IBM watsonx Code Assistant – Lab 2024 (C++)

watsonx Code Assistant for General Programming

Utilizing an intuitive natural language interface, developers

can quickly resolve issues, access documentation, ask for

code explanations, generate unit tests and more

Use cases

V1.0

# Table of Contents

[A. Use Case1: Conversational Chat 3](#_Toc182583126)

[B. Use Case 2: Code Explanation 9](#_Toc182583127)

[C. Use Case 3: Generate Documentation 10](#_Toc182583128)

[D. Use Case 4: Generate Unit Tests 12](#_Toc182583129)

# A. Use Case1: Conversational Chat

Leverage Gen AI in a more organic, conversational way with an embedded chat window inside your editor. It has the ability to gain deeper context with each user response, allowing you to get a more accurate response to your questions.

Go to WCA chat window and start by writing any of the following prompts to generate the corresponding code:

1. Create a JavaScript function that takes an array of numbers as input

and returns the sum of all positive numbers in the array.

1. Generate a Java method that calculates the average temperature

from an array of daily temperature readings.

1. Write a Python class representing a bank account with methods for

deposit, withdrawal, and checking the balance.

1. Develop a JavaScript function that analyses a collection of customer

orders and returns a summary report with the total number of orders,

the average order value, and the most frequently ordered product.

Let’s start now to create a fresh new application in C++, to manage a hypothetical personal banking account.

**Conversational Chat Step 1**

Let’s create the BankingAccount class starting to create the header file:

Write a C++ class representing a bank account with methods for deposit, withdrawal, and checking the balance. Write only the header file without method implementation.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Now you can copy the generated code and put it into a BankAccount.h file

**Conversational Chat Step 2**

Let’s write now the header file to be included in sources in order to use the new generated class:

write code to fully implement the BankAccount class starting from the generated header.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

The generated code contains the duplicate of the constructor, and we can fix asking not to generate it

drop out the contructor from the previous code.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Now you can copy the generated code and put it into a BankAccount.cpp file

**Conversational Chat Step 3**

Now let’s write an end-user application that make use of the defined class to allow the final user to interact with his banking account:

now build a simple C++ application able to use BankAccount class to allow user to work with their banking account.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

You are not satisfied with the application because you want it to be more interactive, so let’s change its behavior:

now add conditions to the application letting user to select which operation to do and to pass it values for data.

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

Get this code and put it into a bank\_account.cpp file.

**Conversational Chat Step 4**

Now that we are more satisfied with app behavior we can run it and test it, but first we need a Makefile to build it:

now create a Makefile to build the created application.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Put this code into a Makefile file in the same directory of the sources.

You can now build and run the application. From command line, in the application folder, type `make` and wait for compilation. Once ended you’ll have your `bank\_account` application ready to be launched.

Note: you must have installed a C++ compiler and Automake package to build the binaries.

# B. Use Case 2: Code Explanation

"Explain this" will give you an explanation of a code snippet's usage, parameters, and output.

Select a code or part of code in your VS Code IDE and ask the watsonx code assistant to explain the code

The below figure is for demonstration purpose. You can choose more complex code for code explanation and different touch points, for example the so called code-lens inside the sources

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

or the contextual menu either selecting portions of the code from the editor or directly from the project tree

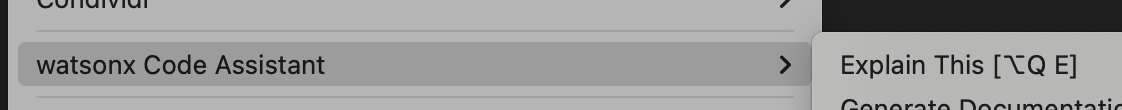


Immagine che contiene testo, schermata, software, Sito Web

Descrizione generata automaticamente

# C. Use Case 3: Generate Documentation

"Generate documentation" creates commented inline notation that will be used to inform other developers of a code snippet's purpose as well as define its parameters.

Select a code from the editor and write ‘Generate documentation’ in the chat window. You can also paste the code directly in the chat window. Alternatively, you can click on ‘Document’ present above the code.

Generate Documentation Example 1:

Here we have written a simple prompt to:

Generate documentation and add comments to the code

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Here we can see that WCA has generated the code documentation along with code comments explaining each step

Generate Documentation Example 2:

Click into the shortcut inside the source editor:

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

As we can see on the left side that WCA has generated the code documentation.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# D. Use Case 4: Generate Unit Tests

Unit test generation creates a full set of tests for the selected code snippet using the common libraries for the detected language.

Go to WCA chat window and start by writing any of the following prompts to generate the corresponding code:

You can select a file in the project explorer and select ‘Unit Test’ item in the contextual menu.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Alternatively, you can click on ‘Unit Test’ above the code to generate unit test for a specific piece of code, for example a single method.

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente