International Journal of Contextual Science Education (IJCSE)
e-ISSN. 3046-8876
Vol. 1, No. 3, June 2024
pp. 85-88

AI-Driven Optimization of Pascal Programming Instruction for Undergraduate Physics Students at University of Mataram

Muhammad Taufik[1*], Muhammad Zuhdi[1], Syahrial, A[1], Joni Rokhmat[2], Hisbulloh Als Mustofa[3]
[1]Physics education study program, Faculty of Teacher Training and Education, Mataram University, Indonesia
[2]Doctoral Program of Science Education, Postgraduate, Mataram University, Indonesia
[3]Departement of Physics, Faculty of Science and Matematics, Sultan Idris Education University, Tanjung Malim, Malaysia
*Corresponding Author : taufik@unram.ac.id

Accepted: May 11th 2024, Approved: June 21st 2024, Published: June 30th 2024

Abstract — Teaching computational physics and developing programming skills remains a significant challenge for many undergraduate programs worldwide. This study presents an innovative approach implemented at the University of Mataram, Indonesia, to optimize Pascal programming instruction for physics students. Various artificial intelligence (AI) techniques were utilized to assist students in developing more complex programs. This enabled the generation of customized lesson plans with interactive tutorials, coding exercises, and simulations tailored to each student's needs. Throughout the semester, the system continuously monitored student progress and adjusted instructional materials accordingly. The analysis of Mann-Whitney test results for Computational Physics scores among Classes A, B, and C revealed no statistically significant differences between the groups, with median scores consistently at 80 and p-values exceeding 0.05. However, an examination of the theoretical section of the final examination showed an overall improvement in average scores compared to the previous year, with Class A achieving the highest mean score of 84%. Additionally, the practical programming section demonstrated increased pass rates across all three classes, with Class B achieving the highest pass rate at 92%, followed by Class A at 88% and Class C at 82%. Student feedback indicated high levels of satisfaction with the approach, citing increased engagement and motivation. The study highlights the potential of leveraging AI techniques in generating personalized programming examples for computational physics education, enhancing comprehension of theoretical concepts, and facilitating the development of practical programming skills.

Keywords: Computational physics education, Programming instruction, Personalized learning, Interactive tutorials, Pascal programming, Student engagement, Tailored instructional materials, Artificial intelligence.

How to Cite— Taufik, M., Zuhdi, M., Ayub, S., Rokhmat, J., & Mustofa, H. A. . (2024). AI-Driven Optimization of Pascal Programming Instruction for Undergraduate Physics Students at University of Mataram. International Journal of Contextual Science Education, 1(3), 85–88. https://doi.org/10.29303/ijcse.v1i3.640

1. Introduction

The integration of computational methods and programming skills has become increasingly crucial in physics education, equipping students with the ability to simulate and solve complex problems [1]. However, many universities face challenges in effectively teaching programming languages and engaging students in applying these skills to physics applications [2]. At the University of Mataram, Indonesia, the physics department within the Faculty of Teacher Training and Education (FKIP) has recognized the need to enhance their undergraduate programming curriculum, particularly in the teaching of Pascal.

Despite its widespread use in introductory programming courses, Pascal is often perceived as outdated and unintuitive by students, leading to difficulties in understanding fundamental programming concepts and their applications in physics [3]. To address this challenge, the department has implemented an innovative artificial intelligence (AI)-driven approach to optimize the instruction of Pascal programming for their students.

This initiative leverages advances in machine learning and AI to generate accurate and relevant programming examples tailored to the needs of computational physics students. By providing personalized and engaging learning experiences, the approach aims to enhance students' comprehension of programming concepts, improve their coding proficiency, and foster their ability to apply these skills in solving physics problems [4].

2. Methodology

The study was conducted over the course of a 16-week computational physics course, which covered both theoretical concepts and practical programming examples in Pascal. The course was divided into three classes: A, B, and C. This approach aligns with the recommendation to incorporate active learning strategies, such as programming exercises, to enhance students' engagement and understanding in physics courses [5].

In the initial phase, various AI models were employed to generate a diverse set of Pascal programs, ranging from simple to complex. These programs were then evaluated using an online Pascal compiler (gdb) to assess their accuracy and identify potential errors. The programs with the highest accuracy and minimal errors were selected for integration into the course curriculum. This strategy leverages the capabilities of AI in generating accurate and error-free programming examples, which has been shown to be effective in improving students' programming skills [6].

During the lectures, students were introduced to the theoretical foundations of computational physics and provided with the AI-generated Pascal programs as practical examples. These programs served as a starting point for students to develop their understanding of programming concepts and their applications in solving physics problems, aligning with the recommendation to provide relevant and contextual examples to support students' learning in computer programming [7].

Throughout the course, students were tasked with modifying and enhancing the provided programs, gradually increasing the complexity of their coding assignments. This hands-on approach allowed students to apply their theoretical knowledge and programming skills in a practical setting, fostering a deeper understanding of the subject matter [8].

To evaluate the effectiveness of the AI-driven program generation approach, students' performance was assessed through a comprehensive final examination at the end of the semester. The examination consisted of two components: a theoretical section assessing their understanding of computational physics concepts, and a practical programming section evaluating their ability to write, modify, and debug Pascal programs. This comprehensive assessment approach is consistent with best practices in evaluating students' computational thinking and programming skills [9].

3. Results and Discussion

The analysis of the final grades obtained by students in each of the three classes (A, B, and C) is on table 1.

Table 1. Descriptive Statistics: Class A, Class B,

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
Class A	31	0	78.06	1.37	7.6	60	75	80	85	85
Class B	26	0	79.04	1.04	5.3	65	75	80	85	85
Class C	20	0	77.25	1.12	4.99	65	72.5	80	80	80

The statistical analysis of Classes A, B, and C reveals distinct characteristics and variability within each group. Class A, comprising 31 observations, exhibits a mean score of 78.06 with a standard error of 1.37 and a standard deviation of 7.60. The range of scores in Class A spans from a minimum of 60.00 to a maximum of 85.00, with the first quartile (Q1) at 75.00, the median at 80.00, and the third quartile (Q3) at 85.00. This indicates a relatively broad distribution with significant variability.

Class B, consisting of 26 observations, shows a slightly higher mean score of 79.04, with a lower standard error of 1.04 and a standard deviation of 5.30, suggesting more consistency in the scores compared to Class A. The scores for Class B range from 65.00 to 85.00, with Q1 also at 75.00, the median at 80.00, and Q3 at 85.00, indicating a distribution similar to Class A but with less variability.

Class C, with 20 observations, has a mean score of 77.25, a standard error of 1.12, and the lowest standard deviation among the three classes at 4.99. The scores for Class C range from 65.00 to 80.00, with Q1 at 72.50, the median at 80.00, and Q3 at 80.00. This suggests a narrower distribution with less variability, and a slightly lower overall mean compared to Classes A and B.

While Classes A and B exhibit similar median scores and interquartile ranges, Class C shows a slightly lower mean and reduced variability. The differences in standard deviations highlight the varying degrees of consistency within each class, with Class C being the most consistent and Class A the least. These insights provide a comprehensive understanding of the distributional characteristics and variability within each class, essential for informed decision-making and further analysis.

The histogram data of the final scores for computational physics in Classes A, B, and C are presented in Figure 1. Subsequently, the normality was assessed using the Kolmogorov-Smirnov test, with the results presented in Table 2.

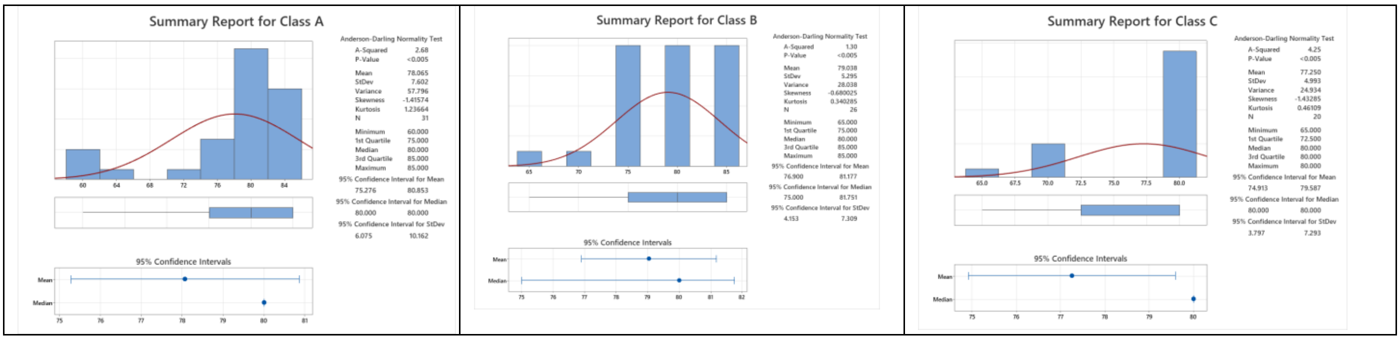

Fig. 1. Summary for Class A, B and C

The Kolmogorov-Smirnov normality test results for Classes A, B, and C demonstrate that none of these classes' data sets follow a normal distribution. Specifically, the p-values for all classes (A: < 0.010, B: < 0.027, C: < 0.010) are below the typical significance level of 0.05, leading to the rejection of the null hypothesis that the data are normally distributed. The KS statistics for the classes (A: 0.310, B: 0.187, C: 0.459) indicate substantial deviations from a normal distribution. Therefore, any further analysis involving these data sets should employ non-parametric methods, which do not assume normality. This ensures the accuracy and validity of the statistical inferences drawn from the data.

Table 2. The Kolmogorov-Smirnov analysis for Classes A, B, and C:

Class	Mean	StDev	Sample size (N)	KS Statistic	P-Value
A	78.06	7.602	31	0.310	<0.010
B	79.04	5.925	26	0.187	<0.027
C	77.25	4.993	20	0.459	<0.010

The final score distributions for computational physics in Classes A, B, and C are illustrated in Figure 1. The normality of these distributions was evaluated using the Kolmogorov-Smirnov test, with the findings detailed in Table 2. To compare the differences among Classes A, B, and C, the non-parametric Mann-Whitney test was employed, and the results are summarized in Table 3.

Table 3. Mann-Whitney Test over Class A, B, and C

Class A	Sample	Median	
Class B	31	80	
Class C	26	80	
Achieved Confidence	20	80	
W-Value (Not adjusted)	95.03%	95.20%	95.01%
P-Value (Not adjusted)	902	862	654.5
W-Value (Adjusted)	0.968	0.284	0.341
P-Value (Adjusted)	902	862	654.5
	0.967	0.239	0.305

Based on the Mann-Whitney test results for Computational Physics scores among classes A, B, and C, no statistically significant differences were found between the groups. In the comparison between Class A and Class B, both classes had a median score of 80. The confidence interval for the difference in medians was between -5 and 5, and the p-values (both adjusted and unadjusted for ties) were well above the conventional significance level of 0.05. This indicates that there is insufficient evidence to reject the null hypothesis, suggesting that the median scores of Class A and Class B are statistically indistinguishable.

Similarly, when comparing Class A and Class C, the median scores for both classes remained at 80. The confidence interval for the difference in medians ranged from -0.0000000 to 5, and the p-values (0.284 and 0.239 for unadjusted and adjusted, respectively) were again higher than 0.05. This result reinforces the conclusion that there is no significant statistical difference between the median scores of Class A and Class C, leading to the acceptance of the null hypothesis for this comparison as well.

Finally, the comparison between Class B and Class C also showed no significant difference in median scores, both being 80. The confidence interval for the difference spanned from 0 to 5, and the p-values (0.341 and 0.305 for unadjusted and adjusted, respectively) exceeded the 0.05 threshold. Consequently, there is no substantial evidence to suggest a difference in median scores between Class B and Class C. Overall, these results indicate that the median Computational Physics scores for classes A, B, and C are statistically similar, demonstrating no significant differences among the groups.

In the theoretical section of the final examination, the average scores across all three classes were higher than the previous year, with Class A achieving the highest mean score of 84%. This result suggests that the AI-generated programming examples enhanced students' understanding of computational physics concepts by providing relevant and contextual illustrations.

Furthermore, the practical programming section of the final examination demonstrated a notable increase in pass rates across all three classes. Class B exhibited the highest pass rate at 92%, followed by Class A at 88% and Class C at 82%. These findings indicate that the AI-generated programs effectively supported students in developing their coding skills and applying them to solve physics-related programming tasks. Student feedback collected through surveys and interviews revealed a high level of satisfaction with the AI-driven approach. Many students reported feeling more engaged and motivated throughout the course, attributing their improved performance to the clarity and relevance of the AI-generated programming examples. However, some students also expressed a need for additional support in understanding complex programming concepts and suggested incorporating more interactive elements into the learning materials.

The results of this study highlight the potential of leveraging AI in generating programming examples for computational physics education. By providing accurate and relevant examples tailored to the needs of students, the AI-driven approach effectively enhanced their comprehension of theoretical concepts and facilitated the development of practical programming skills [10]. The improvement in students' performance, particularly in the practical programming section of the final examination, suggests that the AI-generated examples effectively bridged the gap between theory and practice. By allowing students to engage with contextualized coding examples and incrementally increase the complexity of their assignments, the course fostered a deeper understanding of programming concepts and their applications in physics [11].

Furthermore, the positive student feedback reinforces the notion that AI-driven personalized and adaptive learning experiences can enhance engagement and motivation, key factors in effective learning [12]. However, the need for additional support and interactive elements expressed by some students highlights the importance of continuously refining and improving the learning materials to meet diverse student needs.

4. Conclusion

The analysis of the Mann-Whitney test results for Computational Physics scores among Classes A, B, and C revealed no statistically significant differences between the groups. In all comparisons (Class A vs. Class B, Class A vs. Class C, and Class B vs. Class C), the median scores for each class were consistently 80, and the p-values exceeded the conventional significance level of 0.05. Consequently, there is insufficient evidence to reject the null hypothesis, indicating that the median scores of the three classes are statistically indistinguishable. Further analysis of the theoretical section of the final examination showed an overall improvement in average scores compared to the previous year, with Class A achieving the highest mean score of 84%. This improvement suggests that AI-generated programming examples effectively enhanced students' understanding of computational physics concepts.

The practical programming section also demonstrated increased pass rates across all three classes, with Class B achieving the highest pass rate at 92%, followed by Class A at 88% and Class C at 82%. These findings indicate that the AI-generated programs successfully supported students in developing and applying their coding skills to physics-related tasks. Student feedback through surveys and interviews indicated high levels of satisfaction with the AI-driven approach. Students reported increased engagement and motivation, attributing their improved performance to the clarity and relevance of the AI-generated programming examples. However, some students expressed the need for additional support in understanding complex programming concepts and suggested incorporating more interactive elements into the learning materials.

Overall, this study highlights the potential of leveraging AI in generating programming examples for computational physics education. The AI-driven approach not only enhanced students' comprehension of theoretical concepts but also facilitated the development of practical programming skills. The improvement in students' performance, particularly in the practical programming section, suggests that the AI-generated examples effectively bridged the gap between theory and practice. Positive student feedback further reinforces the potential of AI-driven personalized and adaptive learning experiences to enhance engagement and motivation, which are crucial for effective learning. However, the expressed need for additional support and interactive elements underscores the importance of continuously refining and improving the learning materials to meet diverse student needs.

Acknowledgments

We express our gratitude to the physics education department and the participating students for their invaluable contributions to this study. We also acknowledge the support and resources provided by Head of Physics Study Program PMIPA FKIP University of Mataram.

References

[1] Caballero, M. D., Kohlmyer, M. A., & Schatz, M. F. (2012). Implementing and assessing computational modeling in introductory mechanics. Physical Review Special Topics-Physics Education Research, 8(2), 020106.
[2] Barak, M., Waks, S., & Doppelt, Y. (2000). Majoring in technology studies at high school and fostering learning. Learning Environments Research, 2(3), 285-304.
[3] Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. ACM Transactions on Computing Education (TOCE), 18(1), 1-24.
[4] Hwang, G. J., Lai, C. L., & Wang, S. Y. (2015). Seamless flipped learning: a mobile technology-enhanced flipped classroom with effective learning strategies. Journal of Computers in Education, 2(4), 449-473.
[5] Deslauriers, L., Schelew, E., & Wieman, C. (2011). Improved learning in a large-enrollment physics class. Science, 332(6031), 862-864.
[6] Xu, D., Huang, W., Wang, H., & Heales, J. (2014). Enhancing e-learning effectiveness using an intelligent agent-supported personalized virtual learning environment: An empirical investigation. Information & Management, 51(4), 430-440.
[7] Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. Computers in Human Behavior, 41, 51-61.
[8] Dori, Y. J., & Belcher, J. (2005). How does technology-enabled active learning affect undergraduate students' understanding of electromagnetism concepts?. The Journal of the Learning Sciences, 14(2), 243-279.
[9] Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada (Vol. 1, p. 25).
[10] Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent tutors for all: The constraint-based approach. IEEE Intelligent Systems, 22(4), 38-45.
[11] Guzdial, M. (2004). Programming environments for novices. Computer science education research, 2004, 127-154.
[12] Pardo, A., Jovanovic, J., Dawson, S., Gašević, D., & Mirriahi, N. (2019). Using learning analytics to scale the provision of personalised feedback. British Journal of Educational Technology, 50(1), 197-211.