

Students: Charlie Kake-White, Harrison Bland

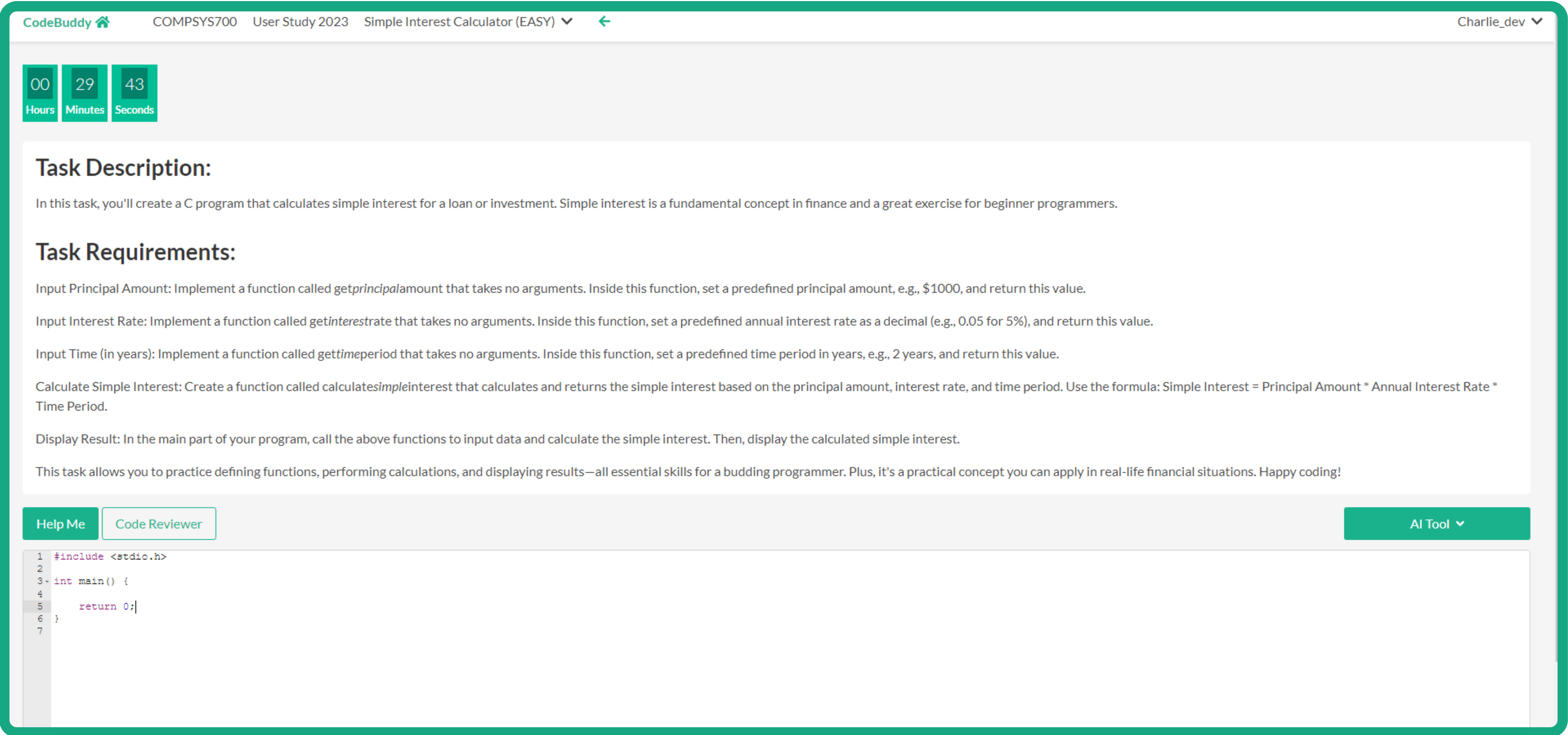
Supervisors: Assoc. Prof. Paul Denny, Prof. Andrew Luxton-Reilly

Introduction

Our project and its research looks to investigate the impact of integrating a Language Model (LLM)-powered AI tool into programming education, addressing the increasing demand for coding proficiency. By focusing on its impact with novice programmers, this study explores how the LLM-based approach influences code quality, problem solving skills, and understanding of programming concepts. Overall, our research aims to showcase the potential of LLMs in programming education, benefiting both educators and learners.

Our Tool

CodeBuddy LLM empowers students with three key features: Next Line Code Generation, "What's Next?" explanations, and the Code Feedback Generator. These features offer instant programming assistance, leveraging the GPT-3.5-turbo model from OpenAI via API calls. Unlike prior solutions, our tool simplifies user interaction by handling all prompting intricacies. Students can access timely support with a single click, ensuring a frustration-free and streamlined learning experience.



Prompting Strategies

During our experimentation phase, we initially employed diverse prompting strategies (left screenshot) that included providing extensive context, such as the student code, the model's role, a step-by-step logical process, and even full instructor solutions. These strategies demonstrated the model's remarkable capacity to generate tasks and complete solutions that catered to the needs of learners with varying levels of expertise. After numerous iterations and careful analysis, we ultimately settled on a refined prompting strategy, as illustrated in the right screenshot. This strategy simplified the input context to include just three key elements: the model's role, the task description, and the user's code. We found that this streamlined approach consistently yielded the most helpful and coherent responses from the model, aligning perfectly with our objective of enhancing the CodeBuddy experience for all learners.

Final Example Prompt (“What’s Next” Explanation):

You have been given the role of a educator who helps students when stuck on a given programming task. You have been given the task description above which gives you context for the programming task that the student is trying to complete. Based on the task description requirements, in one concise sentence, provide a helpful statement that the students can use to progress the implementation of the task. The student code is provided below. Examples of advice are listed below:

- In order to start the task, you should....
- There is a syntax error in the code, you should change it to...

NOTE: The response should be in purely natural language NEVER generate code

Qualitative Data



Results and Evaluation

We had four participants in our user study that were tasked with completing a pre-assessment evaluation before attempting 3 LLM-generated exercises. After attempting the exercises, participants were prompted with questions regarding the effectiveness of the AI Tool within the CodeBuddy interface. There were two metrics for quantitative analysis (for a total of 5 points):

- Accuracy of response (does the provided responses make sense given the context of the user code provided) - score out of 4
- No unnecessary code generation (do the responses contain generated code that shouldn't be present, e.g. multiple lines generated for hint code or generated code in feedback) - 1 for no unnecessary code, 0 otherwise.

Summary of Quantitative Data

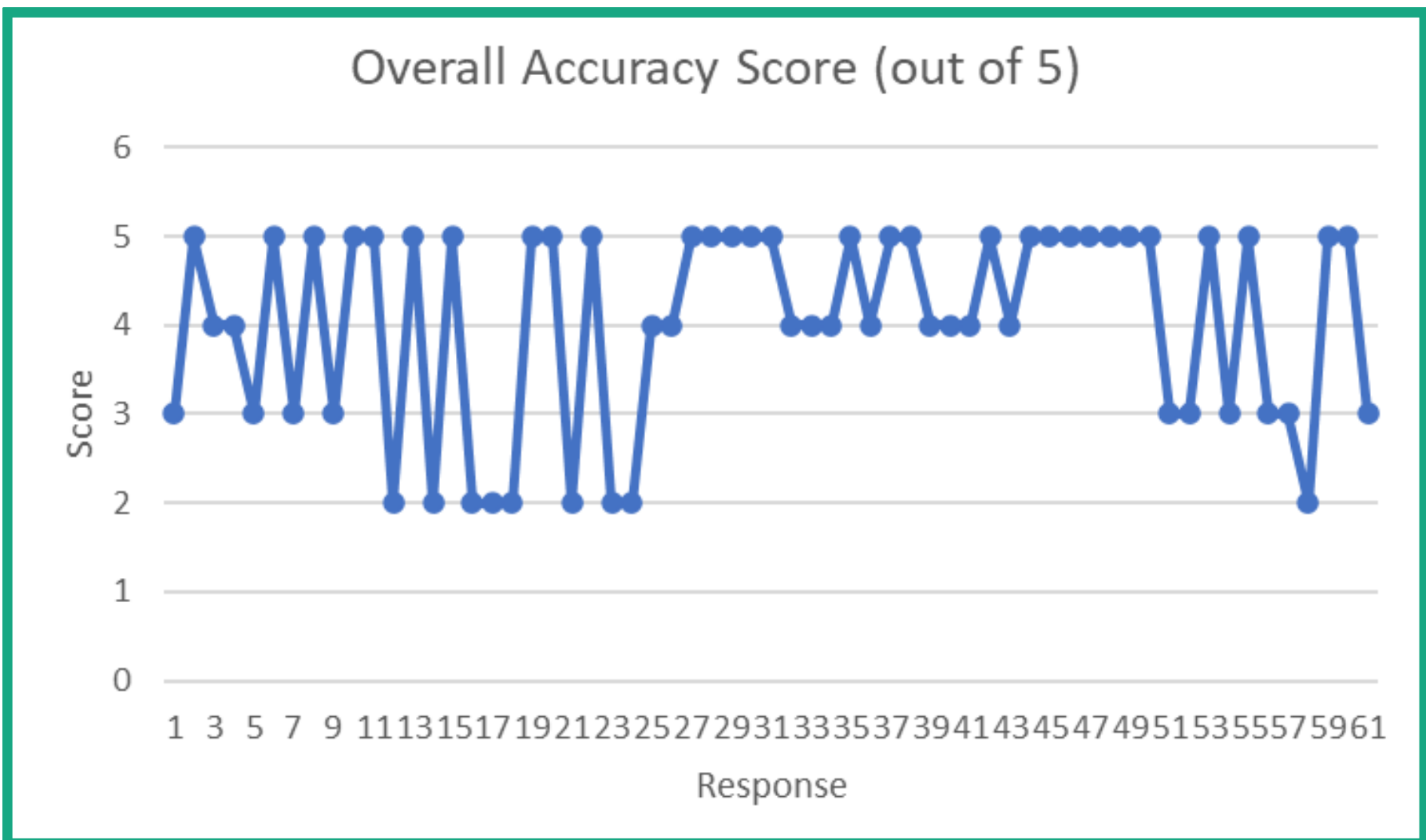
- Overall Response Accuracy: 80.65%
- Feedback Accuracy: 93.79%
- Hint Accuracy: 48.57%
- "What's Next" Accuracy: 80.97%

Feedback feature received the highest accuracy score, while the hint feature performed the worst, affecting the overall average negatively. Participants generally found the tool to be at least adequately useful, although the small sample size suggests the need for further user studies to draw more robust conclusions.

Conclusions

- This positive user perception suggests that AI-driven programming assistance has potential for enhancing the learning experience.
- The Feedback feature stood out with an impressive accuracy of 93.79%, showcasing its potential to assist users effectively.
- Hint feature accuracy needs to be addressed for improving the tool's accuracy

Quantitative Data



Anonymous Reviews

“The step by step feedback about each section of code and how and why to fix it. This made it easy to know if the AI got confused...”

“The generate feedback was the most useful aspect.”

“The hint provided me with built in syntax reference - helped me work with strings. I like how it gave you some info about what to do...”

Future Works

- Explore a larger variety of pretrained models
- Conduct long term studies to assess the impact of AI-guided learning
- Create some learning analytics to provide instructors with detailed insights into student performance and areas for improvement