Check for updates

Transformed by Transformers: Navigating the AI Coding Revolution for Computing Education

An ITiCSE Working Group Conducted by Humans

James Prather*
Abilene Christian University
Abilene, Texas, USA
james.prather@acu.edu

Paul Denny*
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Juho Leinonen*
University of Auckland
Auckland, New Zealand
juho.leinonen@auckland.ac.nz

Brett A. Becker*
University College Dublin
Dublin, Ireland
brett.becker@ucd.ie

Ibrahim Albluwi
Princess Sumaya University for Technology
Amman, Jordan
i.albluwi@psut.edu.jo

Michael E. Caspersen
Aarhus University
Aarhus, Denmark
mec@it-vest.dk

Michelle Craig
University of Toronto
Toronto, Canada
mcraig@cs.toronto.edu

Hieke Keuning
Utrecht University
Utrecht, The Netherlands
h.w.keuning@uu.nl

Natalie Kiesler
DIPF Leibniz Institute for Research and Information in Education
Frankfurt am Main, Germany
kiesler@dipf.de

Tobias Kohn
TU Wien
Vienna, Austria
tobias.kohn@tuwien.ac.at

Andrew Luxton-Reilly
University of Auckland
Auckland, New Zealand
a.luxton-reilly@auckland.ac.nz

Stephen MacNeil
Temple University
Philadelphia, Pennsylvania, USA
stephen.macneil@temple.edu

Andrew Petersen
University of Toronto Mississauga
Toronto, Canada
andrew.petersen@utoronto.ca

Raymond Pettit
University of Virginia
Charlottesville, Virginia, USA
rp6zr@virginia.edu

Brent N. Reeves
Abilene Christian University
Abilene, Texas, USA
brent.reeves@acu.edu

Jaromir Savelka
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
jsavelka@andrew.cmu.edu

ABSTRACT

The recent advent of highly accurate and scalable large language models (LLMs) has taken the world by storm. From art to essays to computer code, LLMs are producing novel content that until recently was thought only humans could produce. Recent work in computing education has sought to understand the capabilities of LLMs for solving tasks such as writing code, explaining code, creating novel coding assignments, interpreting programming error messages, and more. However, these technologies continue to evolve at an astonishing rate leaving educators little time to adapt. This working group seeks to document the state-of-the-art for code generation LLMs, detail current opportunities and challenges related to their use, and present actionable approaches to integrating them into computing curricula.

*Randomly-ordered Co-leaders

ITiCSE 2023, July 8–12, 2023, Turku, Finland

ACM ISBN 979-8-4007-0139-9/23/07.
https://doi.org/10.1145/3587103.3594206

CCS CONCEPTS

• **Social and professional topics → Computing education**; • **Computing methodologies → Artificial intelligence**.

KEYWORDS

AI; artificial intelligence; code generation; Codex; computer programming; Copilot; CS1; GitHub; GPT; large language models; LLM; novice programming; OpenAI; pedagogical practices

ACM Reference Format:
James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michael E. Caspersen, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. Transformed by Transformers: Navigating the AI Coding Revolution for Computing Education: An ITiCSE Working Group Conducted by Humans. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2023), July 8–12, 2023, Turku, Finland*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3587103.3594206

1 INTRODUCTION

Recent advancements in artificial intelligence (AI) have ushered in a new era of computing. One particular class of AI models, known as large language models (LLMs), has shown remarkable capabilities in the generation and interpretation of natural language data and source code interpretation and generation. Many timely and important questions remain unanswered about how we will adapt to the challenges and opportunities this presents. If students are able to generate solutions to all of their programming coursework, how does this impact what is taught, how it is taught, and how students will remain motivated to learn?

For instance, many introductory programming courses use a popular evidence-based approach involving students writing dozens of small programming exercises checked by automated assessment tools. These problems can now be solved quite easily by new tools that provide students access to powerful LLMs [3]. The most recent models can solve even more complex data structures and algorithms -level assignments [4]. This casts doubt about the efficacy and longevity of current pedagogical practices and raises concerns about student learning, plagiarism, and over-reliance [1]. Current tools, such as Github Copilot, can provide code solutions in a student's IDE and are free to use. However, for every right answer these tools provide, they also can provide wrong or ambiguous code and can include unnecessary elements [11]. Students using these tools can also quickly become lost while reading code they didn't write [5] or lazily drift from code suggestion to code suggestion without understanding what they are doing [9].

The emergence of large language models could also bring benefits, however. These models can be used to scaffold instructors in creating educational resources such as novel, personalized programming exercises [2, 10], code explanations that could help support students when they are working on exercises [6, 8], and enhanced programming error messages that might be easier to understand for novice programmers [7].

We believe that large language models will have profound impacts on computing education. This working group will work towards understanding how we can make those impacts as positive as possible.

2 GOALS

This working group is motivated by the following goals:

(1) Identify areas/aspects of computing education where LLMs could be used from both the student and teacher perspective. We plan to collect data from a wide range of computing educators to identify and expand upon the most influential candidate areas.
(2) Present a guide to the opportunities and challenges of LLMs in computing education as well as likely future opportunities and challenges, given that these models seem likely to improve rapidly.
(3) Replicate prior work on the performance of current LLM models on programming problems, exam questions, and other curricula. From this work, choose appropriate benchmarks by which future work can determine the efficacy of these models and provide a standard for replication.
(4) Create an evidence-based resource of pedagogical approaches for which LLMs can be utilised so that programming educators can utilise these new tools effectively.
(5) Cast a bold yet practical vision for the future of programming education in this new era.

Although we plan to include an extensive review of the literature, we recognize that any such attempt in this nascent and rapidly expanding area of research will quickly become out of date. We will therefore focus on the present *status quaestionis* and provide recommendations based on that.

REFERENCES

[1] Brett Becker, James Prather, Paul Denny, Andrew Luxton-Reilly, James Finnie-Ansley, and Eddie Antonio Santos. 2023. Programming Is Hard - Or at Least It Used to Be: Educational Opportunities And Challenges of AI Code Generation *(SIGCSE '23)*. ACM.
[2] Paul Denny, Sami Sarsa, Arto Hellas, and Juho Leinonen. 2022. Robosourcing Educational Resources – Leveraging Large Language Models for Learnersourcing. https://doi.org/10.48550/ARXIV.2211.04715
[3] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming *(ACE '22)*. ACM, Online, 10–19. https://doi.org/10.1145/3511861.3511863
[4] James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know If This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises *(ACE '23)*. ACM, NY, NY, USA, 97–104. https://doi.org/10.1145/3576123.3576134
[5] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the Effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 455, 23 pages. https://doi.org/10.1145/3544548.3580919
[6] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing Code Explanations Created by Students and Large Language Models. arXiv:2304.03938 [cs.CY]
[7] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A. Becker. 2023. Using Large Language Models to Enhance Programming Error Messages *(SIGCSE 2023)*. ACM, NY, NY, USA, 563–569. https://doi.org/10.1145/3545945.3569770
[8] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 931–937. https://doi.org/10.1145/3545945.3569785
[9] James Prather, Brent N. Reeves, Paul Denny, Brett A. Becker, Juho Leinonen, Andrew Luxton-Reilly, Garrett Powell, James Finnie-Ansley, and Eddie Antonio Santos. 2023. "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. https://doi.org/10.48550/arXiv.2304.02491 arXiv:2304.02491 [cs.HC]
[10] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models *(ICER '22)*. ACM, NY NY, USA, 27–43. https://doi.org/10.1145/3501385.3543957
[11] Michel Wermelinger. 2023. Using GitHub Copilot to Solve Simple Programming Problems. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 172–178. https://doi.org/10.1145/3545945.3569830