HeroForge.AI

Lesson Workshop

Claude Code Fundamentals

The lesson workshops are your bridge from learning AI concepts to applying them in your business and life. Learning is potential. Applying what you learn gives you the power to create prosperity. We build these workshops to take approximately 45-60 minutes.

In this lesson, we will:

1. **Configure Your Agentic Environment:** Set up a secure GitHub Codespace and install Claude Code, adding a shortcut for "Dangerously Skip Permissions".
2. **Initialize a Claude Code Project:** Use the /init command to build the CLAUDE.md file, the "brain" that allows the AI to understand your entire project architecture.
3. **Explore Basic Functionality:** Practice basic Claude Code usage.

Module 1: Environment Setup & Configuration (20 Minutes)

Objective:
To establish a secure, cloud-based development environment (GitHub Codespace) and install the Claude Code CLI tool with the necessary permissions shortcuts for efficient "Agentic Engineering."

Concept Review: The Claude Code Advantage & "Dangerously Skip Permissions"

Claude Code is a CLI (Command Line Interface) tool, meaning it has direct control over your computer's terminal.

- **Safety First:** Because the agent has direct control, we run it inside a **GitHub Codespace** (a containerized cloud computer). This prevents the AI from accidentally damaging your local machine.
- **Speed:** By default, Claude asks for permission for every file edit. To move at the speed of thought, we use a command alias (dsp) to "Dangerously Skip Permissions," allowing the agent to work autonomously within the safe container.

Quiz (Answer Key below - don't peek!):

1. Why is it recommended to use GitHub Codespaces instead of your local terminal for Claude Code?
   a) It is the only way to access the internet.
   b) It acts as a safe container, so if the agent deletes files or makes mistakes, your local

machine is safe.
c) It is required by Anthropic's terms of service.

2. What is the primary function of the dsp (Dangerously Skip Permissions) shortcut?
   a) It gives the AI access to your bank account.
   b) It allows the AI to execute commands and edits without asking you to approve every single step.
   c) It turns off the AI's ability to browse the web.

Exercise: Installation and Configuration

You will now build your development environment.

Reference Material: Please open the attached **Claude-flow-setup-guide.pdf** to complete this exercise.

Step 1: Create Your GitHub Account & Repo (5 mins)

- Follow **Part 1** and **Part 2** of the Setup Guide.
- *Action:* Create a private repository named HeroForge-Lab.

Step 2: Launch Codespaces (5 mins)

- Follow **Part 3** of the Setup Guide.
- *Action:* Launch a blank Codespace. You should see a Visual Studio Code interface in your browser.

Step 3: Install Claude Code & Authenticate (5 mins)

- Follow **Part 4** of the Setup Guide.
- *Action:* Run the npm install command.
- *Action:* Authenticate using your Anthropic Console account (API billing must be enabled, or use a Claude Max account).

Step 4: Set the dsp Alias (5 mins)

- Follow **Part 5** of the Setup Guide.
- *Action:* Create the dsp alias. This is critical for the workflow Dr. Allen demonstrates.
- *Test:* Run dsp in your terminal. If Claude initializes without error, you are ready for Module 2.

Module 2: Initialization & Context Engineering (15 Minutes)

Objective:

To teach the AI agent about your project structure using the /init command and understand how to manage the agent's context window.

Concept Review: The CLAUDE.md Brain

When you drop an AI into a codebase, it is "blind." It doesn't know your architecture, your tech stack, or your coding standards.

- **The /init Command:** This scans your file structure and generates a file called CLAUDE.md.
- **System Prompt:** The content of CLAUDE.md is fed to the AI at the start of every session. It tells the agent "This is who I am, this is how this project works, and these are the rules."
- **Context Window:** The AI has a limited short-term memory (Context Window). CLAUDE.md serves as long-term memory/reference to keep the agent grounded.

Quiz:

1. What happens when you run the /init command in Claude Code?
   a) It deletes all files in the directory.
   b) It analyzes the project and creates a CLAUDE.md file summarizing the architecture and rules.
   c) It installs the Python programming language.
2. If you manually customize your CLAUDE.md file to add specific coding rules, what should you avoid doing afterward?
   a) Committing the file to GitHub.
   b) Running /init again, as it might overwrite your manual customizations.
   c) Reading the file.

Exercise: Initializing the Agentic Brain

Step 1: Populate the Project (5 mins)

Since your repository is empty, let's give Claude something to analyze.

- In your terminal (running Claude via dsp), type:"Create a simple Python script for a 'To-Do List' application. Create a README.md file explaining how to use it. Do not execute the code, just create the files."
- *Observation:* Watch how Claude operates. Hit control-o to see more of it's reasoning (while focus in on the Claude Code terminal windo). Watch how Claude generates the files in your file explorer on the left.

Step 2: Initialize the Memory (5 mins)

- Type /init in the Claude prompt.
- *Action:* Claude will scan the files it just created.
- *Action:* Open the newly created CLAUDE.md file in the file explorer.
- *Reflection:* Notice how it identified the language (Python) and the project type

automatically.

Module 3: Planning, execution, and Troubleshooting (20 Minutes)

Objective:
To practice the "Measure Twice, Cut Once" workflow using Plan Mode to architect solutions before allowing the AI to write code.

Concept Review: Modes and "Think Hard"
Claude Code operates in different modes to manage speed, cost, and accuracy.

- **Plan Mode:** The agent analyzes the request and outlines a strategy but *does not* write code or edit files. This is crucial for complex tasks to prevent "spaghetti code."
- **Build Mode:** The default mode where the agent executes the plan.
- **Think Hard:** You can instruct the model to "think hard" or "ultra think," forcing it to spend more compute time reasoning through edge cases before answering.

Quiz:

1. If you are asking Claude to architect a complex new module, which mode should you enter first?
   a) Build Mode
   b) Plan Mode
   c) Sleep Mode
2. What is the purpose of the command "Think Hard"?
   a) It makes the AI answer faster.
   b) It forces the model to use more compute to reason through complex logic or bugs.
   c) It resets the memory.

Exercise: The Planning Workflow

Step 1: The Feature Request (5 mins)

- We want to add a complex feature: "Save the To-Do list to a CSV file."
- Type /plan to enter Plan Mode (or simply tell Claude "Enter plan mode").
- Prompt:"I want to add functionality to save tasks to a CSV file and load them upon startup. Architect this solution."

Step 2: Review the Architecture (5 mins)

- Claude will produce a detailed text output explaining *how* it intends to modify the code. It will *not* write the code yet.
- *Action:* Read the plan. Does it handle errors (e.g., if the file doesn't exist)?
- *Refinement:* If the plan is missing something, tell it: "Update the plan to include error

handling if the CSV is missing."

Step 3: Execution (10 mins)

- Once satisfied with the plan, type:"This looks good. Switch to build mode and implement this plan."
- *Action:* Watch Claude write the code.
- *Testing:* Once finished, run the script (e.g., python todo.py or whatever filename it gave you) to verify it works.

Step 4: The Clean Up (Pro-Tip)

- Dr. Allen notes that Claude creates many files.
- Prompt:"Clean up and organize the project, then update the README.md."

Congratulations!

You have moved beyond simple chatbots and entered the world of **Agentic Engineering**. You now have a functioning CLI environment, a "memory" for your project via CLAUDE.md, and a workflow that separates planning from execution. This infrastructure is the foundation for building software at 100x speed.

Answer Key:

Module 1

1. **b)** It acts as a safe container...
2. **b)** It allows the AI to execute commands...

Module 2

1. **b)** It analyzes the project and creates a CLAUDE.md file...
2. **b)** Running /init again... (It will overwrite your custom rules).

Module 3

1. **b)** Plan Mode
2. **b)** It forces the model to use more compute...