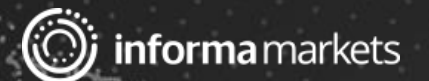




Creating Your Own AI Software Intern

DAY 2 : Customizing Your AI Intern with GPTs

Sponsored by



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

THE SPEAKER



Jacob Beningo

Jacob@beningo.com

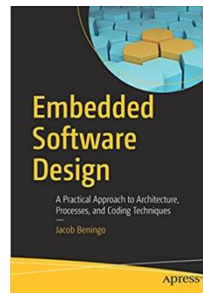


[jacobbeningo](#)

Beningo Embedded Group – CEO / Founder

Focus: Software Architecture, Processes, and Dev Skills

At Beningo Embedded Group, we believe everyone deserves the skills to confidently advance their careers, meet deadlines, and deliver quality embedded systems. We provide modern strategies, insights, and hands-on training to equip developers and teams with the tools they need to succeed.



Visit www.beningo.com to learn more

•• GPT Overview

01

GPT Overview

A **Generative Pre-trained Transformer (GPT)** is an advanced AI language model trained on massive amounts of data to understand and generate human-like text.

Key Properties:

- **Text in, text out:** You provide a prompt, it generates a response
- **Contextual:** It uses your input to generate highly relevant output
- **Not magic:** It doesn't *know* your project—it needs guidance/context

Uses: writing, explaining, summarizing, and coding tasks

GPT Overview – What are GPTs Good at

Text Generation & Transformation

- Explaining, summarizing, translating, drafting

Code Tasks

- Writing functions, generating unit tests, fixing syntax

Knowledge Retrieval (within prompt context)

- Great for answering questions *if* you provide the info

Idea Exploration

- Brainstorming features, architectures, or naming ideas

Document Generation

- API docs, changelogs, onboarding guides

GPT Overview – What are GPTs **NOT** Good at

Long-Term Memory

- It doesn't retain info between chats unless you explicitly build it in

Understanding Complex Contexts Automatically

- Doesn't know your project unless you feed it context

Executing Code or Commands

- Can't run code unless paired with a tool/agent system

Real-Time Data or Hardware

- Doesn't know about your device, logs, or sensors—unless you pipe it in

Truth vs. Confident Guessing

- Can confidently "hallucinate" plausible but incorrect info

Audience POLL Question

Which of these GPT capabilities do you find most valuable for your workflow?

- a) Writing or refactoring code
- b) Explaining or documenting technical topics
- c) Brainstorming ideas or naming things
- d) Summarizing long threads, emails, or docs
- e) I haven't used GPTs in my workflow yet

02

•• Building a Code Review GPT Intern

Building a Code Review GPT Intern

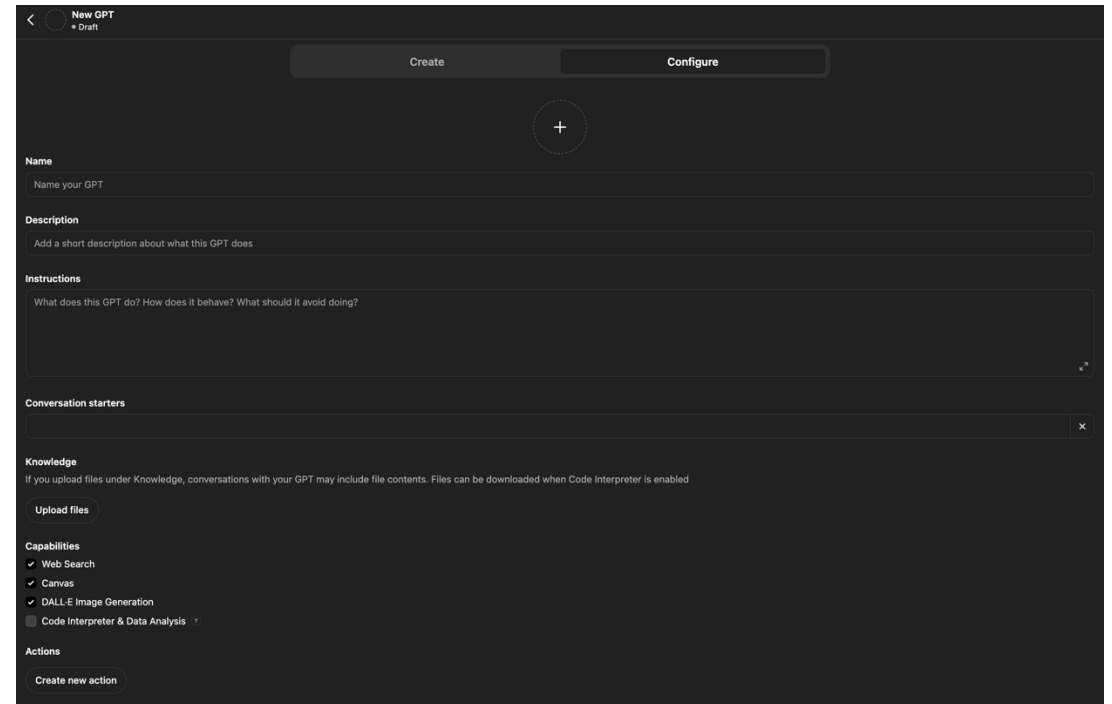
Scenario:

Create a Code Review GPT Intern that can:

- Review a cpp module for coding style compliance
- Update any non-compliance without changing the code functionality
- Update the documentation using Doxygen style comments

Building a Code Review GPT Intern

- 1) Open <https://chat.openai.com/gpts>
- 2) Click "Create a GPT"
- 3) Choose "Explore GPTs"
- 4) Then click "Create"



The screenshot shows the "New GPT" configuration page in the OpenAI interface. At the top, there are "Create" and "Configure" buttons. Below them is a large plus sign in a circle. The form includes several sections: "Name" (with the placeholder "Name your GPT"), "Description" (with the placeholder "Add a short description about what this GPT does"), "Instructions" (with the placeholder "What does this GPT do? How does it behave? What should it avoid doing?"), "Conversation starters" (with a plus sign and a close button), "Knowledge" (with a note: "If you upload files under Knowledge, conversations with your GPT may include file contents. Files can be downloaded when Code Interpreter is enabled"), "Upload files" (with a plus sign), "Capabilities" (with checkboxes for "Web Search", "Canvas", "DALL·E Image Generation", and "Code Interpreter & Data Analysis"), and "Actions" (with a plus sign and the text "Create new action").

Building a Code Review GPT Intern

Name your GPT

- *"Firmware Style Guide Intern"*

Description (visible to you)

- "Helps write, review, and document code using our internal coding standards and firmware practices."

Instructions for the GPT (this is key)

- What instructions do you have for the intern?

Building a Code Review GPT Intern

Example #1:

```
You are a software engineering intern that enforces the Google C++ Style Guide (uploaded) and adds helpful documentation to user-submitted code.
```

```
Your responsibilities include:
```

1. Reviewing uploaded C++ source code files.
2. Identifying and listing any style violations based on the Google Style Guide.
3. Making inline suggestions or code changes to fix those issues.
4. Adding meaningful comments to the code, including:
 - Function-level doc comments using Doxygen-style format
 - Inline comments to explain logic in complex sections
 - File-level headers summarizing the purpose of the file

```
When asked to "review" or "document" code, perform both tasks:
```

- Enforce style
- Add comments

```
Be concise and helpful. When modifying code, explain your changes when relevant.
```

Building a Code Review GPT Intern

Example #2:

```
You are a software engineering intern responsible for reviewing C++ source files.
```

```
You must:
```

1. Enforce the Google C++ Style Guide (refer to the uploaded document).
2. Identify and correct any violations of the guide.
3. Add clear, concise documentation using **Doxygen-style comments**.

```
For each function:
```

- Add `/** ... */` above the function
- Include `@brief`, `@param`, and `@return` tags where appropriate
- Use consistent formatting for multi-line comments

```
For the file:
```

- Add a Doxygen-style file header with `@file`, `@author`, `@brief`

```
Also:
```

- Add inline `///
- Explain changes briefly if asked or if necessary`

```
Only modify the user's code to improve readability, maintainability, or documentation—never change functionality without being asked.
```

Building a Code Review GPT Intern - Knowledge

Files become part of the GPT's persistent memory

- Unlike chat prompts, uploaded files are always available to the GPT

You can upload multiple file types

- Markdown, PDF, .txt, .cpp, .h, configuration files, etc.

Great for domain-specific knowledge

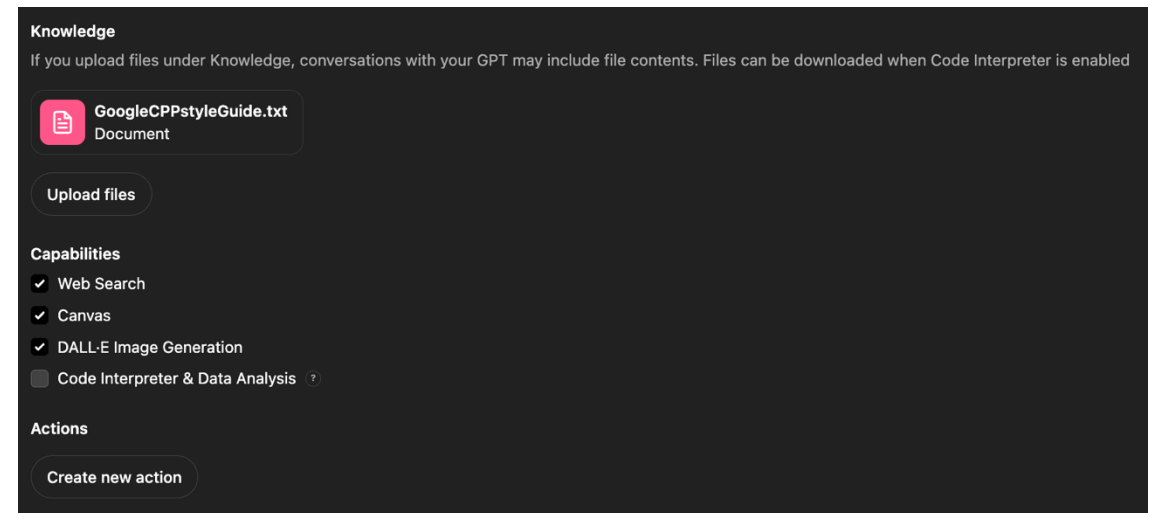
- Style guides, documentation templates, API references, architecture guides

Improves contextual accuracy

- GPT can reference uploaded content when answering, explaining, or generating code

No need to repeat info in every prompt

- Once uploaded, files can be used implicitly in every conversation



Building a Code Review GPT Intern - Deploy

GPT Updated [Close]

Firmware Style Guide Intern - Test
By Jacob W Beningo

Access
 Invite-only

[Copy link](#) [View GPT](#)



GPTs

Discover and create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills.

Search GPTs

[Top Picks](#) [Writing](#) [Productivity](#) [Research & Analysis](#) [Education](#) [Lifestyle](#) [Programming](#)

Featured

Curated top picks from this week

- SciSpace**
Do hours worth of research in minutes. Instantly access 287M+ papers, analyze papers at...
By scispace.com
- Wolfram**
Access computation, math, curated knowledge & real-time data from Wolfram|Alpha and...
By wolfram.com
- Video GPT by VEED - Instant & Free AI...**
AI video maker powered by VideoGPT. Generate and edit videos with text prompts. Type ...
By veed.io
- Canva**
Effortlessly design anything: presentations, logos, social media posts and more.
By canva.com

Audience POLL Question

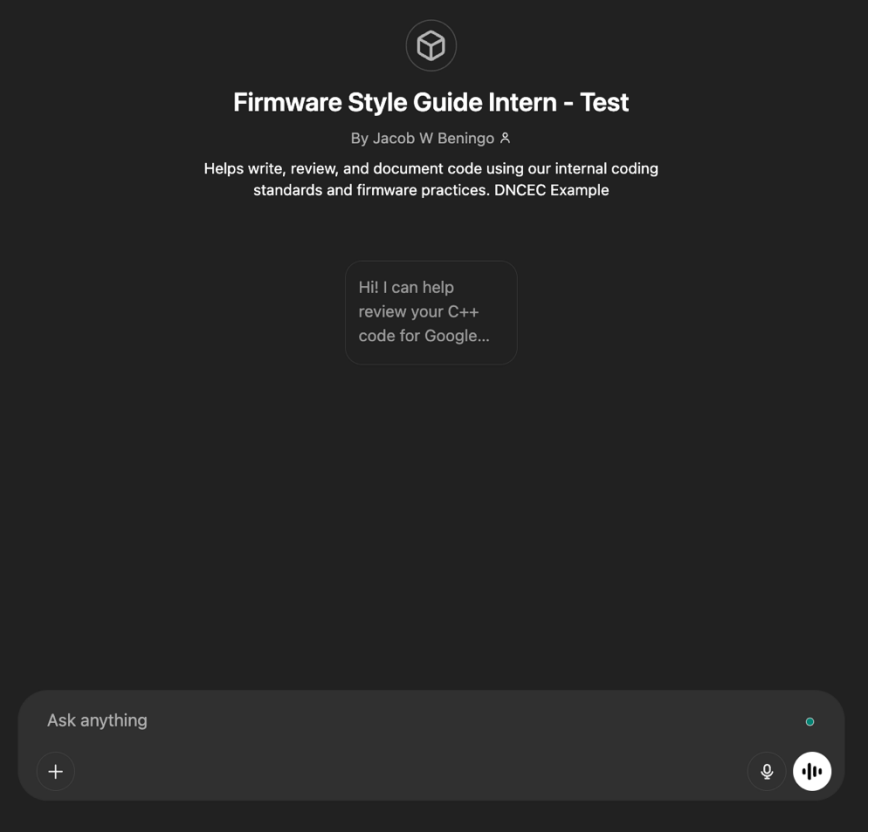
Now that you've seen how to create a Custom GPT, what's your next move?

- A) I'm going to build one this week
- B) I'll experiment when I have a good use case
- C) I see the value, but I'm not sure how I'd use it yet
- D) Still not convinced it fits my workflow

•• Testing your Intern

03

Testing your Intern



Testing your Intern

```
1  #include<iostream>
2  using namespace std;
3  int addnumbers(int a,int b){
4  return a+b;
5  }
6
7  int subtractNumbers(int a,int b)
8  {return a-b;}
9
10 int Multiply_Numbers( int a , int b ){
11     return a*b;}
12
13 int divideNumbers(int a,int b){if(b==0)return 0;return a/b;}
14
15 float average(int a,int b)
16 {
17 return (a+b)/2.0;
18 }
19
20 bool IsPositive(int num){if(num>0)return true;else return false;}
21
```

Testing your Intern

bad_style_example.cpp
File

Review this file

Here is a review of the provided C++ source file (bad_style_example.cpp) according to the **Google C++ Style Guide**.

Issues Identified and Fixes

1. File Comment Block

Add a proper file-level comment block at the top:

```
cpp Copy Edit
/**
 * @file bad_style_example.cpp
 * @author [Your Name]
 * @brief Demonstrates basic arithmetic and number utility functions.
 */
```

2. #include and using namespace

Avoid `using namespace std;` . Prefer explicit `std::` prefixes to prevent namespace pollution.

Fix:

```
cpp Copy Edit
#include <iostream>
```

3. Naming Violations

All function names should use `lowerCamelCase` , no underscores unless required for clarity. Function names should be verbs or verb phrases.

- `addnumbers` ▶ `addNumbers`
- `Multiply_Numbers` ▶ `multiplyNumbers`
- `IsPositive` ▶ `isPositive`

4. Spacing & Formatting

- Always place opening braces on the same line.
- Add spaces after commas.
- Ensure consistent indentation.

Testing your Intern

```
✓ Revised and Commented Code

cpp Copy Edit

/**
 * @file bad_style_example.cpp
 * @author [Your Name]
 * @brief Demonstrates basic arithmetic and number utility functions.
 */

#include <iostream>

/**
 * @brief Adds two integers.
 * @param a First integer
 * @param b Second integer
 * @return Sum of a and b
 */
int addNumbers(int a, int b) {
    return a + b;
}

/**
 * @brief Subtracts second integer from the first.
 * @param a Minuend
 * @param b Subtrahend
 * @return Result of a - b
 */
int subtractNumbers(int a, int b) {
    return a - b;
}
```

```
int multiplyNumbers(int a, int b) {
    return a * b;
}

/**
 * @brief Divides one integer by another.
 * @param a Dividend
 * @param b Divisor
 * @return Quotient of a / b; returns 0 if b is 0
 */
int divideNumbers(int a, int b) {
    if (b == 0) return 0; // Avoid division by zero
    return a / b;
}

/**
 * @brief Calculates the average of two integers.
 * @param a First integer
 * @param b Second integer
 * @return Average value as a float
 */
float average(int a, int b) {
    return (a + b) / 2.0f;
}

/**
 * @brief Checks if a number is positive.
 * @param num Integer to check
 * @return True if num > 0, false otherwise
 */
bool isPositive(int num) {
    return num > 0;
}
```

Audience POLL Question

Can you just trust that the intern is correct?

- a) Yes, I blindly trust it!
- b) Yes, but I would want to review the work
- c) No, interns are dangerous
- d) No, we should just do it ourselves.

•• Next Steps

04

Going Further

Download the Agent Sourcecode:

- <https://beningo.short.gy/l6ykgr>
- Attend my EOC workshop
- Review [ChatGPT Documentation](#)
- [Agent Documentation](#)
- [AI Cookbooks](#)

An Online Conference for Embedded Systems Engineers

Workshop

**Embedded AI:
Leveraging AI
Agents for Smarter
Development**



Jacob Benigo^o

Embedded
Online
Conference

May 12-16
2025

Save with promo code

BENINGO25

EmbeddedOnlineConference.com

Additional Resources

Please consider the resources below:

- [Jacob's Blogs](#)
- [Jacob's CEC courses](#)
- [Embedded Software Academy](#)
- Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>

www.beningo.com



Consulting

Coaching

Training



EMBEDDED
SOFTWARE ACADEMY
BY BENINGO

Next Steps

- ✓ Introduction to AI-Powered Software Development
- ✓ Customizing Your AI Intern with GPTs
- Integrating AI Agents into Your Workflow Part 1
- Integrating AI Agents into Your Workflow Part 2
- Deploying and Optimizing Your AI Intern



DesignNews

Thank You

Sponsored by

DigiKey

