XÂY DỰNG HỆ THỐNG BÀI TẬP VÀ BỘ KIỂM THỬ CHẤM TỰ ĐỘNG TRONG GIẢNG DẠY NGÔN NGỮ LẬP TRÌNH CƠ BẢN NHẰM NÂNG CAO NĂNG LỰC TỰ HỌC LẬP TRÌNH CHO SINH VIÊN TRƯỜNG ĐẠI HỌC HOA LƯ

Đồng Thị Thu*, Vũ Thị Quyên*, Phùng Thị Thao**
***Trường Đại học Hoa Lư**
****Trường Phổ thông Thực hành Sư phạm Tràng An**

Tóm tắt: *Trước yêu cầu đổi mới giáo dục đại học và chuyển đổi số, việc nâng cao năng lực tự học cùng kỹ năng thực hành lập trình cho sinh viên ngành Công nghệ thông tin (CNTT) trở thành nhiệm vụ mang tính then chốt. Dù vậy, công tác giảng dạy các học phần lập trình cơ bản hiện nay vẫn gặp nhiều rào cản: nguồn bài tập chưa đa dạng, quy trình chấm bài thủ công kéo dài độ trễ phản hồi, khiến sinh viên thiếu môi trường rèn luyện thường xuyên và khó tự đánh giá năng lực. Nghiên cứu này đề xuất giải pháp xây dựng hệ thống bài tập phân loại theo chủ đề kiến thức, phân tầng dựa trên chuẩn đầu ra (CLO) và thang nhận thức Bloom, kết hợp bộ dữ liệu kiểm thử (test cases) chấm tự động được chuẩn hóa. Kết quả thực nghiệm sư phạm tại Trường Đại học Hoa Lư cho thấy: việc ứng dụng hệ thống không chỉ rút ngắn chu trình phản hồi và giảm tải áp lực chấm bài cho giảng viên, mà còn gia tăng rõ rệt các chỉ số thành tố của năng lực tự học ở sinh viên (đặc biệt là kỹ năng tự kiểm tra - sửa lỗi và khả năng duy trì động lực học tập). Mô hình khẳng định tính hiệu quả và khả năng nhân rộng trong đào tạo ngành Công nghệ thông tin.*

Từ khóa: *Chấm bài tự động, Online Judge, lập trình cơ bản, năng lực tự học, bộ kiểm thử, giáo dục đại học.*

DEVELOPING AN EXERCISE SYSTEM AND AUTOMATED GRADING TEST CASES IN TEACHING BASIC PROGRAMMING LANGUAGES TO ENHANCE STUDENTS' PROGRAMMING SELF-STUDY CAPACITY AT HOA LU UNIVERSITY

Abstract: *In response to the demands of higher education reform and digital transformation, enhancing self-study capacity and practical programming skills for Information Technology (IT) students has become a pivotal task. However, teaching introductory programming courses currently faces several barriers: a lack of diverse exercise sources and a manual grading process that creates long feedback delays, leaving students without a regular practice environment and making self-assessment difficult. This study proposes a solution by developing a system of exercises categorized by knowledge topics, structured according to Course Learning Outcomes (CLOs) and Bloom's taxonomy, combined with standardized test cases for automated grading. Pedagogical experimental results at Hoa Lu University demonstrate that implementing this system not only shortens the feedback loop and reduces the grading workload for lecturers, but also significantly improves components of students' self-study capacity—particularly self-testing and debugging skills, as well as the ability to maintain learning motivation. The model confirms its effectiveness and potential for scalability in Information Technology education.*

Keywords: *Automated grading, online Judge, basic programming, self-study capacity, test cases, higher education.*

Nhận bài: 09/06/2026 *Phản biện: 09/07/2026* *Duyệt đăng: 12/07/2026*

I. ĐẶT VẤN ĐỀ

Chuyển đổi số (CĐS) trong giáo dục đại học, đặc biệt đối với ngành Công nghệ thông tin (CNTT), đòi hỏi người học phải nâng cao năng lực tự học, khả năng tự nghiên cứu và khả năng thích ứng với sự thay đổi nhanh chóng của công nghệ. Định hướng này phù hợp với Chương trình Chuyển đổi số quốc gia, trong đó giáo dục được xác định là một trong những lĩnh vực cần ưu tiên CĐS nhằm đổi mới phương thức dạy học, nâng cao chất lượng đào tạo và phát triển nguồn nhân lực số (Thủ tướng Chính phủ, 2020).

Trong lĩnh vực giáo dục và đào tạo, CĐS không chỉ dừng lại ở việc số hóa học liệu mà còn hướng đến đổi mới phương thức tổ chức dạy học, kiểm tra, đánh giá và quản trị quá trình học tập dựa trên dữ liệu. Đề án tăng cường ứng dụng CNTT và CĐS trong giáo dục và đào tạo giai đoạn 2022-2025, định hướng đến năm 2030 nhấn mạnh việc phát triển các nền tảng số, học liệu số và hệ thống quản lý học tập nhằm hỗ trợ dạy học linh hoạt, cá nhân hóa và nâng cao chất lượng kiểm tra, đánh giá (Thủ tướng Chính phủ, 2022).

Trong chương trình đào tạo ngành CNTT, các học phần lập trình cơ bản như C++ hoặc Python đóng vai trò nền tảng, giúp sinh viên (SV) hình thành tư duy máy tính, phương pháp phân tích thuật toán và kỹ năng tổ chức dữ liệu. Việc thiết kế nội dung, hoạt động thực hành và phương

pháp đánh giá cần xuất phát từ chuẩn đầu ra của chương trình và chuẩn đầu ra học phần, đồng thời bảo đảm sự tương thích giữa mục tiêu, nội dung, phương pháp giảng dạy và đánh giá theo quy định về chuẩn chương trình đào tạo đại học (Bộ Giáo dục và Đào tạo, 2021).

Tuy nhiên, thực tế giảng dạy tại Trường Đại học Hoa Lư cho thấy việc giao và chấm bài tập lập trình theo phương pháp thủ công còn tồn tại những hạn chế như tốn nhiều thời gian của giảng viên, khó kiểm tra đầy đủ các trường hợp dữ liệu và tạo ra độ trễ lớn trong phản hồi. Những hạn chế này ảnh hưởng đến cơ hội thực hành, khả năng tự phát hiện lỗi và quá trình tự điều chỉnh việc học của SV. Vì vậy, việc xây dựng ngân hàng bài tập kết hợp bộ kiểm thử chấm tự động là giải pháp phù hợp với định hướng đổi mới phương pháp dạy học (PPDH), kiểm tra, đánh giá và tăng cường ứng dụng công nghệ số trong giáo dục (Thủ tướng Chính phủ, 2022, 2024).

II. NỘI DUNG NGHIÊN CỨU

2.1 Cơ sở lý luận và khảo sát thực trạng

2.1.1. Năng lực tự học lập trình trong giáo dục đại học

Trong xu thế chuyển từ giáo dục tiếp cận nội dung sang phát triển năng lực, năng lực tự học đóng vai trò then chốt đối với chất lượng đào tạo và khả năng học tập suốt đời. Đối với ngành CNTT – lĩnh vực thay đổi nhanh chóng về công nghệ, ngôn ngữ và công cụ – SV không thể chỉ phụ thuộc vào giờ học trên lớp mà phải chủ động cập nhật, thực hành và mở rộng kiến thức.

Về mặt lý luận, Holec (1981) coi tự chủ học tập là khả năng người học tự chịu trách nhiệm về quá trình học (xác định mục tiêu, chọn nội dung, phương pháp, theo dõi và đánh giá). Phát triển quan điểm này, Little (1991) nhấn mạnh việc quản lý học tập qua lập kế hoạch, thực hiện, tự đánh giá và điều chỉnh chiến lược. Như vậy, tự học không phải là hoạt động tách biệt giảng viên, mà là quá trình người học tăng cường kiểm soát mục tiêu, phương pháp và kết quả học tập.

Tại Việt Nam, Đào Việt Hùng và cộng sự (2024) xác định tự học là năng lực cơ bản giúp SV tiếp thu kiến thức mới và thích ứng với sự phát triển khoa học - công nghệ. Nhóm tác giả đã xây dựng khung năng lực và công cụ đánh giá theo tiến trình, khẳng định tự học cần được đánh giá qua nhiều biểu hiện thay vì chỉ qua một bài kiểm tra cuối kỳ.

Trong việc học lập trình, Phạm Xuân Nguyện và cộng sự (2024) chỉ ra rằng việc khai thác công cụ số, diễn đàn và AI giúp người học tra cứu, trao đổi, rèn luyện thường xuyên và hình thành năng lực tự nghiên cứu. Hiệu quả học lập trình phụ thuộc vào quá trình thực hành và tương tác liên tục trên các nền tảng số.

Ở ngành CNTT, năng lực tự học lập trình biểu hiện qua việc chủ động đọc tài liệu, phân tích bài toán, xây dựng thuật toán, viết mã, kiểm thử, phát hiện lỗi, tối ưu lời giải và tự đánh giá tiến bộ. Năng lực này chỉ hình thành khi người học trực tiếp giải bài tập, chạy thử và sửa lỗi. Robins và cộng sự (2003) cho biết người mới học thường gặp khó khăn trong dựng mô hình tư duy và chuyển ý tưởng thành mã nguồn. Do đó, một môi trường học tập linh hoạt, cho phép thực hành nhiều lần và nhận phản hồi tức thì có ý nghĩa đặc biệt quan trọng.

Từ đó, nghiên cứu này định nghĩa năng lực tự học lập trình là khả năng SV chủ động xác định mục tiêu, lập kế hoạch, chọn bài tập, thiết kế thuật toán, viết mã, kiểm thử, sửa lỗi và tự đánh giá sự tiến bộ. Năng lực này gồm 5 thành tố cơ bản:

Chủ động xác định mục tiêu và lập kế hoạch: Tự chọn nội dung, bài tập phù hợp và phân bổ thời gian theo chủ đề.

Tự nghiên cứu và giải quyết bài toán: Khai thác tài liệu, phân tích yêu cầu, xác định I/O, chọn cấu trúc dữ liệu và xây dựng thuật toán trước khi viết mã.

Tự kiểm tra và sửa lỗi: Sử dụng phản hồi từ hệ thống để nhận diện lỗi cú pháp, logic, hiệu năng và trường hợp biên, từ đó điều chỉnh lời giải.

Tự đánh giá sự tiến bộ: Sử dụng phản hồi từ hệ thống để nhận diện lỗi cú pháp, logic, hiệu năng và trường hợp biên, từ đó điều chỉnh lời giải.

Duy trì động lực và điều chỉnh chiến lược: Tạo thói quen lập trình thường xuyên, tăng độ khó bài tập, linh hoạt đổi phương pháp và tìm kiếm sự hỗ trợ khi cần.

Tóm lại, tự học lập trình đòi hỏi một môi trường cho phép thực hành liên tục, nộp bài nhiều lần, nhận phản hồi ngay và theo dõi tiến trình học tập.

2.1.2. Hệ thống chấm bài tự động (Online Judge)

Hệ thống chấm bài tự động (Online Judge – OJ) tự động hóa quá trình tiếp nhận, thực thi và đánh giá mã nguồn dựa trên các bộ test case chuẩn bị trước, trả kết quả tức thì. Mô hình này giảm tải công việc cho giảng viên, bảo đảm tính khách quan và có khả năng đánh giá trên quy mô lớn (Douce et al., 2005; Ala-Mutka, 2005).

Theo Ala-Mutka (2005), OJ đánh giá đồng thời tính đúng đắn của thuật toán, độ chính xác của đầu ra và mức sử dụng tài nguyên (thời gian, bộ nhớ), giúp khuyến khích người học tối ưu hóa chương trình. Quy trình OJ gồm 5 bước: Nộp bài (Submission) -> Biên dịch (Compilation) -> Thực thi (Execution) -> Đánh giá (Judging) -> Phản hồi (Feedback).

Để bảo đảm an toàn, chương trình chạy trong môi trường cách ly (sandbox/container) giúp bảo vệ máy chủ và tạo sự công bằng về tài nguyên (Merkel, 2014). Hệ thống trả về phản hồi chuẩn hóa: Accepted (AC), Wrong Answer (WA), Compilation Error (CE), Runtime Error (RE), Time Limit Exceeded (TLE), Memory Limit Exceeded (MLE). Dù vậy, các hệ thống OJ hiện nay chủ yếu báo loại lỗi chứ chưa cung cấp nhiều gợi ý sư phạm chuyên sâu (Keuning et al., 2018).

Về mặt giáo dục, OJ tạo môi trường thực hành phản hồi liên tục, giúp SV hình thành chu trình: Phân tích -> Lập trìnhc -> Nhận phản hồi -> Điều chỉnh. Nghiên cứu này tiếp cận OJ như trung tâm của môi trường học tập số, kết hợp ngân hàng bài tập phân tầng để giúp SV phát triển năng lực tự học.

2.1.3. Khảo sát thực trạng tại Trường Đại học Hoa Lư

Nhóm nghiên cứu đã tiến hành khảo sát 05 giảng viên và 80 sinh viên ngành CNTT tại Trường Đại học Hoa Lư cho thấy:

Về phía giảng viên: 100% nhận thấy chấm thủ công tốn thời gian, khó tạo đủ test case trường hợp biên; >90% nhất trí cần triển khai hệ thống chấm tự động.

Về phía sinh viên: Còn lúng túng trong việc tự nhận diện lỗi logic và thiếu môi trường rèn luyện tương tác ngoài giờ lên lớp. Trên 95% khẳng định sẽ chăm luyện tập hơn nếu có hệ thống chấm trực tuyến phản hồi tức thì.

2.2. Xây dựng hệ thống bài tập và bộ kiểm thử chấm tự động

2.2.1. Nguyên tắc xây dựng hệ thống bài tập

Hệ thống bài tập được thiết kế theo định hướng phát triển năng lực và dựa trên nguyên tắc Thích ứng tương thích (Constructive Alignment) (Biggs, 1996) cùng Giáo dục dựa trên năng lực (OECD, 2018), tuân thủ 5 nguyên tắc:

Một là, thống nhất với chuẩn đầu ra (CLO): Mỗi bài tập gắn với CLO của học phần, bám sát Thông tư 17/2021/TT-BGDĐT.

Hai là, tính kế thừa và phát triển: Bài tập sắp xếp từ dễ đến khó theo thang nhận thức Bloom sửa đổi (Nhớ, Hiểu, Vận dụng, Phân tích, Giải quyết vấn đề) và nguyên tắc Mastery Learning (Bloom, 1968).

Ba là, tính đa dạng và xác thực: Bao quát các chủ đề lập trình cơ bản, đan xen tình huống thực tế để tăng động lực học tập (Jonassen, 1997).

Bốn là, khả năng đánh giá tự động: Chuẩn hóa cấu trúc (ID, tên bài, đề bài, Input/Output, giới hạn, Sample Test, Test Cases) giúp hệ thống chấm khách quan (Douce et al., 2005).

Năm là, hỗ trợ tự học trên môi trường số: Cho phép truy cập mọi lúc, nộp nhiều lần, nhận phản hồi ngay để thúc đẩy quá trình tự điều chỉnh học tập (Zimmerman, 2002; Quyết định 131/QĐ-TTg).

Hệ thống bài tập được chia thành 09 nhóm chủ đề:

Bảng 1: Các nhóm chủ đề trong học phần lập trình căn bản

TT	Chủ đề kiến thức	Nội dung trọng tâm	Mức độ nhận thức (Bloom)
1	Biến và kiểu dữ liệu	Nhập/xuất dữ liệu, phép toán cơ bản	Nhớ
2	Cấu trúc rẽ nhánh	Lệnh if-else, switch-case	Nhớ - Hiểu
3	Vòng lặp	for, while, do-while	Hiểu - Vận dụng
4	Hàm (Function)	Truyền tham số, chia nhỏ chương trình	Vận dụng
5	Mảng (Array)	Mảng một chiều, mảng hai chiều	Vận dụng
6	Chuỗi (String)	Xử lý chuỗi và ký tự	Vận dụng - Phân tích
7	Cấu trúc (Struct)	Kiểu dữ liệu tự định nghĩa	Vận dụng
8	Thư viện chuẩn (STL)	Vector, Set, Map	Phân tích
9	Tệp tin (File)	Đọc và ghi tệp dữ liệu	Phân tích - Giải quyết vấn đề

2.2.2. Quy trình xây dựng bài tập và thiết kế bộ kiểm thử

Thứ nhất, quy trình xây dựng hệ thống bài tập

Quy trình gồm 6 bước khép kín:

Bước 1. Phân tích CLO học phần: Xác định yêu cầu kiến thức, kỹ năng theo chuẩn đầu ra ngành.

Bước 2. Phân tích nội dung kiến thức: Phân chia thành 9 chủ đề cốt lõi gắn với thang Bloom.

Bước 3. Thiết kế hệ thống bài tập: Đối với mỗi chủ đề, nhóm nghiên cứu xây dựng nhiều bài tập với ba mức độ: Mức cơ bản: giúp SV làm quen với cú pháp và các cấu trúc lập trình cơ bản; Mức vận dụng: yêu cầu kết hợp nhiều kiến thức để giải quyết các bài toán có độ phức tạp trung bình; Mức nâng cao: hướng tới các bài toán tổng hợp, yêu cầu phân tích thuật toán, tối ưu chương trình và xử lý các trường hợp đặc biệt.

Bước 4. Xây dựng bộ kiểm thử: Đối với mỗi bài tập, nhóm nghiên cứu xây dựng bộ kiểm thử (test cases) nhằm bảo đảm khả năng đánh giá toàn diện lời giải của SV. Bộ kiểm thử được thiết kế theo năm nhóm chính: Kiểm thử cơ bản (Basic Test); Kiểm thử trường hợp biên (Boundary Test); Kiểm thử dữ liệu ngoại lệ (Corner Case); Kiểm thử dữ liệu ngẫu nhiên (Random Test); Kiểm thử dữ liệu lớn và đánh giá hiệu năng (Stress Test).

Bước 5. Triển khai trên hệ thống Online Judge: Cấu hình bài tập và bộ test lên hệ thống tự động.

Bước 6. Theo dõi, đánh giá và cập nhật: Phân tích dữ liệu học tập (tỷ lệ AC, loại lỗi) để tinh chỉnh độ khó và bổ sung test case.

Thứ hai, quy trình thiết kế bộ kiểm thử (Test Cases)

Theo Myers và cộng sự (2011), bộ kiểm thử cần có khả năng phát hiện lỗi cao nhất. Mọi bài tập đều có bộ test gồm 5 nhóm: (1) Kiểm thử cơ bản (Basic Tests): Dữ liệu nhỏ, đơn giản để xác minh chức năng cơ bản.

(2) Kiểm thử biên (Boundary Tests): Đánh giá khả năng xử lý các giá trị cực tiểu và cực đại của dữ liệu đầu vào theo giới hạn của đề bài (n=1, n=106, 0, số âm, mảng rỗng).

(3) Kiểm thử ngẫu nhiên (Random Tests): Sinh dữ liệu ngẫu nhiên quy mô lớn và đối chiếu với chương trình chuẩn (Reference Solution).

(4) Kiểm thử hiệu năng (Performance Tests): Dữ liệu tối đa nhằm phát hiện lỗi TLE, MLE.

(5*) Kiểm thử trường hợp đặc biệt (Corner Cases):* Bắt các lỗi hiếm gặp như chia cho 0, chuỗi rỗng, tràn số.

Bảng 2. Nguyên tắc thiết kế các nhóm bộ kiểm thử

Nhóm kiểm thử	Mục tiêu	Ví dụ dữ liệu	Lỗi có thể phát hiện
Basic Tests	Kiểm tra chức năng cơ bản	n = 5, mảng nhỏ	Sai thuật toán, sai công thức
Boundary Tests	Kiểm tra giá trị giới hạn	n = 1; n = 10^6	Lỗi điều kiện, phạm vi
Random Tests	Kiểm tra tính ổn định	Dữ liệu ngẫu nhiên	Sai thuật toán ngẫu nhiên
Performance Tests	Đánh giá hiệu năng	Dữ liệu tối đa	TLE, MLE, thuật toán chưa tối ưu
Corner Cases	Kiểm tra tình huống đặc biệt	Mảng rỗng, chuỗi rỗng, số âm	Runtime Error, Overflow, xử lý thiếu trường hợp

2.2.3. Triển khai trên nền tảng VNOI Online Judge

Hệ thống được triển khai trên nền tảng VNOI OJ tại tổ chức riêng của Trường Đại học https://oj.vnoi.info/organization/dhhl. Việc lựa chọn VNOI OJ nhằm tận dụng hạ tầng sẵn có của một hệ thống Online Judge đã được sử dụng rộng rãi trong cộng đồng lập trình tại Việt Nam, đồng thời giảm chi phí đầu tư, rút ngắn thời gian triển khai và bảo đảm tính ổn định của môi trường thực thi chương trình.

Giảng viên tạo các problem set theo tuần, bài về nhà và kiểm tra định kỳ. SV có thể nộp bài không giới hạn và nhận kết quả tức thì. VNOI OJ lưu trữ toàn bộ dữ liệu học tập (số lần nộp, trạng thái lỗi, thời gian chạy, bộ nhớ). Giảng viên sử dụng dữ liệu này (learning analytics) để theo dõi tiến bộ của người học, phát hiện khó khăn và điều chỉnh giảng dạy kịp thời.

2.3. Kết quả thực nghiệm sư phạm và đánh giá sự phát triển năng lực tự học của SV

Việc đưa hệ thống chấm bài tự động vào thực tiễn giảng dạy đã tạo ra những tác động rõ rệt, thay đổi căn bản phương pháp và thói quen học tập của SV ở các khía cạnh chính sau: Hình thành chu trình học tập thực nghiệm "Thử nghiệm – Phản hồi – Sửa lỗi": Phản hồi thời gian thực buộc SV tự rà soát, sửa lỗi (debug) và kiên trì tối ưu bài làm; Phát triển tư duy tối ưu hóa và kỷ luật lập trình: Áp lực về thời gian/bộ nhớ giúp SV chủ động chọn giải thuật và cấu trúc dữ liệu hiệu quả

để đạt Accepted; Cá nhân hóa hỗ trợ giảng dạy: Giảng viên giảm tải chấm bài thủ công, tập trung phân tích dữ liệu để hỗ trợ các SV còn yếu.

Kết quả khi triển khai giảng dạy tại Đại học Hoa Lư khẳng định việc kết hợp ngân hàng bài tập chuẩn hóa và bộ kiểm thử tự động là giải pháp hiệu quả, giúp nâng cao năng lực tự học của SV và tạo tiền đề nhân rộng cho các học phần lập trình nâng cao.

III. KẾT LUẬN

Nghiên cứu đã hoàn thiện việc xây dựng hệ thống bài tập lập trình cơ bản được thiết kế phân tầng theo chuẩn đầu ra (CLO) và thang nhận thức Bloom, đi kèm bộ dữ liệu kiểm thử (test cases) chuẩn hóa bao phủ toàn diện 5 nhóm trường hợp: cơ bản, biên, ngẫu nhiên, cực hạn và các trường hợp đặc biệt. Kết quả triển khai thực nghiệm trên nền tảng VNOI Online Judge tại Trường Đại học Hoa Lư đã minh chứng cho tính khả thi, hiệu quả cũng như độ tương thích cao của mô hình với thực tiễn đào tạo ngành CNTT.

Thông qua giải pháp chấm bài tự động, nghiên cứu không chỉ giải quyết triệt để áp lực quá tải trong khâu đánh giá thủ công của giảng viên, mà quan trọng hơn, đã thiết lập thành công chu trình phản hồi thời gian thực "Thử nghiệm - Phản hồi - Sửa lỗi" cho SV. Quá trình chủ động cải tiến mã nguồn để đạt trạng thái Accepted là yếu tố then chốt giúp SV từng bước hình thành và phát triển tư duy tối ưu thuật toán, nâng cao kỹ năng gỡ lỗi độc lập và phát triển năng lực tự học - phẩm chất cốt lõi quyết định sự thành công của nhân lực ngành công nghệ.

TÀI LIỆU THAM KHẢO

Ala-Mutka, K. M (2005), A survey of automated assessment approaches for programming assignments. Computer Science Education, 15(2), 83–102. https://doi.org/10.1080/08993400500150747

Anderson, L. W., & Krathwohl, D. R. (Eds.) (2001), A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman.

Biggs, J., & Tang, C (2011), Teaching for quality learning at university (4th ed.). Open University Press.

Bloom, B. S (1968), Learning for mastery. Evaluation Comment, 1(2), 1–12.

Bộ Giáo dục và Đào tạo (2021), Thông tư số 17/2021/TT-BGDĐT ngày 22 tháng 6 năm 2021 quy định về chuẩn chương trình đào tạo; xây dựng, thẩm định và ban hành chương trình đào tạo các trình độ của giáo dục đại học.

Đào Việt Hùng, Nguyễn Mậu Đức, & Lê Hoàng Minh (2024), Xây dựng bộ công cụ đánh giá năng lực tự học của sinh viên. Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên, 229(12), 51–57. https://doi.org/10.34238/tnu-jst.10036

Phạm Xuân Nguyện, Phạm Thị Thanh, & Phùng Thị Thao (2024), Nâng cao hiệu quả tự học ngôn ngữ lập trình cho học sinh thông qua sử dụng một số công cụ số. Tạp chí Giáo dục, 24(9), 144–148.

Thủ tướng Chính phủ (2024), Quyết định số 1705/QĐ-TTg ngày 31 tháng 12 năm 2024 phê duyệt Chiến lược phát triển giáo dục đến năm 2030, tầm nhìn đến năm 2045.