AI Debugging Assistant: Enhancing Debugging Skills With Intelligent Guidance

Elizaveta Artser
JetBrains Research
Munich, Germany
elizaveta.artser@jetbrains.com

Daniil Karol
JetBrains Research
Berlin, Germany
daniil.karol@jetbrains.com

Anna Potriasaeva
JetBrains Research
Belgrade, Serbia
anna.potriasaeva@jetbrains.com

Aleksey Rostovskiy
JetBrains Research
Munich, Germany
aleksey.rostovskiy@jetbrains.com

Anastasiia Birillo
JetBrains Research
Belgrade, Serbia
anastasia.birillo@jetbrains.com

ABSTRACT

Debugging is an essential skill in programming education. Current debugging approaches lack interactivity and personalization for students. To address this gap, we introduce an *AI Debugging Assistant* integrated into JetBrains IDEs. The tool analyzes the students' errors in real-time and guides them through the debugging process by recommending breakpoints and explaining them on each step. This poster invites discussion on the effectiveness of the *AI Debugging Assistant* and its implications for programming education.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Human-centered computing** → *Interactive systems and tools.*

KEYWORDS

Intelligent Tutoring, Debugging, In-IDE Learning, Generative AI

ACM Reference Format:
Elizaveta Artser, Daniil Karol, Anna Potriasaeva, Aleksey Rostovskiy, and Anastasiia Birillo. 2025. AI Debugging Assistant: Enhancing Debugging Skills With Intelligent Guidance. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2025), June 27-July 2, 2025, Nijmegen, Netherlands.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3724389.3730777

1 INTRODUCTION

Debugging is an essential but often overlooked part of programming education. Limited guidance makes many beginners learn debugging through frustrating trial and error [4]. Existing teaching approaches often use pre-made exercises [3] or generic frameworks [2], which are not personalized and can be hard to apply in real programming tasks. Moreover, Integrated Development Environment's (IDE's) debuggers are widely used in industry programming, but they are rarely included in the education process or still remain unclear for students who decide to use them. This work introduces an *AI Debugging Assistant* built into an IDE to help students learn debugging with personalized and practical support.

2 AI DEBUGGING ASSISTANT

Tool overview. The *AI Debugging Assistant* integrates seamlessly into any in-IDE programming course [1], supporting students in developing debugging skills. The tool helps them place breakpoints and explains why students need to pay attention there. The assistant emphasizes providing a practical debugging experience rather than explaining the components of the debugging process, *e.g.* what breakpoints are or how to add a new one.

Usage pipeline. If the student's code fails to pass the task's tests, the *AI Debugging Assistant* offers the student to initiate a guided debugging session via a pop-up notification. Upon starting the session, the tool automatically places recommended breakpoints within the student's code. During execution, when the program pauses at a *particular* breakpoint, the assistant provides a context-specific hint, *e.g.* which variable value to examine. As this debugging process occurs entirely inside the IDE, it closely mirrors the standard debugging workflow commonly used in professional development. At the same time, it teaches students the necessary skills.

Technical details. Breakpoint recommendation involves two key steps: (1) error fixing and (2) breakpoint suggestion with explanations. To improve accuracy in the first step, we utilize advanced ML techniques, such as Retrieval-Augmented Generation (RAG) combined with LLMs. For the second step, we leverage static analysis heuristics and program slicing within the IDE to generate breakpoint suggestions. Explanations for the suggested breakpoints are produced using a lightweight language model.

REFERENCES

[1] Anastasiia Birillo, Mariia Tigina, Zarina Kurbatova, Anna Potriasaeva, Ilya Vlasov, Valerii Ovchinnikov, and Igor Gerasimov. 2024. Bridging Education and Development: IDEs as Interactive Learning Platforms. In *Proceedings of the 1st ACM/IEEE Workshop on Integrated Development Environments.* 53–58.

[2] Chen Li, Emily Chan, Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2019. Towards a framework for teaching debugging. In *Proceedings of the Twenty-First Australasian Computing Education Conference.* 79–86.

[3] Andrew Luxton-Reilly, Emma McMillan, Elizabeth Stevenson, Ewan Tempero, and Paul Denny. 2018. Ladebug: an online tool to help novice programmers improve their debugging skills. In *Proceedings of the 23rd annual acm conference on innovation and technology in computer science education.* 159–164.

[4] Tilman Michaeli and Ralf Romeike. 2019. Improving debugging skills in the classroom: The effects of teaching a systematic debugging process. In *Proceedings of the 14th workshop in primary and secondary computing education.* 1–7.

ITiCSE 2025, June 27-July 2, 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-1569-3/2025/06
https://doi.org/10.1145/3724389.3730777