Study Preparation

- Assign participant to a group
- Copy the log template
- Add participant to the repo right before the study
    - P## repo, click Settings→Collaboratorators and teams, click Add people, enter the user's GitHub id, click Write access, then click Add [GitHub id] to this repository.
- Put protocol and experiment log on separate display.
- Put your devices into focus mode
- If in-person connect laptop to the screen for demo

Study Intro

Hello, thank you for agreeing to participate in this study. My name is [name] and I'm a PhD student at [university]. From this study, we're trying to understand the usefulness of AI-based developer assistant tools. We are specifically interested in how effectively our AI-based tool can assist developers in understanding unfamiliar code.

Environmental Setup

We will be working in GitHub CodeSpaces for this study. CodeSpaces provides Visual Studio Code and a development environment that is accessible via a web browser.

Can you check your email for an invitation from GitHub to join our experiment repo? Accepting will give you access to the study repo for the study today. Once you join the repository, please do not open any file yet.

If haven't signed the consent form yet

> Before we begin, I'd like to have you sign this consent form if you haven't done so already. Let me know if you have any questions. You may also notice there's a question about recording you and your screen - please let me know if you are okay with that or not.

Enable participant screen-sharing in Zoom.

If the study is in person

> Please open zoom and start a new meeting. And please share your entire screen, not just a window. You can turn off your video.

Then, please start recording by clicking record and record to the cloud. Then, please enable the captioning by clicking show captions and choosing English.
We will ask you to provide the link to the cloud recording at the end of the study.

If the study is over zoom

Ready to begin? Let's turn off our cameras and I'll start recording. Is that ok?

Turn off all cameras,
turn on screen-share,
turn on captioning,
start cloud recording

Now, we can go to CodeSpaces. Now click the green Code button, then the CodeSpaces tab, then click the green Create CodeSpace button. Now the CodeSpace is starting.

- No Safari
- If using Chrome, they need to NOT use Incognito mode
- If using Firefox in strict mode (not the default), CodeSpaces won't load. Click the shield and flip the switch to disable strict mode for CodeSpaces if needed.

Great! We're in the Visual Studio Code IDE. All necessary extensions are being installed now.

If color theme is light

In the meantime, for your best experience, let us change the theme. Please type command/Ctrl + shift + p (F1 if firefox), type theme in the search box, and choose Github dark default.

In the meantime, we actually have a brief survey to understand your general information processing style and experience in Python programming. Could you submit your answers to the survey? I'm pasting the URL into the chat. Please let me know when you're done.

Now, let's go back to our CodeSpace. Just to double check everything is installed, please click the Extensions tab in the sidebar (second from the bottom, looks like four squares).

If other extensions installed

Please click the options button (with three dots) and click "disable all installed extensions". Now, please click AI for code understanding and live preview and enable them. And please reload the IDE by refreshing it.

Task Description

In this study session, we will be working on two Python programming tasks using two different approaches for program understanding. Before each task, I will explain the approach you will

use. The purpose of this is not to test your programming skills; we want to understand how you search, and what information you use in completing the tasks.

To get started, I will show you an example task.
From your experiment repo, please open the Task 0: task description link in a new tab.
This is an example task description for you to familiarize yourself with the experiment setting, and you do not need to actually solve this task. The task description includes a high-level description of the task, starting and goal outputs of the program, and a list of sub-tasks.
Please take a look and let me know when you're done reading.

For each task, your objective is to modify the given program to meet our requirements. To do so, you will need to understand the starting source code and the library used for the code. You will have 20 minutes to work on each task. We designed the tasks to be difficult without the expectation of anyone finishing them all, so just complete as many as you can in order. In the real tasks, you will have multiple sub-tasks. We highly recommend following the order of the sub-tasks as they may have dependencies. However, if you prefer to work on tasks out of order, please let me know when you start a new sub-task.

For your tasks, we provide the starting source code that is slightly different from what we want to implement. The code described in the task description can be found in your Codespace. Can you go back to Codespace and open pandas_demo.py?
The ideal solution for this example task is to change line 10 to df[A] + df[B] instead of sum.

Once you are satisfied with your modified program, let me know, and then I will ask you to answer questions to understand your level of confidence and understanding.

After you have finished the study, please do not talk about the tasks or libraries with potential future participants until we have completed the study for everyone.

Do you have any questions so far?

Task 1

LLM
For this task, we will use an AI plugin to get information you need for program comprehension. This extension will provide AI-generated information to assist your program understanding.

Before we start the task, I'll show you how to use the tool.
If participant is on zoom:
Could you stop sharing your screen for a second? I will share my screen for the demo.

This demo video uses the pandas_demo.py file to demonstrate all the features provided by the tool. While you watch the demo, I will explain each feature to you. Please feel free to stop me wherever you want to read the text or have questions.

To trigger the tool, you can highlight a line in the code and click on “AI Explanation” on the bottom blue bar or use “alt/option + a” shortcut. On the left tab, you will see a summary description of the highlighted code.

You can click any of the three buttons (API, Concept, Usage) at the bottom to trigger the tool further, as many times as you want.
The API button will give you more detailed explanations about the API calls used in the code.
The Concept button will provide you with domain-specific concepts that you might need to learn to fully understand the highlighted code.
The Usage button will provide you with a code example involving the API calls used in the highlighted code.

You can ask the AI model questions directly using the input box. For example, you can ask “Where do I do addition?” and click the “Ask” button or type enter. You can also ask a question about a specific part of the code by highlighting the code and asking a question.

To better understand the code, you can trigger the tool as many times as you want, and for every block or line.
For example, you can also trigger the tool by selecting the entire source code.
Or, to understand the code at a finer granularity, you can trigger the tool for each block of the code.
You can use the Embed button when you want to keep the information. The AI generated response will be added to the line before the highlighted code or the line before the cursor.

You can also ask a follow up question to probe the AI model further when you do not select any part from your code.

If you do not need to ask a follow-up question, you can simply click the clear all button or trigger the tool with new code context.

Finally, you can collapse or expand the previous information by clicking on the arrow icon next to the Embed button.

Now, can you open pandas_demo.py and try using the tool once by yourself?
If API key request prompted:
[provide api key]

> We need to input an API key to use the extension. This API key will only be used for this study, and will be removed after this study. However, please do not share it with anyone. Please copy and paste the key in the chat, and paste the key.

Please adjust the size of the panel to display everything.

Please let me know if you have any questions or if you feel like you are ready to start working on the tasks.

One important point to note is that the AI model we are using, which is GPT-4, is currently in a limited beta setting, and as such, the response time may not always be reliable. Due to our experience with slow response times, we have switched our plugin to use GPT-3.5 instead of GPT-4. However, if you prefer to use GPT-4, we can also accommodate that. Are you okay with GPT-3.5?

If they want to use GPT-4:

> Please click the settings icon on the bottom left, and click Extensions, AI for code understanding. Then, under GPT: Chat Model, replace GPT-3.5-turbo with GPT-4.

During the study, please let me know if you feel like the extension cannot provide information you need - we will allow you to use Google for that specific information.

Do you have any questions about the extension?

Now, let's start the task.

bokeh

The first task is data visualization.

To make sure that we can run the code correctly in this setting, let us try to run it before we start the task.
Please open the terminal (by typing "Ctrl+`(Grave accent (backtick))" or click the "x" icon on the bottom blue bar), and do "python task1.py".
When you run bokeh_task.py, it will generate task1.html, which is the output of your code.

To view the output of this code, please right click on task1.py in your explorer, click show preview, click the green button in your popup, and click task1.html.

Now we are ready to work on the task. Please go to our experiment repo, and open task 1 task description in a new tab.
Please read the description and let me know when you are ready to work on the task.
When you finish each subtask, please record the time at the end of your code.

Start the 20 minute task timer.
Mark the task start time

Stop and reset the 20 minute task timer.
Mark the task end time

Please save, close your tabs, stage, commit, and push to the repo using the user interface. The commit message does not matter: just use something like “update.” Use ...->Push to push to the repo.

We have some questions for you about the task that you just worked on. Please go to our experiment repository, and open task 1 questionnaire.
Please do not consult the original code and let me know when you’re done with the survey.

Task 2

Search engine
For this task, we will use search engines to retrieve any information you need for program comprehension. Please feel free to use web search whenever you need, and visit any web pages, but please do not use AI-based tools such as ChatGPT or Bing Chat.

Open3d
In this task, you will work on a task using a python 3d rendering library Open3d.

Now, please open the terminal by typing “Ctrl+`(Grave accent (backtick))” or click the “x” icon on the bottom blue bar, and type python3 task2.py
Whenever you run this code, it will open a new window. You might need to allow pop ups (check the right corner of your address bar).

Now we are ready to work on the task. Please go to our experiment repo, and open task 2 task description in a new tab.
Please read the description and let me know when you are ready to work on the task.
When you finish each subtask, please record the time at the end of your code.

Start the 20 minute task timer.
Mark the task start time

Stop and reset the 20 minute task timer.
Mark the task end time

Please save, close your tabs, stage, commit, and push to the repo using the user interface. The commit message does not matter: just use something like “update.” Use ...->Push to push to the repo.

We have some questions for you about the task that you just worked on. Please go to our experiment repository, and open task 2 questionnaire.
Please do not consult the original code and let me know when you're done with the survey.

Post Survey

Now, we are done with the tasks.
Please close Visual Studio Code now and we can stop screen sharing and recording now.
We have a survey about the two approaches you used today. Please go back to our experiment repo, and open the post-study survey,

We are particularly interested in your thoughts on the usability and usefulness of this AI plug-in. However, please do not reflect too much on the response time, as we expect it to be improved in the near future. If you have any questions or concerns, please don't hesitate to ask me. I will remain available while you complete the survey.

Great! We're all set. Thanks for filling out the survey.

If the study is in person

Please go to https://cmu.zoom.us/recording and click on the share icon, and copy the shareable link and the passcode, and send me to my email.

Within a week you'll receive the Amazon gift card in the email you provided. If you don't get it for some reason, please email me – my email is on the invitation.
Do you have any questions before we go?

End of study

1. Archive the data: Upload the Zoom recording & transcript to Google Drive. Name each file by participant id.
2. Remove the participant from CodeSpaces
    a. Go to: [github repo] → People→Outside Collaborators
    b. Click the gear beside the participant → Remove from all repositories
    c. Click Remove outside collaborators
3. Issue payment to the participant

Troubleshooting

- Check if the extension is installed. You should be able to see a square icon underneath the github icon.If you don't see it, you have to install the extension, run the following command in your terminal:
  - `code --install-extension .vsix`
- If you get an error (`ImportError: libGL.so.1: cannot open shared object file)` while running open3d task, run the following command in your terminal:
  - `sudo apt-get update && sudo apt-get install libgl1`
- If you get Not found errors, run the following command in your terminal to install Python packages:
  - `pip3 install -r requirements.txt`
- If you cannot see the output image, check if you have installed "Live server", and "Live preview" extensions. Check if you're using the pre-release version of "Live preview".
- If the LLM extension prompts you to input an API key, input the following (It is my API key so please do not share with others).
  - API Key