

Understand, develop and test code



How you can do it

This three-step guide presents one possible approach to using generative AI to have a piece of code explained and then test the explanation in practice.

1. Choose and examine the code

Choose a specific piece of code that you would like to understand. It could be your own code, an example from class, or code from a tutorial or academic source.

First, try to work through the code yourself. Consider:

- What do you think the purpose of the code is?
- What input does it receive?
- What output do you expect?
- Which lines or constructs are unclear?
- How does the code's actual result differ from what you expected?

Write down your preliminary understanding and your questions. This makes it easier

to use the chatbot to investigate specific parts rather than leaving the entire interpretation to it.

2. Get the code explained

Give the chatbot the necessary context, including the programming language, relevant libraries, and the code's intended purpose. Ask it to tailor the explanation to your level of expertise.

For example, you can ask the chatbot to:

- summarise the code's function
- explain specific lines or constructs
- walk through the logic step by step
- show how values change along the way
- explain the code's expected input and output
- connect the code to relevant programming concepts
- point out assumptions or possible sources of error
- provide a small example that illustrates the same function

Use follow-up questions if parts of the explanation remain unclear. Consider asking for a different type of explanation rather than simply a longer version of the first one.

3. Test and verify the explanation

Investigate whether the chatbot's explanation matches what the code actually does. Run the code in a suitable environment and test it with different inputs.

You can, for example:

- predict the result before running the code
- track the values of key variables
- change one part at a time and observe the result
- insert print statements or use a debugging tool
- compare the explanation with official documentation
- explain the code's function in your own words

If the explanation and the code's actual behaviour do not match, investigate the difference. This may be due to errors in the explanation, missing context, or differences between versions and libraries.



CENTRE FOR EDUCATIONAL DEVELOPMENT
AARHUS UNIVERSITY

AU STUDYPEDIA