NoCom

*Helping developers read and understand code with interactive explanations.*

## Team HubDivers

Anthony Weaver, Ben Caras, Bryce Roy, Kempton Maillett, Obed Appiah

# Problem

Many beginner programmers struggle to read and understand code, which limits their ability to progress in learning programming languages and solving complex problems.

For example, novice programmers often find it challenging to understand the logic and structure of even simple code snippets. This confusion can stem from unfamiliar syntax, poorly commented code, or difficulty in grasping the relationships between different parts of the code. As a result, beginners frequently become discouraged and may abandon learning programming altogether.

Existing solutions such as online tutorials and coding platforms often focus on writing code from scratch or solving programming challenges, but they do not emphasize code reading skills. Some coding platforms provide examples, but they lack interactive tools that explain code line-by-line or break down concepts in a way that is accessible for beginners.

While current platforms help beginners practice coding, they don’t provide enough focus on teaching how to read and interpret existing code effectively. This gap leaves new programmers struggling to understand the foundational building blocks of coding, preventing them from developing strong problem-solving skills and confidence.

How can we design an interactive platform or program that teaches beginner programmers how to read and interpret code efficiently, providing real-time feedback and clear explanations to enhance their learning process?

Additional paragraphs may go into more detail about causes, covering how well/if the cause is addressed by prior solutions.

# Prior Solutions

## Gidget

Gidget is a game that both programmers and non-programmers can use to improve their code reading abilities. The game has players assist a robot while fixing defects in programs in a tier based difficulty format. For instance level 1 would be easy difficulty while level 10 would be hard difficulty. This fundamentally provides a strategy for improving concepts of computing in an informal manner. Some of the weaknesses for this game include individualized blocks of code that the user might have future issues understanding. While the game does provide a good grasp of the concepts, it does not account for any future more complex blocks of code a programmer might need to try and understand themselves.

Our app will be different from Gidget because our app uses code directly supplied by the user. Which will assist the user in understanding how different lines of code operate within an entire block. Our app will then link relative concepts using generative AI to further assist in getting a better understanding of how the block of code operates. This will provide a context based analysis for programmers to learn from using real world experiences.



## ChatGPT/Generative AI

ChatGPT or other generative AIs are chatbots that use machine learning or deep neural networks that have learned to understand and respond to human language; these generative AIs typically use large datasets to generate “new content” based on what it has learned. Generative AIs can be useful for helping someone create outlines for papers, codes, emails, and many other things as well as helping someone understand a topic in a faster, more productive way, or summing key points of articles, and texts. One of the things generative AI is useful for is helping people understand and debug code. While generative AI can be useful for testing and debugging code it attempts to help people understand the code itself and what each line's functionality, and semantics. Many people struggle with writing efficient and reusable code and many also struggle to understand what each line of code executes and how it functions. It can be easy to understand a few simple lines of code but as the files start to become larger and more complex it can become difficult for people to follow along and understand what the code is doing. This is where generative AI tends to fall off. It can help someone understand minimal easy to grasp segments of code but it doesn’t explain in detail large blocks of code. Also the user typically has to ask the AI to refine its response multiple times before the answer is acceptable.

Our app will build upon generative AI to give comments and easily understandable explanations for code snippets, but we will build it with references to specific resources for novice programmers. This includes having prebuilt scenarios for common concepts in coding, while going through them they will receive active feedback on their comprehension. These scenarios will be suggested alongside the interpretation to give people a more structured path in their understanding.



## Online tutorials

Online tutorials are the most available source for reading and interpreting code. They provide information and context for complex concepts that are hard to learn by just reading about them in textbooks. They are also a great place to see relevant examples of the concepts in practice. While these tutorials are good, they have two major deficiencies. They often lack relevant practice and often fail to motivate people to self-practice, which causes a lack of retention and a practical understanding of the concept. They also are often hit or miss on what the intended audience is some are meant for people who know nothing about programming, while others are meant for professionals who just need to focus on one concept at a time

NoCom will be different because it will be an interactive tool that actively supports a user's learning instead of the passive approach that tutorials support. This tool will address the main problem of tutorials by providing active feedback on how well you have comprehended a concept or piece of code, it will also increase the recency problem by using generative AI, the user will be able to input snippets of code then our program will be able to link them to relevant concepts and interpret the purpose of the code.



# 

# NoCom

NoCom is a web application designed to help beginner programmers improve their ability to read and understand code. Users can input code snippets into the platform and receive interpretation of code. It breaks down the code into smaller, digestible blocks and provides detailed explanations for each section, highlighting syntax, logic and structure. The platform also offers links to key programming concepts related to code, helping users deepen their understanding.

NoCom uses AI to adapt to explanations based on the user's skill level hence ensuring that beginners can gradually advance to more complex concepts.



*NoCom Task diagram*

NoCom focuses on helping beginner programmers understand code, especially when they struggle with reading and interpreting unfamiliar syntax. This platform will offer real time feedback and explanations for code input by users hence supporting their learning journey with interactive and personalized help. Unlike traditional tutorials that passively teach concepts, NoCom engages users actively by interpreting code and linking the concepts that power each block.

To support novice programmers, NoCom provides a feature that links the code to prebuilt scenarios of common coding concepts, giving users a chance to practice with real world examples.

An existing platform like Gidget teaches code reading through isolated blocks which limits users’ ability to understand complex code. NoCom improves on this by using real world code inputs by users, helping them understand how the code blocks work together while generative AI links code to relevant concepts for a better learning experience.

ChatGPT on the other hand helps beginners to debug and gives them basic explanations but it struggles with large complex codes sometimes. NoCom will focus on structured learning offering clearer explanations, references and active feedback on comprehension.  
Online tutorials are often passive and lack interactive real-time practice. NoCom addresses this by providing real time feedback allowing users to input code and receive immediate explanations and concept links, making learning more dynamic and effective.