-agent, Browser Support and autonomous A2A (Agent-to-Agent) pair programming
- **MCP in future AI workflows:** concept of standardized interoperability for AI tools
- Measuring long-term productivity, maintainability, and quality improvements
- **Capstone Project:**
  Build and deploy a full-stack mini-application using GitHub Copilot, including AI-generated documentation, versioned prompts, and automated test workflows.

Outcome:
Participants demonstrate end-to-end capability in AI-augmented software development with governance, testing, and scalability.

Deliverables & Certification:

- **Assignments** after each module
- **Capstone Project** evaluation based on innovation and AI utilization