

CLAUDE AI MASTERY

From Basics to Advanced — Complete Course Syllabus

9 Modules • Hands-on Projects in Every Module • Beginner to Developer Track

Prepared by EchoScale Digital

shashwatyadav.in

How to Use This Syllabus

This is a complete, teach-ready course structure. Every module below follows the same 5-part format so you can deliver it consistently:

- Objective — what the learner will be able to DO after this module (not just know)
- Tools Required — exact platforms/accounts needed, all free unless noted
- Lesson-by-Lesson Teaching Points — the exact things to demonstrate live, in order
- Hands-On Project — a real use case with step-by-step build instructions and a gradable deliverable
- Assessment — how you (or the learner) verify the module was actually learned

Course Format Recommendation

Total Duration	8–9 weeks (1 module/week), or 4 weeks intensive (2 modules/week)
Format	Live/recorded lesson (20–30 min) + hands-on project (30–60 min) per module
Prerequisite	None for Modules 1–6. Basic comfort with command line/terminal helps for Modules 7–9
Cost to run	₹0 mandatory — Claude free tier covers Modules 1–6. Modules 7–9 need a few \$ of API credit
Certification idea	Award completion certificate after Module 9 capstone is submitted and reviewed

MODULE 1: FOUNDATIONS | Week 1 · 90 min

Objective: Learner can explain what Claude is, navigate the interface confidently, and knows which model to pick for a given task.

Tools Needed	claude.ai account (free tier), Claude mobile app (optional)
Prerequisite	None

Lesson-by-Lesson Teaching Points

1.1 What is Claude & Anthropic

- Anthropic's mission: building AI safely — mention this is core to why Claude behaves the way it does
- Explain the model family in plain terms: Haiku (fastest/cheapest) → Sonnet (balanced, default for most work) → Opus (deepest reasoning) → Mythos (Anthropic's newest tier, currently limited access)
- Live demo: open claude.ai, show the model picker, point out where model name appears

1.2 What LLMs are (non-technical)

- Analogy: Claude predicts the next most useful piece of text based on patterns learned from huge amounts of text — like an extremely well-read assistant, not a search engine
- Clarify: Claude doesn't 'look up' answers by default — it reasons from training, and uses tools (web search) when it needs current info
- Teach the knowledge cutoff concept and why Claude searches the web for recent events

1.3 Claude vs ChatGPT vs Gemini — positioning

- Claude's differentiators to highlight: strong writing quality, careful/safety-conscious responses, Artifacts (live document/code preview), Projects with persistent context, strong coding via Claude Code
- Keep this section factual and even-handed — show a comparison table, don't oversell

1.4 Interface Walkthrough

- Sidebar: New chat, Projects, Recents
- Chat box: attach files, toggle web search, model selector, style selector
- Settings: Personal preferences, memory, connected apps
- Mobile app: voice mode, camera input, same account sync

HANDS-ON PROJECT: Claude Setup + First Prompt Log

Use case: Every learner needs a working environment before touching prompting — this project forces account setup and first real interaction.

Tools needed: claude.ai account, a notes doc (Google Docs/Notion)

Build steps:

1. Create a free claude.ai account and verify email/phone
2. Open a new chat and ask Claude to explain its own model family in 3 lines
3. Switch models (if available on plan) and ask the same question — compare response depth
4. Try one prompt with web search ON and OFF (e.g., 'what's today's top tech news') — note the difference
5. Write a 5-line personal note: 'What I understand Claude to be, in my own words'

Deliverable to grade: Screenshot of the chat + the 5-line personal note

MODULE 2: PROMPTING BASICS | Week 1–2 · 90 min

Objective: Learner can write a clear, specific prompt on the first try and knows how to fix a bad response with a follow-up.

Tools Needed	claude.ai account only
Prerequisite	Module 1

Lesson-by-Lesson Teaching Points

2.1 Writing Clear, Specific Prompts

- Teach the 4-part prompt formula: Task + Context + Format + Constraints
- Live before/after demo: vague prompt ('write about marketing') vs specific prompt ('write a 200-word Instagram caption for a Raipur bakery's Diwali sale, warm tone, include one CTA')

2.2 Role / Context Setting

- Show how to open with a role: 'You are a senior copywriter for a D2C skincare brand...'
- Explain when role-setting genuinely changes output vs when it's just flavor text

2.3 Iterative Refinement

- Teach learners to treat the first response as a draft, not a final answer
- Demo 3 follow-up refinement patterns: 'make it shorter', 'make it sound more [X]', 'give me 3 alternate versions'

2.4 Common Beginner Mistakes

- Being too vague and blaming Claude for a generic answer
- Not giving format instructions (length, tone, structure) and getting the wrong shape of answer
- Treating one response as final instead of iterating
- Asking multiple unrelated questions in one prompt

HANDS-ON PROJECT: Prompt Rewrite Challenge

Use case: A small business owner needs 3 pieces of content but only knows how to write vague requests — the learner fixes this.

Tools needed: claude.ai

Build steps:

1. Take 3 deliberately vague prompts provided by the instructor (e.g., 'write something for my shop')
2. Rewrite each using the Task + Context + Format + Constraints formula
3. Run both the vague and rewritten versions in Claude, side by side
4. Refine the best output twice using iterative follow-ups (shorten, change tone)

Deliverable to grade: A doc showing: vague prompt → rewritten prompt → before/after output → 2 refinement rounds

MODULE 3: CORE FEATURES | Week 2 · 120 min

Objective: Learner can organize work using **Projects**, generate a real document/code via **Artifacts**, upload and analyze a file, and use memory correctly.

Tools Needed	claude.ai (free or Pro), one sample PDF/CSV file
Prerequisite	Module 2

Lesson-by-Lesson Teaching Points

3.1 Projects

- Show how to create a Project, add custom instructions, and upload a knowledge base (e.g., brand guidelines)
- Explain persistent context: every chat inside a Project remembers the uploaded files and instructions

3.2 Artifacts

- Demo generating a document, then a simple webpage/code snippet — show live preview and edit-in-place
- Teach when Artifacts trigger automatically vs when to explicitly ask for a downloadable file

3.3 File Uploads

- Upload a PDF and ask Claude to summarize it
- Upload a CSV/XLSX and ask for a quick analysis or chart
- Upload an image and ask Claude to describe/extract text from it

3.4 Memory / Search Past Chats

- Show the memory settings toggle and explain what gets remembered vs what doesn't
- Demo asking Claude to recall something from an earlier conversation

3.5 Voice Mode & Mobile Basics

- Quick live demo of voice mode on mobile app
- Show cross-device sync (start on desktop, continue on mobile)

HANDS-ON PROJECT: Mini Business Knowledge Hub

Use case: A freelancer wants Claude to always know their brand voice and services without re-explaining every chat.

Tools needed: claude.ai Projects, 1 sample brand doc (or resume/portfolio), 1 CSV of sample data

Build steps:

1. Create a new Project named after a fictional or real business
2. Add custom instructions describing brand tone, target audience, and services
3. Upload one reference document (brand guide, resume, or product list) as project knowledge
4. Inside the project, ask Claude to draft one piece of content — check it matches the brand tone
5. In a separate chat, upload a CSV and ask Claude to generate 3 insights + one chart

Deliverable to grade: Project link/screenshot + the generated content + the CSV insight output

MODULE 4: INTERMEDIATE PROMPTING | Week 3 · 120 min

Objective: Learner can build a reusable prompt template using few-shot examples, structured reasoning, and XML tags — the same technique used in production prompt/skill systems.

Tools Needed	claude.ai
Prerequisite	Module 2 & 3

Lesson-by-Lesson Teaching Points

4.1 Few-Shot Examples

- Explain: showing Claude 2-3 examples of desired input → output is often more powerful than describing rules
- Demo: give Claude 2 example product descriptions in your exact style, then ask it to write a 3rd

4.2 Chain-of-Thought / Step-by-Step Reasoning

- Teach the phrase 'think step by step before answering' and why it improves accuracy on harder tasks
- Demo on a math/logic problem, comparing direct answer vs step-by-step answer

4.3 XML Tags for Structuring Prompts

- Show how to wrap instructions, examples, and data in tags like <instructions>, <example>, <data> so Claude reliably separates them
- Demo a prompt with and without tags on a multi-part task — show the tagged version is more consistent

4.4 Prompt Templates for Repeatable Tasks

- Introduce the idea of a 'master prompt' — reference your own SEO blog writer / GBP review reply as real working examples
- Teach learners to identify: what changes every time (variables) vs what stays fixed (rules)

4.5 Style Customization

- Show claude.ai's Style feature — create a custom style once, reuse across chats

HANDS-ON PROJECT: Build Your Own Reusable Content Template

Use case: A learner runs a recurring content task (e.g., weekly social captions) and wants Claude to produce consistent output every time without re-explaining rules.

Tools needed: claude.ai, a text file for the template

Build steps:

1. Pick one recurring content type (social caption, product description, email reply, review reply)
2. Write 2 hand-crafted example outputs in the exact desired style
3. Build a template prompt using XML tags: <role>, <rules>, <examples>, <input>
4. Add a 'think step by step' instruction before final output for quality-checking
5. Test the template on 3 different inputs and confirm consistent formatting/tone
6. Save the finalized prompt as a reusable Style or Project instruction

Deliverable to grade: The final template prompt (as text) + 3 test outputs generated from it

MODULE 5: PRACTICAL USE CASES | Week 3–4 · 150 min

Objective: Learner can apply Claude to 5 real work categories and produce one finished deliverable in each.

Tools Needed	claude.ai, sample XLSX/CSV, one image
Prerequisite	Modules 1–4

Lesson-by-Lesson Teaching Points

5.1 Writing & Content

- Demo: blog post, social captions, ad copy — same product, 3 different formats

5.2 Research & Summarization

- Demo: paste a long article/PDF and ask for a structured summary with key takeaways
- Show web-search-backed research for a current-events question

5.3 Data Analysis with Files

- Upload a sales CSV, ask Claude to find trends, generate a chart, and explain it in plain English

5.4 Image/Visual Tasks

- Demo asking Claude to analyze/describe an uploaded image, and separately generate a simple diagram via Artifacts

5.5 Coding Help for Non-Developers

- Demo asking Claude to build one small useful tool (e.g., a simple calculator or form) using Artifacts — no coding background needed to follow along

HANDS-ON PROJECT: 5-in-1 Use Case Portfolio

Use case: Learner builds a small portfolio proving they can apply Claude across writing, research, data, visuals, and basic tooling — usable later as freelance proof-of-skill.

Tools needed: claude.ai, 1 CSV, 1 image, 1 article/PDF

Build steps:

1. Writing: generate a blog post + 2 social captions for the same topic
2. Research: summarize a real article/PDF into 5 bullet takeaways
3. Data: upload a CSV, get 3 insights and one chart via Artifacts
4. Visual: upload an image and get a description; separately generate a simple diagram
5. Coding: ask Claude to build one small interactive tool (e.g., a tip calculator) as an Artifact

Deliverable to grade: One folder/doc with all 5 outputs labeled by category

MODULE 6: CLAUDE FOR WORK | Week 4 · 120 min

Objective: Learner understands the full ecosystem of Claude products beyond chat and can identify which tool fits which work task.

Tools Needed	claude.ai, Claude in Chrome extension (if available), Slack (optional)
Prerequisite	Modules 1–5

Lesson-by-Lesson Teaching Points

6.1 Claude in Chrome / Excel / PowerPoint

- Explain Claude in Chrome as a browsing agent that can act on webpages
- Explain Claude for Excel as a spreadsheet agent for formulas/analysis
- Explain Claude for PowerPoint as a slide-building agent
- NOTE for instructor: verify current feature availability/rollout via docs.claude.com and support.claude.com before teaching this live, since these ship and update frequently

6.2 Claude Cowork

- Explain Cowork as an agentic desktop app for multi-step knowledge work (non-developers), using tools together on real files

6.3 Claude Tag (Slack)

- Show tagging @Claude inside a Slack channel to delegate a task, if a Slack workspace is available for demo

6.4 Skills — What They Are

- Explain Skills as packaged, reusable instructions Claude follows for a specific recurring task — connect this directly to the templates built in Module 4
- Demo browsing/adding a skill if the environment supports it

HANDS-ON PROJECT: Tool-Fit Map for a Real Workflow

Use case: A small marketing team wants to know which Claude product to use for which weekly task, to avoid wasting time in the wrong tool.

Tools needed: claude.ai, docs.claude.com (for feature verification)

Build steps:

1. List 5 real recurring work tasks (e.g., inbox triage, slide deck, spreadsheet cleanup, content drafting, research)
2. For each task, research on docs.claude.com/support.claude.com which Claude product fits best
3. Build a simple 2-column table: Task → Best Claude Tool
4. Actually attempt ONE task end-to-end using the matched tool

Deliverable to grade: The Task → Tool table + proof (screenshot) of the one completed task

MODULE 7: DEVELOPER TRACK (ADVANCED) | Week 5–6 · 180 min

Objective: Learner can make a working Claude API call, understand tool use/MCP conceptually, and run one command in Claude Code.

Tools Needed	console.anthropic.com account + API key, Node.js or Python, terminal, VS Code (optional)
Prerequisite	Basic comfort with terminal; Modules 1–4

Lesson-by-Lesson Teaching Points

7.1 Claude API Basics

- Create an API key at console.anthropic.com
- Explain the /v1/messages endpoint: model, max_tokens, messages array
- Explain tokens and pricing at a high level — always verify current numbers on docs.claude.com since pricing/models update

7.2 System Prompts vs User Prompts

- Demo the difference: system prompt sets persistent behavior, user messages are the conversation turns

7.3 Tool Use / Function Calling

- Explain the concept: Claude can call a defined 'tool' (function) instead of just replying in text, e.g., a weather lookup function
- Walk through a simple example tool schema and how Claude decides to call it

7.4 MCP (Model Context Protocol)

- Explain MCP as a standard way to connect Claude to external data/tools (like Google Drive, Slack) without custom integration code each time

7.5 Claude Code

- Install Claude Code, run it in a sample folder, ask it to explain or fix a small piece of code

7.6 Building Simple AI-Powered Apps

- Show a minimal working script that sends one prompt to the API and prints the response

HANDS-ON PROJECT: First API Call + Mini Tool-Use Demo

Use case: A learner wants to move from 'using Claude' to 'building with Claude' — the foundational step for any automation or custom app.

Tools needed: Node.js or Python, console.anthropic.com API key, terminal

Build steps:

1. Create an API key in the Anthropic Console
2. Write a script that sends one prompt to /v1/messages and prints the text response
3. Add a system prompt that changes Claude's tone/persona and re-run
4. Define one simple tool (e.g., a function that returns a fixed 'weather' value) and get Claude to call it
5. Install Claude Code and ask it to explain what the script does

Deliverable to grade: The working script (file) + a screenshot of the terminal output for each of the 4 steps

MODULE 8: BEST PRACTICES & SAFETY | Week 6–7 · 90 min

Objective: Learner can identify hallucination risk, apply verification habits, and follow responsible-use principles when deploying Claude for real work or clients.

Tools Needed	claude.ai, docs.claude.com (Prompt Engineering guide)
Prerequisite	Modules 1–5 minimum

Lesson-by-Lesson Teaching Points

8.1 Prompt Engineering Principles (Official Guide)

- Walk through Anthropic's own core principles: be clear & direct, use examples, let Claude think, use structure — tie back to Module 4

8.2 Understanding Hallucination & Verification Habits

- Explain that any LLM can state something confidently-sounding but wrong, especially on facts/citations/numbers
- Teach the verification habit: cross-check factual claims, dates, and statistics before publishing client-facing content
- Demo asking Claude to cite sources or search the web when accuracy matters

8.3 Data Privacy — What Not to Share

- Teach learners never to paste client passwords, financial account numbers, or unnecessary personal data of third parties into any AI tool
- Explain that in professional/agency use, client consent and data-handling policy matter, not just tool settings

8.4 Responsible / Ethical Use

- Discuss: disclosing AI-assisted content where relevant, not generating deceptive content, respecting copyright when using AI for content

HANDS-ON PROJECT: Fact-Check & Privacy Audit

Use case: An agency is about to publish a client blog post drafted by Claude and needs a pre-publish safety/accuracy pass.

Tools needed: claude.ai with web search

Build steps:

1. Take any factual piece of content generated earlier in the course (Module 5 blog post works well)
2. List every factual claim, date, statistic, or named entity in it
3. Verify each one using web search — mark verified / incorrect / unverifiable
4. Rewrite or remove any claim that couldn't be verified
5. Write a 5-point data-privacy checklist you'll personally follow before pasting any client info into Claude

Deliverable to grade: The claim-verification table + the personal 5-point privacy checklist

MODULE 9: CAPSTONE PROJECT | Week 7–8 · Self-paced (1–2 weeks)

Objective: Learner builds one complete, real workflow end-to-end that combines skills from every prior module — proof of actual competence, not just course completion.

Tools Needed	Everything used across Modules 1–8, depending on chosen track
Prerequisite	All Modules 1–8

Capstone Track Options (learner picks ONE)

Track A — Content Engine (non-developer)

- Build a repeatable content pipeline: a reusable prompt template (Module 4) inside a Project (Module 3) that produces one content type on demand, verified for accuracy (Module 8)

Track B — Simple Automation (light technical)

- Build an automation that takes an input (e.g., form submission or CSV row) and produces an AI-generated output using the API (Module 7), e.g., auto-drafted review replies

Track C — AI-Powered Mini App (developer)

- Build a small working app (web page or script) that takes user input, calls the Claude API, and returns a useful result — e.g., a FAQ chatbot for a business

HANDS-ON PROJECT: Capstone: End-to-End Claude Workflow

Use case: Learner picks a real business problem (their own, a client's, or a provided case study) and solves it completely using Claude, demonstrating the full course.

Tools needed: Depends on chosen track — Projects/Artifacts (Track A), Claude + Sheets/n8n (Track B), API + code editor (Track C)

Build steps:

1. Define the problem in one sentence: who has this problem, and what should exist at the end
2. Choose Track A, B, or C based on comfort level
3. Design the workflow on paper first: inputs → Claude's role → output → who uses it
4. Build the working version using the appropriate tools from the course
5. Test with 3 different real inputs and fix any inconsistent outputs
6. Run the Module 8 verification/privacy checklist against the final output
7. Record a 3–5 min walkthrough (screen recording) explaining the workflow and demoing it live

Deliverable to grade: Working workflow/file + 3 test outputs + the walkthrough recording + one paragraph on what you'd improve next

Certification Criteria

- All 9 module deliverables submitted
- Capstone workflow runs successfully on at least 3 different real inputs
- Fact-check/privacy checklist applied to the capstone output
- Walkthrough recording clearly explains the 'why' behind each design choice, not just the 'what'

Appendix: Trusted Reference Sources

Use these as your fact-checking source before teaching any lesson — features, pricing, and model names update frequently.

- docs.claude.com — official product & API documentation
- support.claude.com — how-to guides, feature availability, plan limits
- anthropic.skilljar.com (Anthropic Academy) — free official courses, mirrors this syllabus closely
- Anthropic's Prompt Engineering guide (linked from docs.claude.com) — source for Module 4 & 8 content
- Anthropic Cookbook (GitHub) — real code patterns for Module 7 & 9
- console.anthropic.com — API keys, usage, billing for Module 7

Instructor note: because Claude's product lineup (new models, new integrations) ships frequently, re-verify Modules 1, 6, and 7 against the live docs each time before teaching a new cohort — don't rely on this document's specifics staying current indefinitely.