

# Claude Conversation: Projects & Comparisons

February 20, 2026

## Jim

explain claude projects

## Claude

Claude Projects is a feature in Claude.ai that lets you organize conversations around a specific context, goal, or workflow. Here's how it works:

What a Project does A Project gives Claude persistent context across multiple conversations within that project. Instead of re-explaining your codebase, role, preferences, or background every time you start a new chat, you set it once and it carries through.

Key components Project instructions — A system prompt you write that tells Claude how to behave, what your goals are, your preferred formats, constraints, etc. Claude reads this at the start of every conversation in the project. Project knowledge — Files and documents you upload (code files, docs, style guides, data, etc.) that Claude can reference in any conversation within the project. Conversation history — Chats are organized under the project so they stay grouped and easy to find.

Practical uses A coding project where you upload your repo files and define coding standards; a writing project with your style guide and existing drafts; a research project with reference documents Claude should always know about; a work assistant with your role, team context, and recurring tasks pre-loaded.

One important nuance The past chats memory tool (which lets Claude search prior conversations) is scoped per-project. If you're inside a project, it only searches conversations within that project. Outside any project, it only searches non-project conversations. So projects create a real separation of context.

Pro/Team/Enterprise distinction Projects are available on paid Claude.ai plans (Pro, Team, Enterprise). The amount of knowledge you can upload scales with the plan tier.

In short, Projects turn Claude from a stateless chat tool into something closer to a persistent, context-aware assistant tuned for a specific purpose. For your work at Syracuse — for example, maintaining your educational repos or your C++ tooling — a project per codebase would let you skip the setup context every session.

## Jim

how do I create and use a project

## Claude

Creating and using a project in Claude.ai is straightforward:

Creating a Project In the left sidebar, look for a "Projects" section or a "New Project" button. Click it and give your project a name and optional description. You're now inside the project.

Setting up Project Instructions Once inside the project, find the "Project instructions" area (usually visible when you're in the project view, not in a conversation). This is where you write your persistent system prompt — things like your role and context, preferred response format, and recurring constraints or rules.

Adding Knowledge Files Also in the project view, you can upload files — source code, docs, README files, style guides, anything Claude should always have access to. Supported types include text, code files, PDFs,

and more.

**Starting Conversations** Once the project is configured, just click "New Chat" from within the project. Every chat you start there will automatically have your instructions and knowledge loaded.

**Day-to-day use** Start a new chat in the project each session (or continue an old one). Claude will already know your context — no re-explaining needed. You can update instructions or add files to the project at any time from the project settings view.

**Practical tip for your work:** You could create a project for your C++ tools repo, upload key source files and your naming convention guide, and instruct Claude to always follow your style preferences and target CMake builds on Windows. Every chat would start already knowing all of that.

### **Jim**

Do ChatGPT and Gemini have similar capabilities

### **Claude**

Yes, both have similar concepts though they differ in implementation:

ChatGPT (OpenAI) Has "Projects" as well — very similar to Claude's. You get persistent instructions, file uploads, and conversations grouped together. Introduced relatively recently and closely mirrors what Claude offers. Also has a memory system where ChatGPT remembers things across all conversations (not just within a project), which is somewhat analogous to Claude's memory feature.

Gemini (Google) Has "Gems" — these are custom AI personas/assistants you configure with instructions and a specific purpose. Similar idea to projects but more focused on creating a reusable assistant than organizing a workflow. Gemini also integrates deeply with Google Workspace (Docs, Drive, Gmail), so if your context lives in Google's ecosystem it can pull from there natively — an advantage Claude and ChatGPT don't have out of the box.

How they compare overall The core idea — persistent instructions + uploaded context + organized conversations — exists across all three. The differences are more about ecosystem integration and depth. Gemini wins if you're deep in Google Workspace. ChatGPT has a large plugin/tool ecosystem and strong memory features. Claude tends to have a larger context window and strong performance on long documents and code, which makes its Projects particularly useful for technical work like yours. All three are converging on similar feature sets, so the gaps are narrowing quickly.