The Habit of Coding: Towards Enhancing Introductory Programming

It is no doubt that computer programs offer different benefits to society and business. Thus, Introductory programming courses (IPC) are now a mandatory part of most undergraduate programs. IPCs are usually challenging to some students, and high dropout rates are reported [1] [2]. Many pedagogical techniques were introduced and experimented with students [3] [4]. However, finding one effective method for all students to achieve the desired goals seems to be difficult [5]. Having a user-centered adaptive learning system, taking into consideration the individual needs and learning abilities of the student, can be the solution [6] [7]. However, the current state of knowledge and practice in this field is limited. In an attempt to fill in this gap, our aim is to build what we call "The Habit of Coding": by getting students motivated to learn and work autonomously.

We need to investigate the following research questions:

1. What are the factors that contribute to developing the "Habit of Coding"?
2. How to employ these factors in building an automated intelligent tutoring system for an IPC?
3. How effective is developing a "Habit of Coding" in improving the learning experience of novice programmers?

In order to answer those questions, we need to experiment extensively with students and observe their coding patterns and performance, by logging student-generated data and analyzing the data qualitatively and quantitatively.

We would start by experimenting how autonomy can be applied in learning programming, for instance by setting a learning path while giving the students the freedom to progress on their own pace. Students can choose the programming language and tools. Moreover, they can choose to code independently, in pairs or in groups. One more factor can be learning coding as separate independent tasks or working towards building a project. All of these factors are to be experimented independently first and then merged in order to decide on the actual impact of each of these components on students' performance.

---

[1] Carneiro, M. G., Dutra, B. L., Paiva, J. G. S., Gabriel, P. H. R., & Araújo, R. D. (2022). Educational data mining to support identification and prevention of academic retention and dropout: a case study in introductory programming. *Revista Brasileira de Informática na Educação*, *30*, 379-395.

[2] Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM inroads*, *10*(2), 30-36.

[3] Kanika, Chakraverty, S., & Chakraborty, P. (2020). Tools and techniques for teaching computer programming: A review. Journal of Educational Technology Systems, 49(2), 170-198.

[4] Almasri, A., Ahmed, A., Almasri, N., Abu Sultan, Y. S., Mahmoud, A. Y., Zaqout, I. S., ... & Abu-Naser, S. S. (2019). Intelligent tutoring systems survey for the period 2000-2018.

[5] Figueiredo, J., & García-Peñalvo, F. J. (2020, October). Intelligent tutoring systems approach to introductory programming courses. In Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (pp. 34-39).

[6] Vesin, B., Mangaroska, K., & Giannakos, M. (2018). Learning in smart environments: user-centered design and analytics of an adaptive learning system. Smart Learning Environments, 5(1), 1-21.

[7] Alqahtani, R., Kaliappen, N., & Alqahtani, M. (2021). A review of the quality of adaptive learning tools over non-adaptive learning tools. International Journal for Quality Research, 15(1), 45.