Huang, A. Y. Q., Chang, J. W., Yang, A. C. M., Ogata, H., Li, S. T., Yen, R. X., & Yang, S. J. H. (2023). Personalized Intervention based on the Early Prediction of At-risk Students to Improve Their Learning Performance. *Educational Technology & Society*, *26*(4), 69-89. https://doi.org/10.30191/ETS.202310_26(4).0005

Personalized Intervention based on the Early Prediction of At-risk Students to Improve Their Learning Performance

Anna Y. Q. Huang[1], Jei Wei Chang[1], Albert C. M. Yang[2,4], Hiroaki Ogata[3], Shun Ting Li[1], Ruo Xuan Yen[1] and Stephen J. H. Yang[1*]

[1]Computer Science & Information Engineering, National Central University, Taoyuan City, Taiwan // [2]Graduate School of Informatics, Kyoto University, Japan // [3]Academic Center for Computing and Media Studies, Kyoto University, Japan // [4]Computer Science and Engineering, National Chung-Hsing University, Taichung City, Taiwan // anna.yuqing@gmail.com // a0978122159@gmail.com // albertyang1994@gmail.com // hiroaki.ogata@gmail.com // ariel055132@g.ncu.edu.tw // apple310565@gmail.com // stephen.yang.ac@gmail.com

[*]Corresponding author

(Submitted June 24, 2022; Revised March 6, 2023; Accepted March 24, 2023)

ABSTRACT: To improve students' learning performance through review learning activities, we developed a personalized intervention tutoring approach that leverages learning analysis based on artificial intelligence. The proposed intervention first uses text-processing artificial intelligence technologies, namely bidirectional encoder representations from transformers and generative pretrained transformer-2, to automatically generate Python programming remedial materials; subsequently, learning performance prediction models constructed using various machine learning methods are used to determine students' learning type, enabling the automatic generation of personalized remedial materials. The participants in this study were 78 students from a university in northern Taiwan enrolled in an 8-week Python course. Students in the experimental (n = 36) and control (n = 42) groups engaged in the same programming learning activities during the first 5 weeks of the course, and they completed a pretest of programming knowledge in Week 6. For the review activity in Week 7, the 36 students in the experimental group received personalized intervention, whereas those in the control group received traditional class tutoring. We examined the effect of the self-regulated learning and personalized intervention on the learning performance of students. Compared with the traditional class tutoring, the personalized intervention review activity not only helped students obtain higher learning performance but also prompted greater improvements in the following learning strategies: rehearsal, critical thinking, metacognitive self-regulation, effort regulation, and peer learning. We also observed that students' rehearsal and help-seeking learning strategies indirectly affected learning performance through students' note-taking in the provided e-book.

Keywords: Personalized intervention, Self-regulated learning, Machine learning, Artificial intelligence

1. Introduction

With its roots in learning analysis, precision education aims to improve students' learning performance through the four steps of diagnosis, prediction, treatment, and prevention, drawing inspiration from precision medicine (Lu et al., 2018). Personalized intervention involves formulating specific measures according to the needs of each student, guiding students to overcome learning difficulties, helping students confront high risks, and supporting students in improving their learning performance (Zhang et al., 2020). Thus, the objective of both precision education and personalized intervention is designing unique interventions for different students.

According to the US Department of Education (2010), personalized learning refers to teachers customizing learning objectives, teaching methods, and teaching content (and the sequence in which it is presented) according to the needs of learners and subsequently guiding students to engage in meaningful learning activities. Personalized learning aims to improve students' learning by meeting their diverse needs, and research has indicated that students learn more effectively when teaching meets their needs (Benedict, 2010; Lin et al., 2016). Thus, personalized learning can be regarded as an alternative to traditional models because it focuses on providing guidance and addressing knowledge gaps on the basis of students' current level of understanding (Johnson & Samora, 2016); notably, teachers have been increasingly adopting this approach in remedial tutoring for students (Foshee et al., 2016; Hsieh et al., 2013; Lin et al., 2016). Leveraging rich and diverse learning data, artificial intelligence and machine learning techniques can be used to analyze and predict student learning performance or to reveal their learning patterns at an early stage (Lu et al., 2018; Marbouti et al., 2016). Personalized learning based on learning analytics does not merely involve the assessment of learning and performance, but it can also improve learner engagement in the learning process (Bernacki et al., 2021). Siemens and Gašević (2012) defined learning analytics as measuring, collecting, analyzing, and reporting data about learners and their contexts to understand and optimize learning and the contexts in which it occurs. Personalized intervention applied in after-class tutoring has developed into an effective approach to improving students' learning effectiveness (Zhang et al., 2020).

In personalized intervention, students' learning status in terms of target concepts must first be predicted or identified; subsequently, they are provided with personalized learning adjustment suggestions. The personalized intervention proposed in this study first determines the students' mastery of concepts and then provides students with relevant remedial materials according to their learning status to help them review what they have learned, thereby improving students' conceptual proficiency. We used natural language processing (NLP) technologies, namely bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018) and generative pretrained transformer (GPT-2) (Radford et al., 2018), for the generation of personalized intervention remedial learning content according to the status of students. Because self-regulated learning (SRL) can be used to explore students' cognition during the programming process (Zimmerman, 1989; Zimmerman, 1990), we used the learning strategy subscale of the motivated strategies for learning questionnaire (MSLQ) (Pintrich, 1991; Pintrich et al., 1993) to group students; subsequently, we explored the impact of the personalized intervention on students with different learning strategies. The following research questions guided this study:

- RQ1: For different learning concepts, what are the key learning features that affect students' mastery of concepts?
- RQ2: Can personalized intervention tutoring improve students' learning performance to a greater extent than traditional classroom tutoring?
- RQ3: Can learning strategies and online learning features effect on learning performance?
- RQ4: What is the impact of personalized intervention tutoring on the learning performance of students with different learning strategy abilities?

2. Literature review

2.1. SRL in a programming course

In SRL, learners examine their learning behavior and make adjustments to achieve learning goals and tasks. According to Zimmerman (1990), the first component of SRL is the student's metacognitive strategies for planning, self-monitoring, and controlling learning at various stages of the learning process. The second component of SRL is students' motivation and emotional processes for engaging in learning tasks (Pintrich, 1999; Zimmerman, 1989); for this component, researchers can use variables such as self-efficacy, task value, intrinsic goal orientation, and test anxiety as assessment dimensions. The third component of SRL is student behavioral processes, such as how students create and structure their learning environment (Zimmerman, 1989). The preceding description highlights that SRL is an active and constructive process by which students set goals for their learning and attempt to monitor, regulate, and control their cognition, motivation, and behavior in the learning process (Pintrich, 2000).

Problem-solving is a key skill required for the 21st century (Lai & Hwang, 2014), and computer programming has emerged as a popular subject for developing such skills. Because of the prominence of computer technology in modern society, the programming process is increasingly being used in educational settings to cultivate students' problem-solving ability. In the programming process, the problem is the main focus and the goal is to solve the problem through computer programming. Programming for problem-solving has become an essential ability for learners to construct the knowledge needed to perform new tasks; in this learning process related to new tasks, we can consider students' self-adaptive learning to observe their cognitive construction processes. In SRL, student cognition is the main element, and the relationship between students' metacognition, motivation, and behavioral participation as they complete learning tasks can be explored. Thus, in the learning process of programming for problem-solving, we can use SRL and adopt students' cognition as the main observation; we can then further explore each student's learning status in the programming process. In recent years, some researchers have begun to guide students to learn programming languages through SRL and investigate the correlation between students' SRL ability and learning performance (Cheng, 2021; Song et al., 2021).

2.2. Personalized intervention for at-risk students

In the field of teaching, at-risk students can be helped in a timely manner through early identification. Artificial intelligence technology and machine learning are increasingly being used to construct models for identifying at-risk students or learning patterns in educational settings. These models have mostly been used to predict student learning trends, student performance levels (Villagrá-Arnedo et al., 2017), whether students will pass or fail (Huang et al., 2020), and their academic scores (Lu et al., 2018). Most predictive models adopt machine learning methods, such as decision trees, linear regression, and support vector machines. These models can be used to identify at-risk students and determine when human intervention is needed as well as to provide assistance to such students. Although models can be used to identify at-risk students, the tutoring of these students to help improve their academic performance is still handled by teachers. Research on automatic generation of personalized learning suggestions for students with different learning trends, with the aim of improving the subsequent learning performance of at-risk students is limited. Therefore, the development of interventions for at-risk students has become a popular topic.

Personalized intervention refers to the provision of unique interventions according to the distinct learning characteristics and learning states of students (Zhang et al., 2020). In early education research, most personalized interventions were based on teachers' observations of students. Because of the effort involved, teachers cannot implement personalized interventions for all students in a given class. Fortunately, learning analysis based on big data analysis technology has grown rapidly, thereby facilitating the development of personalized interventions. For example, on the basis of learners' learning styles and cognitive abilities, Yi et al. (2017) implemented personalized interventions in an online learning environment; the intervention involved sending notification messages and emails to all students in the classroom. Most current personalized interventions send personalized learning messages according to students' personal characteristics (e.g., learning styles); however, personalized interventions can also be adopted for remedial materials aimed at addressing learning difficulties. Therefore, we developed a system that provides personalized remedial learning material content according to students' learning proficiency.

2.3. Automatic generation of remedial materials in personalized intervention

After learning, students may still have gaps in their understanding of certain concepts. Remedial materials are designed to help students fill those gaps in a timely manner (Bauman & Tuzhilin, 2018). Big data–based learning analysis is increasingly being adopted to provide students with personalized remedial materials to help them master subject content (Bauman & Tuzhilin, 2018; Bethard et al., 2012). Although the use of personalized remedial materials is receiving growing research attention, related research has continued to rely on teachers to generate such materials in advance to address students' learning difficulties. The automatic generation of remedial teaching materials for personalized intervention would thus be of considerable use to educators.

NLP toolkits for text processing have become a popular research topic. Novel NLP tools introduced recently include the Natural Language Toolkit (NLTK), TextBlob, and FLAIR. In NTLK, before the text is applied for different tasks, text processing, such as tokenization, tagging, parsing, is performed (Bird, 2006). The main goal of TextBlob is to calculate the polarity and subjectivity of text. FLAIR allows the application of different models, such as named entity recognition and part-of-speech tagging, to user tasks (Akbik et al., 2019). However, the aforementioned three toolkits are used mostly for sentiment analysis and text classification; they are not suitable for applications involving automatic text content generation.

Models pretrained on large corpora can learn common language representations from large amounts of unlabeled textual data; these language representations can then be used for applications in other natural language tasks, thereby avoiding training new models from scratch (Lu et al., 2021; Yang, 2021). Users only need to fine-tune the pretrained NLP model to be able to complete various downstream NLP tasks, with no need for users to design new neural network architectures for different tasks, thus substantially improving the training efficiency (Radford et al., 2018). The BERT pretrained model allows for the intuitive building of the model pipeline and for modeling many downstream tasks. Thus, researchers only need to input specific inputs and outputs into BERT; the model then fine-tunes all parameters during the training process (Devlin et al., 2018). This mechanism enables researchers to represent the steps of BERT in a more interpretable and localizable manner (Tenney et al., 2019). Therefore, we used two well-known pretrained models—BERT (Devlin et al., 2018) and GPT (Radford et al., 2018)—to automatically generate remedial materials.

3. Methodology

3.1. Participants and instruments

We conducted an 8-week experiment beginning in March 2022 to investigate whether the proposed personalized intervention tutoring could improve students' reviewing performance as they learned Python. The study participants are 78 students in two classes at a university in northern Taiwan; the two classes will be assigned as an experimental class and a control class. There were 36 students in the class of the experimental group, including 16 males and 20 females; the class of the control group consisted of 42 students, consisting of 28 males and 14 females. The only difference between the two groups was that, one week before the test, the students in the experimental group reviewed Python knowledge through personalized intervention tutoring, whereas those in the control group received traditional classroom tutoring.

The MSLQ is commonly used to measure students' self-regulation ability; it comprises two subscales—learning motivation and learning strategy—with items being rated on a five-point Likert scale (Pintrich, 1991, Pintrich et al., 1993). Thus, the instructor measured students' self-regulation ability by using the learning strategy subscale in the MSLQ. The learning strategy subscale of the MSLQ contains a total of 31 items on rehearsal (4 items), elaboration (6 items), organization (4 items), critical thinking (5 items), metacognitive self-regulation (12 items), time and study environment (8 items), effort regulation (4 items), peer learning (3 items), and help seeking (4 items). For the experimental group, the Cronbach reliability coefficients for our data in the pretest and posttest were .909 and .912, respectively. For the control group, these values were .884 and .948, respectively.

To reveal students' Python programming background knowledge, we conducted a programming knowledge test during the first week of the course. Only 41 and 29 students in the control and experimental groups, respectively, completed this test. No significant difference was observed between the experimental and control groups (Table 1); that is, the students in the two groups had substantial and comparable background knowledge of Python programming at the beginning of the course.

Table 1. Python programming background knowledge in the control and experimental groups

Group	*N*	Mean	*SD*	*t*-value
G_C	41	42.78	17.95	.70
G_E	29	39.97	14.67	

Note. Independent sample *t* tests; $^*p < .05$.

In this study, ethics review approval (NTU-REC No.: 202005ES032) was granted by the National Taiwan University Research Ethics Committee. In addition, all participants in this study were informed that their learning event data would be collected, and all participants signed an informed consent form.

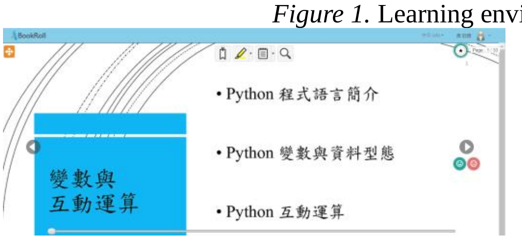
3.2. Experimental design of learning activities

We constructed a Python integrated learning environment, comprising an e-book reading system called BookRoll, a review system, and an assessment system, specifically for students in non information fields. BookRoll is an online e-book learning platform developed by the School of Social Information at Kyoto University (Japan). Because after-class review has a considerable impact on academic performance, the review system established in this study provides students with questions through cloze and short-answer questions. Finally, the assessment system allows students to tests their knowledge of key concepts after class through the use of multiple-choice questions; in this manner, students obtain an overview of their proficiency in key concepts.

In the Python programming course, the principle concepts are program output and input (C_1), strings (C_2), lists (C_3), and selection logic (C_4). The learning activities of the experimental and control groups are presented in Figure 2. In Week 1, students completed a test on Python programming background knowledge as well as a pretest of SRL ability measured using the MSLQ learning strategy subscale; during this week, students also received an introduction on how to operate the various learning environments (see Figure 1(a) to 1(d)) adopted in this study. Students in both groups engaged in two types of learning activities, namely programming learning activities and programming review activities, with only the review activities differing between the groups: The control group received traditional classroom tutoring review activities, whereas the experimental group received personalized intervention review activities.

From Weeks 2 to 5, the students in the experimental and control groups learned the basic concepts of Python programming (C_1–C_4) through three programming learning activities: programming instruction, programming knowledge review, and programming knowledge self-assessment. For the programming instruction activity, the teacher first uploads learning materials to BookRoll (see Figure 1(a)) and subsequently explains Python programming concepts in the classroom. In the programming knowledge review, students can review the key content of the learning materials through the two types of questions in the review system: cloze (see Figure 1(b)) and short answer (see Figure 1(c)). In the programming knowledge self-assessment, students use the assessment system (see Figure 1(d)) to confirm their mastery of key concepts through multiple-choice questions.

Figure 1. Learning environments adopted in this study

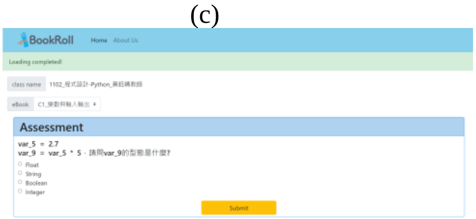


(a) BookRoll

(b) Cloze question in the review system

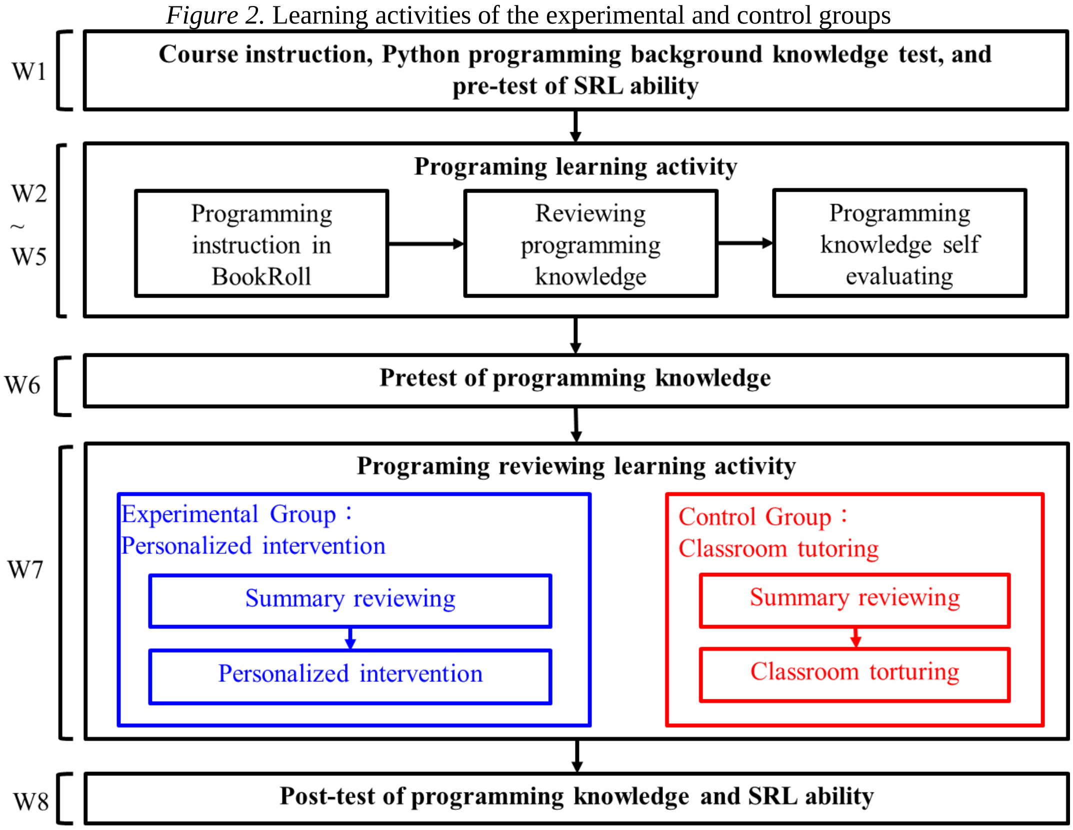
(c)

(c) Example short-answer question in the review system



(d) Assessment system

Figure 2. Learning activities of the experimental and control groups

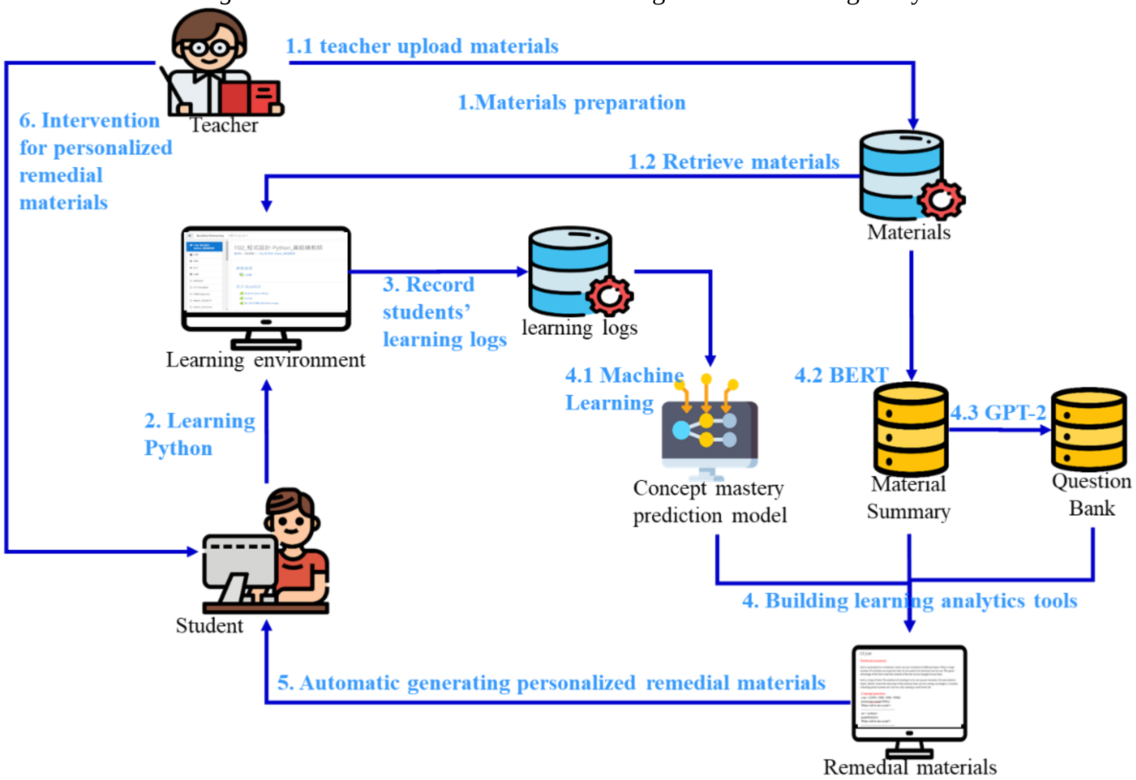


To help students prepare for the programming knowledge exam, we conducted various programming review activities for both groups in Week 7. First, the students in the experimental and control groups completed the same summary review activity, which focused on the key content of the four concepts (C_1 to C_4) learned in the course. Subsequently, the students in the experimental group received the personalized intervention (see Figure 2), as described in Section 3.3, whereas those in the control group received traditional classroom tutoring where students could clarify their doubts by asking the teacher questions and receiving related guidance. To measure the effect of the different review activities, a pretest of programming knowledge was conducted in Week 6, and then a posttest of programming knowledge and SRL ability was conducted in Week 8. The pretest and posttest questions are presented in Appendix A.

3.3. Personalized intervention process

To provide teachers with reference information regarding students' learning behavior during the tutoring intervention process, we designed a personalized intervention tutoring process that involves 6 steps based on learning analysis (see Figure 3). Step 1 involves material preparation. In Step 1.1, teachers upload the learning content to the materials database before class. In Step 1.2, the learning content is sent from the materials database to the online learning environments such as BookRoll. In Step 2, students learn Python programming in BookRoll under the guidance of a teacher. Step 3 involves students using the online learning environment, with their learning events collected and recorded in the learning log database. Step 4 involves the construction of learning analysis tools; in Step 4.1, a concept mastery prediction model is constructed using machine learning methods (see Section 3.5); in Step 4.2, a summary of materials is automatically generated using BERT and stored in the materials summary database (see Section 3.4.1); and in Step 4.3, questions are automatically generated using GPT-2 and stored in the question bank (see Section 3.4.2). On the basis of the learning analysis tool established in Step 4, in Step 5, personalized learning materials are automatically generated and sent to each student. Subsequently, in Step 6, teachers conduct personalized tutoring interventions for students according to the personalized materials automatically generated in Step 5.

Figure 3. Personalized intervention tutoring based on learning analytics



3.4. Automatic generation process of personalized remedial materials

In Step 5 of the personalized intervention process (see Figure 3), personalized remedial content is generated for each student; the steps in this content generation process are illustrated in Figure 4. The personalized remedial materials are obtained using two components: learning diagnosis and remedial content generation. Learning diagnosis involves three analysis systems: concept mastery prediction (see Section 3.5), self-evaluation results in the assessment system (see Section 3.2), and the extracted features (presented in Appendix B). The primary goal of the learning diagnosis component is to place each student into one of the following performance categories: proficient, practical improvement, and nonproficient. In the remedial content generation component, remedial content is automatically generated through two retrievers, one related to questions and the other to material summaries. The question retriever is designed to retrieve the practice questions automatically generated by the GPT-2 from the question bank; the summary retriever is designed to retrieve material summaries automatically extracted by BERT. Figure 5 depicts a screenshot of the content of the automatically generated personalized remediation material.

Figure 4. Steps in the automatic generation of personalized remedial materials

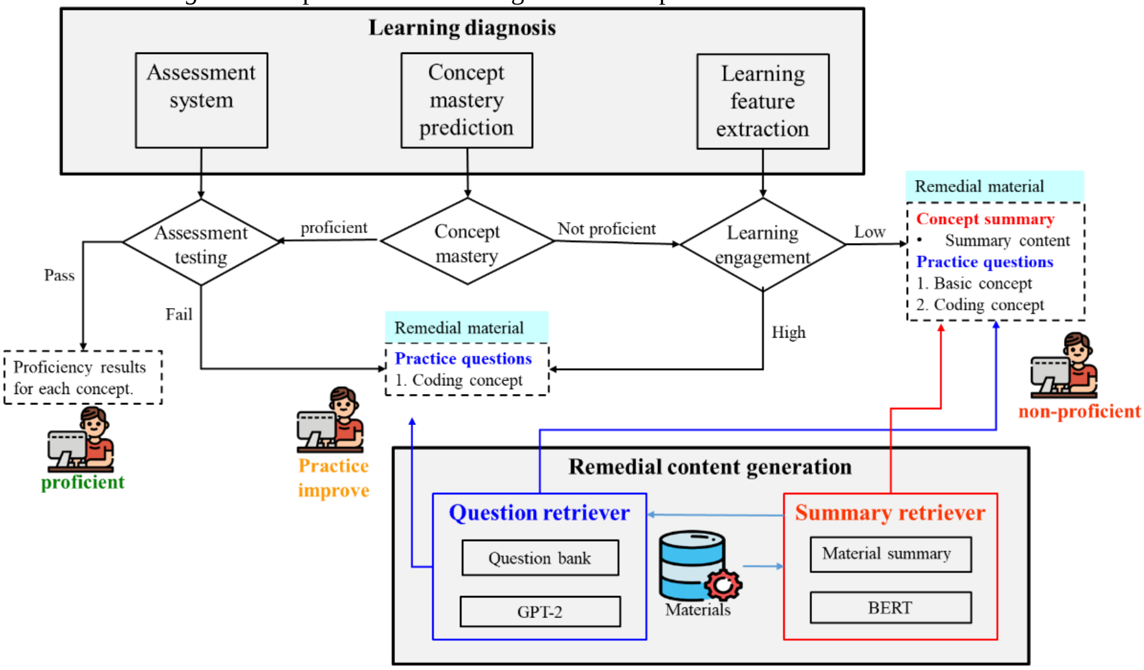
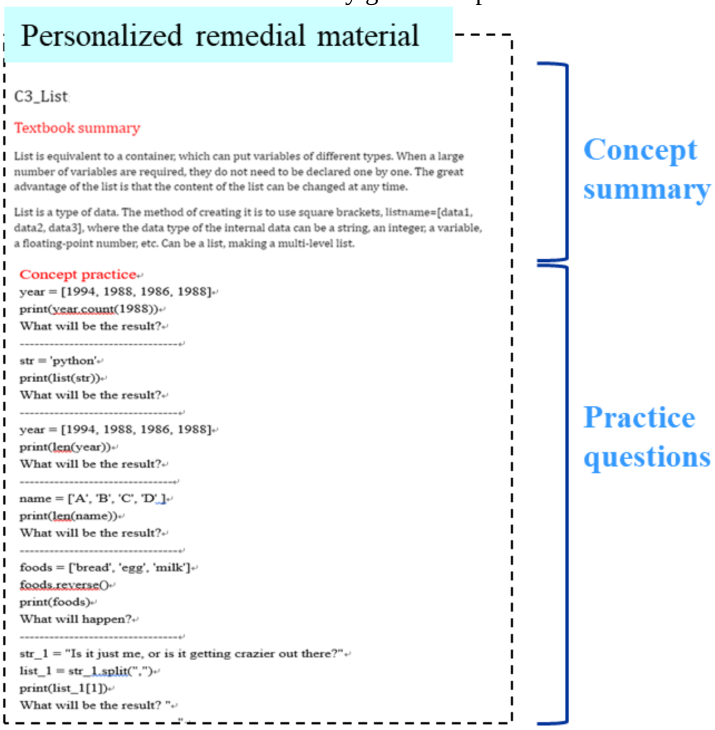


Figure 5. Screenshot of the automatically generated personalized remediation material



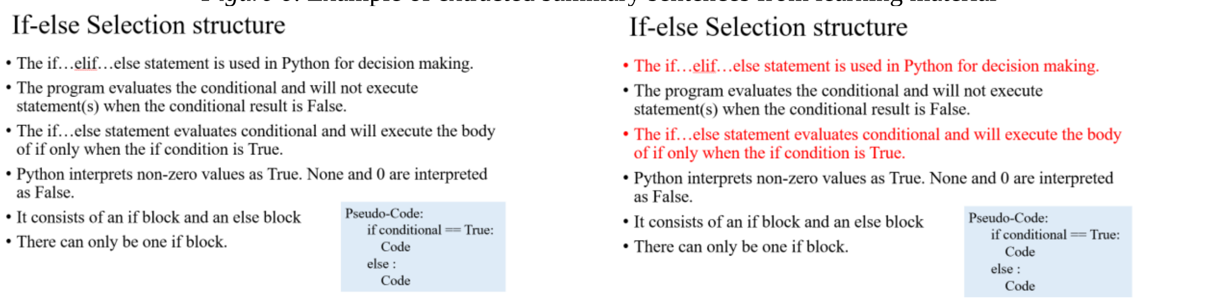
Proficient students are those students who are predicted to be proficient in learning concepts by the concept mastery prediction model and who receive a passing result in the assessment system. Because of their proficiency in learning concepts, these students do not need to receive additional supplementary materials, but this study will give proficiency results for each concept, thereby encouraging students to continue to study hard. Nonproficient students are those predicted to be nonproficient by the concept mastery prediction model and who have low values for extracted features. These low values indicate that these students lack active online learning behaviors. Students' poor online learning behaviors may lead to a lack of understanding of the learning content, which in turn can lead to them being deemed as nonproficient by the concept mastery prediction model. To help these students rapidly review key material before the test, we provide them with summary content through BERT, guide them to organize and acquire key concepts, and provide them with practice questions including basic concept questions and coding concept questions.

Finally, students are placed in the practice improvement category in two cases: (1) they are predicted to be proficient by the concept mastery model, but they receive a failing grade in the assessment system and (2) they are predicted to be nonproficient by the concept mastery model, but they have high values for the extracted features, indicating active participation in online learning. In the second case, although the students are active in online learning, they are still predicted to be nonproficient in key concepts. Such students require additional practice to achieve proficiency, hence the name of this group. Students in the practice improvement category lack the ability to apply what they have learned to programming; therefore, they receive coding concept questions, thereby helping them develop their programming skills as they solve problems.

3.4.1. Material summary automatic generation process

The main goal of summarization is to extract the main idea of a document, generally combining relevant or important information into a concise structure (Allahyari et al., 2017). Summarization help students to not only rapidly obtain key content from learning material but also improve their review effectiveness. The BERT extractive summarizer model (see https://github.com/dmmiller612/bert-extractive-summarizer) first embeds the sentences in the input text into BERT's sentence embedding vector, then uses k-means to group all sentences in the text, and finally extracts the sentences closest to the cluster centroids as summaries (Miller, 2019). The BERT extractive summarizer model allows the user to specify the number of summary sentences to generate. For the summary extraction of Python learning materials, we first sorted the textbook content from the e-book and then used the BERT extractive summarizer model to extract the summary sentences and store them in the material summary database. Figure 6(a) and 6(b) are list examples of summary sentences extracted from the learning material.

Figure 6. Example of extracted summary sentences from learning material



(a)Screenshot of learning content from e-book. (b)The extracted summary sentences are marked in red font.

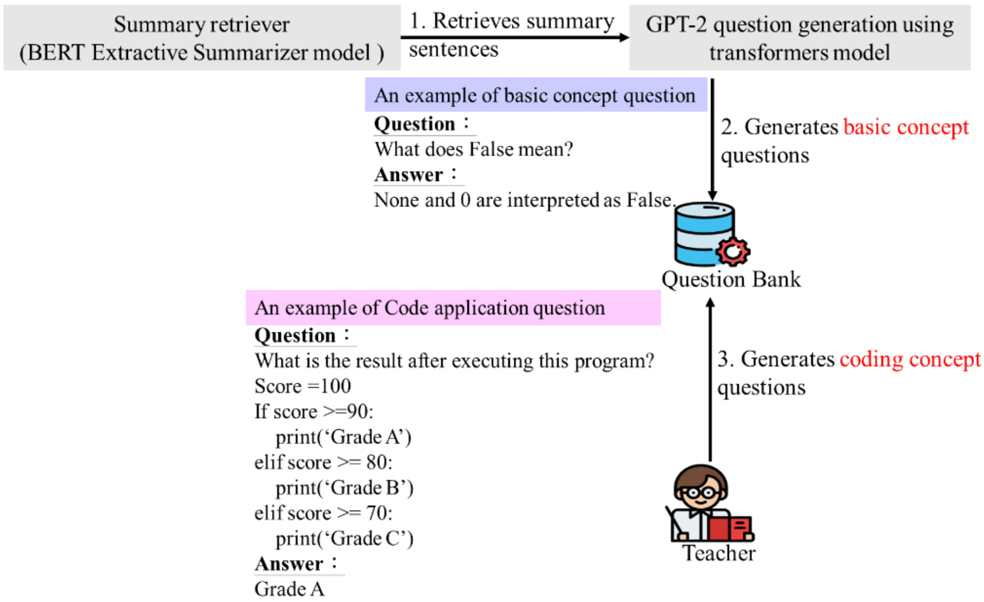
3.4.2. Question automated generation process

Proposed by OpenAI, the GPT model is a language model that predicts the next word in an incomplete sentence; after the predicted new words are added, the next word is predicted again until a complete sentence is produced. The GPT-2 language model (Radford et al., 2018) focuses on the completion of some tasks such as answering questions and generating text output. The GTP can thus serve as a sentence generator; for this reason, it has been widely used in dialogue systems, medical text simplification, and many other applications (Ghojogh & Ghodsi, 2020).

The GPT-2 model of question generation using transformers (see https://github.com/patil-suraj/question_generation#question-generation-using-transformers) aims to generate questions with pretrained transformers through simplified data preprocessing. It can generate different types of questions and answers by using three models: the single-task question generation model, multitask question generation model, and end-to-end question generation model. The goal for the model in this study was to generate questions and answers together, with the answers appearing in the text. Therefore, we adopted the single-task question generation model.

Programming knowledge includes basic conceptual knowledge and coding conceptual knowledge. Basic conceptual knowledge can be extracted from the material content of textbooks, while coding conceptual knowledge focuses on explaining programming syntax through program code. In order to improve students' review efficiency through program practice questions, this study generates basic concept questions and coding concept questions for each learning concept. Figure 7 shows two examples of generating basic concept and coded concept questions. We applied GPT-2 model of question generation using transformers, which uses a single-task question generation model to automatically generate basic conceptual questions and answers from material summaries automatically generated by the summary retriever, and handed over to the instructor to confirm and modify the GPT-2 automatic generated basic concept questions and answers. For coding concept questions, the instructor will design coding application practice questions. Finally, both basic concept questions and coded concept questions will be stored in the question bank.

Figure 7. Examples of generated questions for basic concept questions and coding application questions



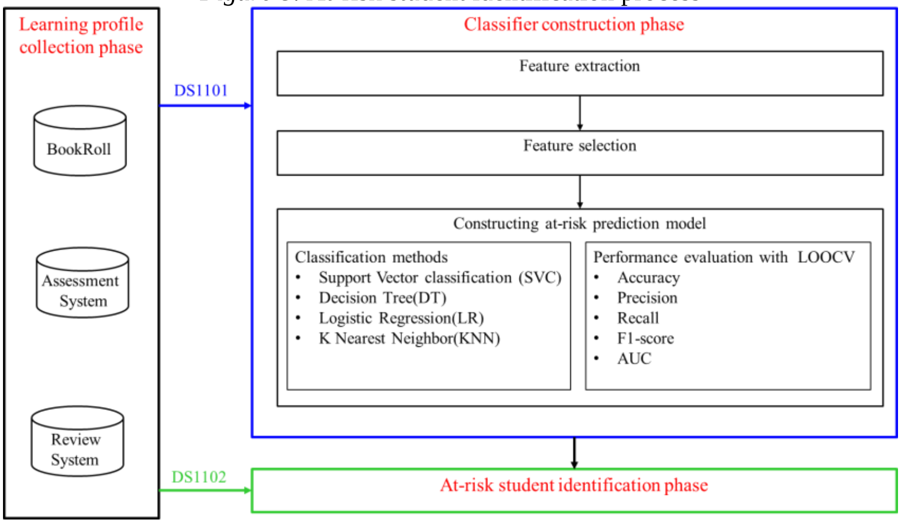
3.5. Construction process of at-risk student prediction model

To identify at-risk students in the targeted programming course, we proposed an at-risk student identification process (Figure 8), which comprises the learning profile collection, classifier construction, and at-risk student identification phases. In the learning portfolio collection phase, the learning events of students in the integrated learning environments (i.e., BookRoll, assessment system, and review system) were collected. In the classifier construction phase, with reference to the training samples, a classifier was constructed using the following steps: feature extraction, feature selection, and classifier construction. Finally, in the at-risk student identification phase, validation samples were identified through the constructed classifier.

In the learning portfolio collection phase, we collected students' learning logs from BookRoll, review system, and assessment system. In the at-risk student identification phase, the classification model is first constructed based on the training data set of DS1101, and then it is possible to predict whether the students in the data set DS1102 will pass or fail. In the classifier construction phase, we generated an at-risk student prediction model through three steps: feature extraction, feature selection, and construction. In the feature extraction step, relevant features were extracted from log data. Detailed instructions for each extracted feature are described in Appendix B. In the feature selection step, we use three selection methods—minimal redundancy maximal relevance (mRMR), chi-square test (Chi2), and relief algorithm—to identify the most relevant and powerful features from among the extracted features (Chandrashekar & Sahin, 2014). The mRMR method is used to identify the set of features in the original feature set that exhibits a high correlation with the output and a low correlation between the features themselves (Li et al., 2012); the chi-square test estimates whether a class label is independent of a feature (Jin et al., 2006); and the relief algorithm calculates a statistic for each feature that can be used to estimate feature quality or relevance to the output (Kira & Rendell, 1992).

In the construction phase of the at-risk student prediction model, we constructed prediction models by using support vector machine, decision tree, logistic regression, and k-nearest neighbor. Support vector machine is a supervised learning model that finds a separable hyperplane for samples by mapping samples to a high-dimensional space; it then predicts the class of new data samples. Logistic regression is used to construct a classification model by finding a regression line based on the probability of occurrence of sample classes (Kutner et al., 2005). K-nearest neighbor is a nonparametric classification method that establishes a classification model through the k nearest data points for a data sample (Cover & Hart, 1967). Finally, with decision trees, the focus is on building a classification model to find a tree structure that represents all known training data samples; the aim is to reveal hidden rules that identify categories based on feature values (Quinlan, 1983). To evaluate the prediction performance, this study used five metrics of accuracy, precision, recall, F1 measure, and area under the curve (AUC) (Ferri et al., 2009; Fawcett, 2004).

Figure 8. At-risk student identification process



4. Results and discussion

4.1. The critical online learning features for each programming learning concept

For the early identification of at-risk students, we used four feature selection methods to improve the prediction performance of the key learning features of each learning concept. The prediction performance of the four models is presented in Appendix C. Table 2 presents summaries of the prediction performance of classifications with the four feature selection methods for concepts C_1 to C_4. The results indicate that the classification using mRMR feature selection generally yielded the highest AUC values, ranging from 0.90 to .94. Therefore, we adopted mRMR as our feature selection method.

Table 2. Summaries of prediction performance of classifications with feature selection methods for concepts C_1 to C_4

Feature selection	Accuracy	Precision	Recall	F1	AUC
mRMR	.89~.94	.94~.95	.89~.90	.88~.90	.90~.94
Chi-Square	.78~.89	.84~.94	.78~.89	.76~.90	.69~.94
Relief	.80~.89	.91~.93	.80~.89	.84~.89	.83~.92

For exploring the key learning features that affect students' mastery of each learning concept through the features selected by the mRMR method (RQ 1). Table 3 indicates the key learning features selected by the mRMR for concepts C_1–C_4. For concepts C_1 and C_2, the key features for determining whether students have mastered the concepts were mostly related to the features extracted from BookRoll and the review system, namely review time (f_4), preview time (f_5), and correct answer rate in assessment (f_7). This is because concepts C_1 and C_2 involve relatively basic knowledge in programming; students can understand these two concepts by reading and reviewing the content of the e-book. For the content of concepts C_3 and C_4, the importance of the features associated with the assessment system (f_6, f_7) and review systems (f_8, f_9) increased accordingly. The conceptual content of concept C_3 (List) centers on the definitions of composite data types. In addition to understanding the definition of composite data type List in the textbook, students must also learn the operation and application of List. The conceptual content of concept C_4 (choice structure) centers on program logic. Students must familiarize themselves with various types of selection structures and methods through practical exercises; examples include the if...else structure, if–elif–else structure, and nested if structure. Therefore, because of the complexity of concepts C_3 and C_4, students cannot master them merely by reading or reviewing the content of the textbook. They would need to also further enhance their conceptual proficiency by practicing the questions relating to application or program tracking. That is, for students to become proficient in concepts C_3 and C_4, they must refer to the programming applications or coding track questions covered in the assessment system.

Table 3. Key learning features selected using the mRMR method for each learning concept

Concept	At-risk prediction model	Key features
C_1	SVM with mRMR	f_4, f_9, f_5, f_7
C_2	SVM with mRMR	f_4, f_1, f_3, f_7
C_3	DT with mRMR	f_7, f_9, f_8, f_1, f_6
C_4	DT with mRMR	f_7, f_6, f_9, f_1

4.2. Effect of personalized intervention tutoring approach on students' learning performance

The independent sample t-test results for the pretest and posttest are listed in Table 4. We observed no significant difference in the pretest results between the experimental and control groups ($t = -.12$, $p > .05$). This result suggests that students in the two groups achieved the same level of programming knowledge after engaging in Python programming learning activities for 4 weeks. The posttest scores of the experimental and control groups were 89.63 and 83.29 ($t = -2.44$, $p < .05$), respectively, representing a significant difference. The personalized intervention tutoring approach aims to recommend programming learning materials based on the predicted results of students' learning performance through machine learning methods. This result is consistent with previous research findings that learning resources recommended by machine learning can guide students to achieve higher learning performances (El-Bishouty et al., 2018). Accordingly, in response to RQ2, we found that compared with traditional classroom tutoring review, personalized intervention review can more effectively improve students' learning performance.

Table 4. Independent t-test results of pretest and posttest between control and experimental groups

Group	N	Pre-test		Post-test	
		Mean/SD	t-value	Mean/SD	t-value
G_C	42	74.62/14.33	-.12	83.21/15.65	-2.44*
G_E	36	75.02/13.71		89.72/6.86	

Note. $^*p < .05$.

After the programming review learning activity, students in the experimental group were asked to respond to four feedback questions to elucidate their views on the personalized intervention tutoring they received. The four feedback questions (see Appendix D) were answered using a 5-point Likert scale, ranging from 1 (*strongly disagree*) to 5 (*strongly agree*). Table 5 presents a summary of their responses. Of the 36 students in the experimental group, 30 provided feedback. The first question focused on how helpful students found the review activities conducted by the teacher. The average score for this question was 4.17, indicating that most students had positive perceptions of the classroom review activities and regarded them as effective. Notably, 16 of the 30 students who answered the question were classified as proficient. For students who are familiar with each concept, although no remedial materials will be given in this study, the familiarity evaluation results of each concept will still be provided to encourage students to continue to study hard. Since only 14 students in this study were not conceptually proficient, only these 14 students answered questions 2-4 and were used to discuss the feedback received on the remedial materials.

Table 5. Responses to the four feedback questions

Question	Number of responds	Mean/ SD	Number of selected students for each point				
			1(SD)	2(D)	3(N)	4(A)	5(SA)
Q1	30	4.17/.79	0	1	4	14	11
Q2	14	4.29/.83	0	1	0	7	6
Q3	14	4.36/.84	0	1	0	6	7
Q4	14	4.36/.84	0	1	0	6	7

Note. SD: strongly disagree; D: disagree; N: neutral; A: agree; SA: strongly agree.

Thirteen students indicated that the personalized intervention remedial materials not only helped them to review the conceptual content independently but also helped them to review the conceptual content that they were not familiar with (Table 5). This means that the students generally had a positive perception of the materials and were satisfied with personalized intervention tutoring approach. For Questions 2–4, only one student provided a rating of 2 points. The teacher interviewed this student to inquire into why he disagreed that the individualized intervention tutoring approach was effective. The student indicated that although the teacher's review activities were explained clearly, for computer programming, more time should be dedicated to practicing programming skills on the computer; moreover, he regarded conceptual learning as having low importance, which explains his disagreement with the relevant item on the questionnaire. This student argued that an individualized intervention tutoring approach that provides both conceptual review material regarding programming knowledge and personalized programming exercises would be of greater benefit to students' programming skills.

4.3. Impact of learning strategies and online learning features on learning performance

For the experimental and control groups, only 35 and 32 students, respectively, completed both the pretest and posttest for student's SRL ability. We used the SRL ability pretest result as a covariate and used analysis of covariance (ANCOVA) to identify significant differences between the two groups in the posttest of SRL ability; the ANCOVA results are presented in Table 6. For the rehearsal ($F = 8.15$, $p < .05$), critical thinking ($F = 11.93$, $p < .05$), metacognitive self-regulation ($F = 12.24$, $p < .05$), effort regulation ($F = 10.42$, $p < .05$), and peer learning ($F = 4.15$, $p < .05$) dimensions, the students in the experimental group scored significantly higher than those in the control group. These results indicate that, after the review activities, students who engaged in personalized intervention activities had significantly higher abilities in the aforementioned five learning strategies than those who engaged in traditional classroom tutoring activities. That is, personalized tutoring intervention was more effective than traditional classroom tutoring activities in improving students' abilities in the aforementioned five learning strategies.

Table 6. Analysis of covariance results of posttest SRL ability for control and experimental groups.

MSLQ	Group	*N*	Mean/*SD* of pre-test	Post-test Mean/*SD*	Post-test Adjusted Mean	Post-test Std. Error	*F*
Rehearsal	G_C	32	3.80/.42	4.10/.58	3.61	.094	8.153**
	G_E	35	3.93/.34	4.14/.44	3.99	.090	
Elaboration	G_C	32	3.68/.44	4.12/.57	3.74	.085	2.912
	G_E	35	3.76/.50	4.09/.49	3.94	.081	
Organization	G_C	32	3.57/.36	3.76/.40	3.71	.095	1.057
	G_E	35	3.45/.30	3.71/.30	3.85	.091	
Critical thinking	G_C	32	3.56/.47	3.72/.55	3.06	.081	11.927***
	G_E	35	3.35/.78	3.74/.76	4.00	.077	
Metacognitive self-regulation	G_C	32	3.62/.48	3.67/.43	3.45	.051	12.243***
	G_E	35	3.53/.51	4.05/.68	3.70	.048	
Time and study environment	G_C	32	3.72/.45	3.95/.51	3.55	.047	0.571
	G_E	35	3.60/.47	3.94/.64	3.60	.045	
Effort regulation	G_C	32	3.58/.33	4.02/.52	3.57	.084	10.423**
	G_E	35	3.42/.29	3.72/.35	3.93	.080	
Peer learning	G_C	32	3.49/.23	3.65/.33	3.71	.136	4.149*
	G_E	35	3.54/.42	3.94/.56	3.32	.131	
Help seeking	G_C	32	3.63/.75	3.32/.92	3.37	.077	3.705
	G_E	35	3.37/.39	3.55/.51	3.58	.073	

Note. $^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$.

In terms of the effect that students' SRL ability has on their academic performance, Song, Hong, and Oh (2021) found that although students' learning strategy ability was not significantly correlated with their academic performance, the online learning features in programming courses, such as the number of chosen program tasks and overall code-run trials, were significantly correlated with students' learning abilities. Therefore, we examined the correlation between online learning features (listed in Table 6) and learning strategy ability as well as the correlation between online learning features and learning performance; the results are presented in Table 7. Learning Features f_3 and f_8 were significantly correlated with the pretest result for learning performance ($r = .37$, $p < .05$ for f_3; $r = .48$, $p < .01$ for f_8). Moreover, although f_8 was not related to any of the learning strategies, feature f_3 was significantly related to LS_1 (rehearsal), LS_3 (organization), LS_5 (metacognitive self-regulation), LS_6 (time and study environment), LS_7 (effort regulation), LS_8 (peer learning), and LS_9 (help seeking).

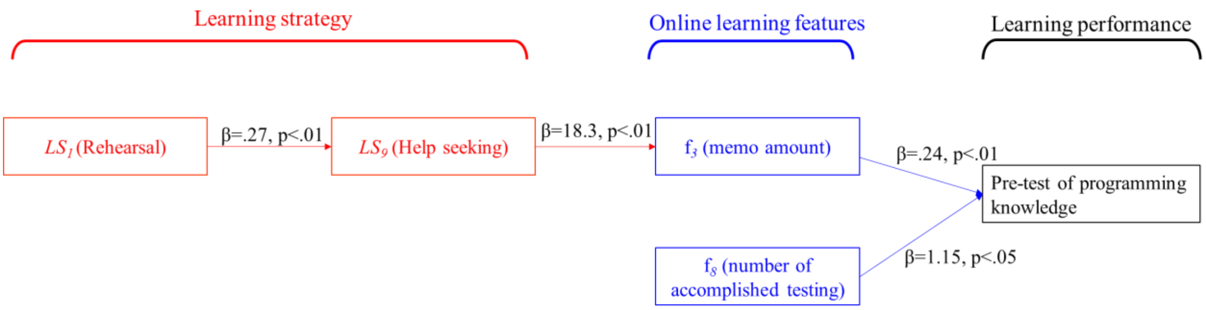
On the basis of the correlation results between online learning features and learning strategies (Table 7), we used multiple regression analysis to explore the correlations among online learning features, learning strategies, and learning performance (shows in Appendix E). Our results indicated that features f_3 (β = .24, p < .01) and f_8 (β = 1.15, p < .05) were significant predictors of pretest learning performance, LS_9 (help seeking) (β = 18.30, p < .01) was a significant predictor of feature f_3, and LS_1 (rehearsal) (β = .27, p < .05) was a significant predictor of LS_9 (help seeking). In terms of learning strategies, these results suggest that rehearsal first affects help seeking, which then affects the amount of notes students add, eventually affecting their learning performance. Figure 9 presents the conceptual diagram of the present regression analysis results for predictors f_3, f_8, LS_1, and LS_9 for pretest learning performance. The results in Table 6 indicate that students in the experimental group had significantly higher values than those in the control group for the rehearsal learning strategy. In addition, the results in Table 7 reveal that the rehearsal learning strategy was a predictor of help seeking, which in turn was a predictor of memo amount (f_3); finally, the amount of notes students wrote (f_3) and number of tests they completed (f_8) were predictors of learning performance. In sum to reply RQ3, the personalized intervention tutoring approach improved students' ability in terms of help seeking by promoting their rehearsal learning strategy, which in turn prompted students to write more notes on BookRoll, ultimately leading to improved learning performance.

Table 7. Pearson correlations among the extracted features, learning strategies, and learning performance.

	f_1	f_2	f_3	f_4	f_5	f_7	f_8	f_9
LS_1 (Rehearsal)	0.008	0.166	.409*	0.142	.447**	0.164	0.054	-0.044
LS_2 (Elaboration)	0.002	0.195	0.301	0.11	0.251	0.219	-0.05	0.144
LS_3 (Organization)	0.041	.451**	.428**	0.122	0.212	0.081	0.158	0.005
LS_4 (Critical thinking)	-0.033	0.075	0.169	0.075	0.047	0.001	-0.207	-0.034
LS_5 (Metacognitive self-regulation)	0.14	0.263	.480**	0.287	0.246	0.134	0.157	0.105
LS_6 (Time and study environment)	0.318	0.185	.505**	.329*	0.309	0.304	0.297	0.191
LS_7 (Effort regulation)	0.05	0.054	.455**	.433**	-0.018	0.148	0.08	0.135
LS_8 (Peer learning)	0.324	0.218	.338*	0.208	.445**	0.308	0.285	.441**
LS_9 (Help seeking)	0.249	0.09	.495**	0.226	0.229	0.036	0.184	0.021
Pre-test	0.28	0.3	.37*	0.26	0.17	-0.03	.48**	0.15
Post-test	-0.11	0.08	-0.04	-0.13	0.03	0.09	0.08	0.21

Note. $^*p$ < .05; $^{**}p$ < .01.

Figure 9. Conceptual diagram of the regression analysis results



4.4. Effect of personalized intervention on learning performance for each learning strategy

To respond to RQ4, we used two-way analysis of variance (ANOVA) to explore the impact of the interaction of review activity and learning strategies on learning performance. Because review activity is a categorical variable pointing to either personalized intervention or traditional classroom tutoring, we adopted k-means clustering to divide students' learning strategy ability into categorical variables at three levels: high, medium, and low. Among the nine learning strategies, four had significant effects on the interaction between the posttest results for learning strategies and learning performance, namely review activity × rehearsal (F = 6.94, p = .002), review activity × elaboration (F = 4.21, p = .019), review activity × organization (F = 6.58, p = .002), and review activity × critical thinking (F = 9.53, p = .000). The present two-way ANCOVA results are presented in Table 8, and the descriptive statistics for the high, medium, and low groups for rehearsal, elaboration, organization, and critical thinking are presented in Appendix F. Our results indicated that the review activity approach, four learning strategies (rehearsal, elaboration, organization, and critical thinking), and the interaction between the learning strategies and review activity had significant effects on students' learning performance.

Table 8. Two-way ANOVA results for review activity and four learning strategies—rehearsal, elaboration, organization, and critical thinking

Variables	*SS*	*df*	*MS*	*F*	Significant
Review activity	2119.55	1	2119.55	18.53	.000
Rehearsal	2608.87	2	1304.44	11.40	.000
Review activity * Rehearsal	1588.77	2	794.38	6.94	.002
Review activity	1847.91	1	1847.91	14.30	.000
Elaboration	1498.82	2	749.41	5.80	.005
Review activity * Elaboration	109.13	2	545.06	4.21	.019
Review activity	2071.90	1	2071.90	16.90	.000
Organization	2216.64	2	1108.32	9.04	.000
Review activity * Organization	1613.81	2	806.91	6.58	.002
Review activity	212.80	1	212.80	18.52	.000
Critical Thinking	1927.18	2	963.59	8.41	.001
Review activity * Critical Thinking	2183.45	2	1091.72	9.53	.000

Table 9. Simple main-effect analysis of learning performance in terms of the four learning strategies

Variables	*SS*	*df*	*MS*	*F*-value	Post-Hoc
Review activity					
(G_C)Review with traditional class tutoring	3351.44	2	1675.72	9.77^{**}	$Re_L = Re_M > Re_H$
(G_E)Review with personalized intervention	10.09	2	5.05	1.07	
Learning strategy: Rehearsal					
Re_L	31.07	1	31.07	.92	
Re_M	273.97	1	273.97	.10	
Re_H	2747.76	1	2747.76	9.65^{**}	$G_E > G_C$
Review activity					
(G_C)Review with traditional class tutoring	233.11	2	1165.06	5.89^{**}	$El_L = El_M > El_H$
(G_E)Review with personalized intervention	51.21	2	25.60	.53	
Learning strategy: Elaboration					
El_L	9.20	1	9.20	.28	
El_M	173.23	1	173.22	1.66	
El_H	216.00	1	216.00	7.39^{*}	$G_E > G_C$
Review activity					
(G_C)Review with traditional class tutoring	2734.82	2	1367.41	7.30^{**}	$Or_L = Or_M > El_H$
(G_E)Review with personalized intervention	127.22	2	63.61	1.36	
Learning strategy: Organization					
Or_L	104.17	1	104.17	2.688	
Or_M	216.75	1	216.75	2.207	
Or_H	2700	1	2700	1.15^{**}	$G_E > G_C$
Review activity					
(G_C)Review with traditional class tutoring	3422.29	2	1711.15	1.08^{***}	$CT_L = CT_M > CT_H$
(G_E)Review with personalized intervention	19.95	2	9.98	.20	
Learning strategy: Critical Thinking					
CT_L	105.94	1	105.94	.98	
CT_M	5.98	1	5.98	.13	
CT_H	3217.47	1	3217.47	12.36^{**}	$G_E > G_C$

Note. $^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$.

On the basis of the results in Table 8, we conducted a simple main effects analysis to examine the effect of review activities on the learning performance of students with different SRL ability levels; the results are listed in Table 9. In the traditional classroom tutoring, students with different ability levels in rehearsal, elaboration, organization, and critical thinking exhibited significantly different learning performances. In the personalized intervention, no significant differences were observed in students' learning performance at different ability levels for the four aforementioned learning strategies. The post hoc results of traditional classroom tutoring revealed that students with medium and low ability levels in rehearsal ($F = 9.77$, $p < .01$, $Re_L = Re_M > Re_H$), elaboration ($F = 5.89$, $p < .01$, $El_L = El_M > El_H$), organization ($F = 7.30$, $p < .01$, $Or_L = Or_M > El_H$), and critical thinking ($F = 1.08$, $p < .001$, $CT_L = CT_M > CT_H$) had significantly higher learning performance than students with high ability levels in these four learning strategies. This result suggests that students with strong learning strategy abilities may need more review content to maximize their learning performance after the review activity. This result also confirms that personalized intervention activities can help students with high learning strategy ability maximize their learning performance through the provision of additional review content; this explains why the experimental group students at all three learning strategy levels did not differ significantly in learning performance.

We subsequently examined the effect of different learning strategy levels on the learning performance prompted by the two review activities; for high-level rehearsal ($F = 9.62$, $p < .01$, $G_E > G_C$), elaboration ($F = 7.39$, $p < .05$, $G_E > G_C$), organization ($F = 1.15$, $p < .01$, $G_E > G_C$), and critical thinking ($F = 12.36$, $p < .01$, $G_E > G_C$), we observed significant differences in learning performance between the experimental and control groups (see Table 9). However, no significant difference was noted in students' learning performance between the two groups for low- and medium-level learning strategies. These results suggest that students with strong learning strategy abilities can achieve significantly higher learning performance through personalized intervention activities than through traditional class tutoring activities.

5. Conclusions

Artificial intelligence and machine learning technology has stimulated the development of personalized interventions in remedial coaching. We proposed a personalized intervention approach based on artificial intelligence technology for use in a computer programming course. Our results indicate that after the review activity, students who received personalized intervention had significantly higher learning performance than those who merely received class tutoring. This result confirms the effectiveness of the personalized intervention approach in helping students to review content.

Because SRL can be used to explore students' cognition during problem-solving as part of programming learning, we also considered how students' SRL abilities would influence their learning performance under the proposed personalized intervention approach. Our results reveal that the proposed personalized intervention prompted improvements in the following learning strategies: rehearsal, critical thinking, metacognitive self-regulation, effort regulation, and peer learning. We found that, although students' ability in each learning strategy was not directly related to learning performance, the learning strategies of rehearsal and help seeking indirectly affected learning performance through Learning Feature f_3 (memo amount).

In terms of the interaction effect between individualized intervention and learning strategies on learning performance, we observed significant results for rehearsal, refinement, organization, and critical thinking. Students with strong abilities in these four learning strategies achieved higher learning performance in the personalized intervention approach than in the class tutoring approach. That is, for such students, the additional review content provided by the personalized intervention effectively supported their reviewing, thereby improving their learning performance.

5.1. Limitation

The participants in this study were mainly students from noninformation fields; thus, the curriculum of this course would not be suitable for students majoring in computer science. Moreover, the proposed personalized intervention review activity focuses on programming-related knowledge, with the assessment of learning performance also focusing on programming concepts; that is, students' actual program coding ability was not tested. Therefore, the proposed intervention may require some modifications for effectively improving students' coding ability. The participants in this manuscript are only the number of students in two classes (the students in the experimental group and the control group were 36 and 42, respectively), and subsequent large-scale empirical studies are need to be verified.

Acknowledgment

This work was supported by grants from the Taiwan Ministry of Science and Technology, including MOST-111-2410-H-008-010-MY3 (its Research Ethics Committee case number is 202205ES065), MOST-109-2511-H-008-007-MY3 (its Research Ethics Committee case number is 202005ES032), MOST-108-2511-H-008-009-MY3 (its Research Ethics Committee case number is 201812ES028), and the Taiwan Ministry of Education.

References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An Easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics* (pp. 54-59).

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). *Text summarization techniques: A Brief survey*. https://doi.org/10.48550/arXiv.1707.02268

Bauman, K., & Tuzhilin, A. (2018). Recommending remedial learning materials to students by filling their knowledge gaps. *MIS Quarterly, 42*(1), 313-332.

Benedict, N. (2010). Virtual patients and problem-based learning in advanced therapeutics. *American Journal of Pharmaceutical Education, 74*(8), 1-5.

Bernacki, M. L., Greene, M. J., & Lobczowski, N. G. (2021). A Systematic review of research on personalized learning: Personalized by whom, to what, how, and for what purpose (s)? *Educational Psychology Review, 33*(4), 1675-1715.

Bethard, S., Okoye, I., Hang, H., Sultan, A., Martin, J. H., & Sumner, T. (2012). Identifying science concepts and student misconceptions in an interactive essay writing tutor. In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications* (pp. 12-21). Association for Computational Linguistics.

Bird, S. (2006). NLTK: The Natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions* (pp. 69-72). Association for Computational Linguistics.

Chandrashekar, G., & Sahin, F. (2014). A Survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16-28.

Cheng, G. (2021). The Effects of self-regulated learning strategies on computer programming achievement in teacher education. *Frontiers in Education Technology, 4*(4), 37-58.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory, 13*(1), 21-27.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding.* https://doi.org/10.48550/arXiv.1810.04805

El-Bishouty, M., Aldraiweesh, A., Alturki, U., Tortorella, R., Yang, J., Chang, T., Sabine, G., & Kinshuk. (2018). Use of Felder and Silverman learning style model for online course design. *Educational Technology Research and Development, 67*, 161–177.

Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. *Machine learning, 31*(1), 1-38.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An Experimental comparison of performance measures for classification. *Pattern recognition letters, 30*(1), 27-38.

Foshee, C. M., Elliott, S. N., & Atkinson, R. K. (2016). Technology-enhanced learning in college mathematics remediation. *British Journal of Educational Technology, 47*(5), 893-905.

Ghojogh, B., & Ghodsi, A. (2020). Attention mechanism, transformers, BERT, and GPT: Tutorial and survey. https://doi.org/10.31219/osf.io/m6gcn

Hsieh, T.-C., Lee, M.-C., & Su, C.-Y. (2013). Designing and implementing a personalized remedial learning system for enhancing the programming learning. *Journal of Educational Technology & Society, 16*(4), 32-46.

Huang, A. Y., Lu, O. H., Huang, J. C., Yin, C. J., & Yang, S. J. (2020). Predicting students' academic performance by using educational big data and learning analytics: Evaluation of classification methods and learning logs. *Interactive Learning Environments, 28*(2), 206-230.

Jin, X., Xu, A., Bie, R., & Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using Sage gene expression profiles. In J. Li, Q. Yang, & A. H. Tan (Eds.), *Data Mining for Biomedical Applications. BioDM 2006. Lecture Notes in Computer Science (Vol. 3916)*. Springer. https://doi.org/10.1007/11691730_11

Johnson, D., & Samora, D. (2016). The Potential transformation of higher education through computer-based adaptive learning systems. *Global Education Journal, 2016*(1), 1-17.

Kira, K., & Rendell, L. A. (1992). A Practical approach to feature selection. In *Machine learning proceedings 1992* (pp. 249-256). Elsevier.

Kohavi, R. (1995). A Study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI95: Proceeding of International Joint Conference on Articial Intelligence, 14*(2), 1137-1145.

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models*. McGraw-Hill New York.

Lai, C. L., & Hwang, G. J. (2014). Effects of mobile learning time on students' conception of collaboration, communication, complex problem–solving, meta–cognitive awareness and creativity. *International Journal of Mobile Learning and Organization, 8*(3-4), 276-291.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A Data perspective. *ACM computing surveys (CSUR), 50*(6), 1-45.

Li, B. Q., Hu, L. L., Chen, L., Feng, K. Y., Cai, Y. D., & Chou, K. C. (2012). Prediction of protein domain with mRMR feature selection and analysis. *PloS one, 7*(6), e39308. https://doi.org/10.1371/journal.pone.0039308

Lin, C. C., Guo, K. H., & Lin, Y. C. (2016). A Simple and effective remedial learning system with a fuzzy expert system. *Journal of Computer Assisted Learning, 32*(6), 647-662.

Lu, O. H., Huang, A. Y., Huang, J. C., Lin, A. J., Ogata, H., & Yang, S. J. (2018). Applying learning analytics for the early prediction of Students' academic performance in blended learning. *Educational Technology & Society, 21*(2), 220-232.

Lu, O. H., Huang, A. Y., Tsai, D. C., & Yang, S. J. (2021). Expert-authored and machine-generated short-answer questions for assessing students learning performance. *Educational Technology & Society, 24*(3), 159-173.

Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education, 103,* 1-15.

Miller, D. (2019). *Leveraging BERT for extractive text summarization on lectures*. https://doi.org/10.48550/arXiv.1906.04165

Pintrich, P. R. (1991). *A Manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ).* The Regents of The University of Michigan.

Pintrich, P. R. (1999). The Role of motivation in promoting and sustaining self-regulated learning. *International journal of educational research, 31*(6), 459-47.

Pintrich, P. R. (2000). The Role of goal orientation in self-regulated learning. In *Handbook of self-regulation* (pp. 451-502). Elsevier.

Pintrich, P. R., Smith, D. A., Garcia, T., & McKeachie, W. J. (1993). Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and psychological measurement, 53*(3), 801-813.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In *Machine learning* (pp. 463-482). Springer.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

Siemens, G., & Gašević, D. (2012). Special issue on learning and knowledge analytics. *Educational Technology & Society, 15*(3), 1–2.

Song, D., Hong, H., & Oh, E. Y. (2021). Applying computational analysis of novice learners' computer programming patterns to reveal self-regulated learning, computational thinking, and learning performance. *Computers in Human Behavior, 120*, 106746. https://doi.org/10.1016/j.chb.2021.106746

U.S. Department of Education. (2010). *Transforming American education: Learning powered by technology*. Office of Educational Technology.

Villagrá-Arnedo, C. J., Gallego-Durán, F. J., Llorens-Largo, F., Compañ-Rosique, P., Satorre-Cuerda, R., & Molina-Carmona, R. (2017). Improving the expressiveness of black-box models for predicting student performance. *Computers in Human Behavior, 72*, 621–631.

Wolff, A., Zdrahal, Z., Herrmannova, D., Kuzilek, J., & Hlosta, M. (2014). Developing predictive models for early detection of at-risk students on distance learning modules. In *Machine Learning and Learning Analytics Workshop at the 4th International Conference on Learning Analytics and Knowledge (LAK14)*.

Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*(9), 2839-2846.

Yang, S. J. H. (2021). Guest editorial: Precision education - A New challenge for AI in education. E*ducational Technology & Society, 24*(1), 105–108. https://doi.org/10.30191/ETS.202101_24(1).0008

Yi, B., Zhang, D., Wang, Y., Liu, H., Zhang, Z., Shu, J., & Lv, Y. (2017). Research on personalized learning model under informatization environment. In *2017 International Symposium on Educational Technology (ISET)* (pp. 48–52). IEEE. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8005386

Zhang, J. H., Zou, L. C., Miao, J. J., Zhang, Y. X., Hwang, G. J., & Zhu, Y. (2020). An Individualized intervention approach to improving university students' learning performance and interactive behaviors in a blended learning environment. *Interactive Learning Environments, 28*(2), 231–245.

Zimmerman, B. J. (1989). A Social cognitive view of self-regulated academic learning. *Journal of educational psychology, 81*(3), 329–339. https://doi.org/10.1037/0022-0663.81.3.329

Zimmerman, B. J. (1990). Self-regulated learning and academic achievement: An Overview. *Educational psychologist, 25*(1), 3-17.

Appendix A

Pretest and posttest questions for programming knowledge

Concept: C1	
Pre-test	Post-test
Question C1_1: Which of the following variables is named incorrectly? Option 1: _name Option 2: name1 Option 3: 1name Option 4: name_ **Question C1_2:** The symbol of annotation in Python is ? **Question C1_3:** Which of the following options are not reserved words? Option 1: not Option 2: and Option 3: or Option 4: xor **Question C1_4:** A Python program is given below: print(‘\’’, ‘\’’, sep = ‘A’) What is the result after executing this program?	**Question C1_1:** Which of the following variables is named incorrectly? Option 1: _996apple Option 2: _apple Option 3: 996_apple Option 4: apple_5 **Question C1_2:** What symbols do I need to replace ? to print the ‘ symbol print(‘ ? ‘) Option 1: # Option 2: !: Option 3: / Option 4: \ **Question C1_3:** A Python program is given below: print(“=“, “=“, sep = “u”) What is the result after executing this program?
Concept: C2	
Question C2_1: A Python program is given below: mail = ‘apple@gmail’ mail = mail.split(‘@’) print(mail[1]) What is the result after executing this program? **Question C2_2:** A Python program is given below: mail = ‘meme_@gmail.com’ index = mail.find(‘@’) ans = mail[0:index:1] print(ans) What is the result after executing this program? **Question C2_3:** What are the results after executing these arithmetic formulae respectively? (Be careful: 5 and 5.0 are different answers) 7/2 float(7//2) int(7%2) **Question C2_4:** What are the results after executing these programs? (If the program cannot execute, please fill in unable to execute) L = [‘Python’, ‘Hello world’, 5] L.spilt() L[1].spilt()	**Question C2_1:** Which of the following is the return value of ‘test’.find(‘t’)? Option 1: 3 Option 2: -1 Option 3: 0 Option 4: True **Question C2_2:** Which of the following options has the meaning of squared root? Option 1: **(1/2) Option 2: *2 Option 3: //2 Option 4: **2 **Question C2_3:** A Python program is given below: temp = 7 // 2 ans = temp % 2 print(ans) What is the result after executing this program? **Question C2_4:** A Python program is given below: temp = ‘中央大學’ index = temp.find(‘大’) temp[index :] What is the result after executing this program? **Question C2_5:**

L[2].spilt()

Question C2_5:
A Python program is given below:
alphabet = ‘abcdefghijklmnopqrstuvwxyz’
n = 26
print(alphabet[n%2])
What is the result after executing this program?

Question C2_6:
Suppose s = ‘abc’, what will the output of s.find(‘z’) be?
Option 1: 0
Option 2: 2
Option 3: 1
Option 4: -1

A Python program is given below:
url = “https://www.ncu.edu.tw/tw/index.html”
print(url.count(‘/’))
What is the result after executing this program?

Concept: C3

Question C3_1:
Suppose a = [1,2,3] becomes [4,1,2,3] through a certain program, which of the following functions is needed for the program?
Option 1: None of them
Option 2: a.append(4)
Option 3: a[0] = 4
Option 4: a.insert(0,4)

Question C3_2:
Suppose a = [1,2,3], what are the results after executing the following programs respectively?
Program 1: myList[2:-1]
Program 2: myList[1][-1]

Question C3_3:
Which of the following is the functions that calculates the number of times that a particular elements is inside the list?
Option 1: .find()
Option 2: .count()
Option 3: .replace()
Option 4: .len()

Question C3_1:
A Python program is given below:
myList = [[1,2,3], [4,5,6], 7, [1, [2, 3]]]
Which of the following is the output of len(myList)?
Option 1: 5
Option 2: 10
Option 3: 3
Option 4: 4

Question C3_2:
A Python program is given below:
mylist = [1, 2, 3]
Which of the following is the output of mylist.count(2)?
Option 1: True
Option 2: -2
Option 3: 0
Option 4: 1

Question C3_3:
A Python program is given below:
temp = [1,2,3]
temp.insert(2,4)
temp.append(1)
print(len(temp))
What is the result after executing this program?

Question C3_4:
A Python program is given below:
temp = [1,2,5]
temp.reverse()
print(temp[0])
What is the result after executing this program?

Concept: C4

Question C4_1:
A Python program is given below:
mail = ‘anna@gmail.com’
if(‘gmail’ in mail):
print(True)
else:
print(False)
What is the result after executing this program?

Question C4_2:
Which of the following options will output false?
Option 1: print(1 != 1 and 1 == 1)
Option 2: print(1 != 2)
Option 3: print(1 == 1)
Option 4: print(1 != 1 or 1 == 1)

Question C4_1:
Which of the following is not a reserved word for logical judgement?
Option 1: or
Option 2: not
Option 3: and
Option 4: break

Question C4_2:
Which of the following options will output false?
Option 1: 1 == 1 and 1 != 2
Option 2: 1 == 1 or 1 == 3
Option 3: ‘a’ == ‘a’
Option 4: not(1 == 1 or 1 == 3)

Question C4_3:

Question C4_3:
In a conditional program (nested if is not considered), what are the maximum number and minimum number of if and else?
	A Python program is given below:
	score = 59
	if score > 60:
	    print('ans1')
	    if score < 60:
	        print('ans2')
	else:
	    print('ans3')
	What is the result after executing this program?
	Question C4_4:
	A Python program is given below:
	if 'a' in ['ab', 'c']:
	    print('ans1')
	else:
	    print('ans2')
	What is the result after executing this program?

Appendix B

Extracted features for each learning system

Learning system	Feature	Description
BookRoll	f_1: reading time	Total e-book reading time in class for each concept.
	f_2: marker amount	Number of markers added to the e-book per concept.
	f_3: memo amount	Number of memos added to the e-book per concept.
	f_4: review time	Total e-book reading time after class for each concept.
	f_5: preview time	Total e-book reading time before class for each concept.
Assessment system	f_6: time of first assessment	The first time each concept was tested in the assessment system.
	f_7: correct answer rate in assessment	Correct answer rate for each concept in the assessment system.
	f_8: number of completed tests	Number of completed tests in the assessment system.
Review system	f_9: correct answer rate in review	Correct answer rate for each concept in the review system.
	f_{10}: time of first review	The first time each concept was tested in the review system.

Appendix C

Predictive performance of various feature selection algorithms with different models for each concept.

Concept	Model	Performance: Accuracy/Precision/Recall/F1/AUC			
		Without feature selection	Feature selection method		
			mRMR	Chi2	Relief
C_1	SVM	.70/.80/.70/.64/.63	.90/.95/.90/.91/.94	.70/.75/.70/.72/.63	.80/.84/.80/.76/.67
	LR	.75/.81/.75/.68/.58	.90/.91/.90/.88/.67	.80/.84/.80/.76/.67	.70/.79/.70/.74/.39
	KNN	.65/.78/.65/.55/.56	.90/.91/.90/.89/.83	.80/.64/.80/.71/.50	.80/.93/.80/.84/.89
	DT	.80/.80/.80/.80/.61	.80/.91/.90/.89/.83	.80/.80/.80/.80/.69	.80/.80/.80/.80/.69
C_2	SVM	.78/.77/.78/.76/.71	.89/.94/.89/.90/.94	.78/.78/0/78/.78/.75	.67/.68/.67/.67/.68
	LR	.78/.77/.78/.76/.71	.94/.95/.94/.94/.83	.67/.79/.67/.61/.63	.67/.70/.67/.68/.67
	KNN	.67/.67/.67/.67/.63	.78/.89/.78/.80/.86	.78/.78/.78/.78/.68	.89/.92/.89/.89/.92
	DT	.78/.76/.78/.76/.66	.89/.91/.89/.89/.90	.78/.87/.78/.78/.83	.67/.79/.67/.61/.63
C_3	SVM	.78/.69/.78/.73/.47	.89/.90/.89/.87/.75	.78/.83/.78/.74/.67	.78/.67/.78/.68/.50
	LR	.78/.83/.78/.72/.60	.89/.90/.89/.87/.67	.77/.78/.77/.77/.44	.78/.93/.78/.82/.88
	KNN	.72/.77/.72/.75/.41	.89/.90/.89/.87/.75	.89/.94/.89/.90/.94	.89/.90/.89/.87/.75
	DT	.78/.78/.78/.78/.44	.89/.94/.89/.90/.94	.78/.84/.78/.76/.75	.67/.92/.67/.73/.81
C_4	SVM	.77/.77/.77/.77/.60	.89/.94/.89/.94/.58	.78/.78/.78/.78/.68	.89/.79/.89/.84/.50
	LR	.77/.75/.77/.74/.66	.89/.90/.89/.87/.75	.78/.61/.78/.68/.50	.78/.78/.78/.78/.44
	KNN	.77/.76/.77/.75/.71	.89/.79/.89/.84/.50	.67/.70/.67/.68/.67	.89/.91/.89/.88/.83
	DT	.82/.80/.82/.81/.63	.89/.91/.89/.88/.83	.67/.68/.67/.67/.68	.78/.78/.78/.78/.75

Appendix D

The following are the feedback questions posed to students after the review activity to gauge their opinions on the individualized intervention tutoring approach. Students indicated their agreement by using a 5-point Likert scale.

Q1. The teacher’s summary activities for each concept addressed in class were helpful.
Q2. The programming remedial materials provided today will be helpful for exam preparation.
Q3. The programming remedial materials provided today helped me review concepts that I am not familiar with.
Q4. The programming remedial materials have helped me to review my knowledge of programming concepts, and these materials can help me improve my test scores.

Appendix E

Regression models for learning characteristics, learning strategies, and learning performance.

Independent variable: Pre-test of learning performance						
Dependent variables	β value	Standard error	*t*	R^2	Adjusted R^2	*F*
				.32	.28	7.92***
Constant	60.62	4.34	13.98***			
f_3	.24	.38	2.99**			
f_8	1.15	.12	2.11*			
Independent variables: Learning feature f_3 (memo amount)						
dependent variables	β value	standard error	*t*	R^2	Adjusted R^2	*F*
				.25	.22	11.01**
Constant	-51.30	20.40	-2.52*			
LS_9(Help seeking)	18.30	5.52	3.32**			
Independent variables: LS_9 (Help seeking)						
dependent variables	β value	standard error	*t*	R^2	Adjusted R^2	*F*
				.13	.10	5.0*
Constant	2.57	.50	5.16***			
LS_1(Rehearsal)	.27	.12	2.24*			

Note. $^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$.

Appendix F

Descriptive statistics of SRL ability pretest and learning performance posttest for high/medium/low groups for the rehearsal, elaboration, organization, and critical thinking strategies

				Mean/*SD*		Mean/ *SD*	
		Number		SRL ability pre-test		Learning performance post-test	
	group	G_C	G_E	G_C	G_E	G_C	G_E
Rehearsal	Re_L	17	6	3.53/.31	3.33/.30	87.35/5.89	9.00/5.48
	Re_M	20	19	3.97/.18	3.96/.22	85.75/11.50	91.05/7.56
	Re_H	5	11	4.79/.25	4.88/.18	59.00/3.08	87.27/6.07
	total	42	36	3.91/.48	4.11/.58	83.21/15.65	89.72/6.86
Elaboration	Re_L	10	19	3.67/.14	3.86/.26	85.50/1.66	89.21/7.31
	Re_M	26	7	3.98/.20	4.00/.27	86.54/1.75	92.14/7.56
	Re_H	6	10	1.85/.26	4.72/.25	65.00/27.57	89.00/5.68
	total	42	36	4.02/.41	4.11/.45	83.21/15.65	89.72/6.86
Organization	Re_L	8	4	3.00/.29	3.13/.48	88.75/6.94	95.00/4.08
	Re_M	30	20	3.82/.23	3.98/.21	85.00/11.22	89.25/7.48
	Re_H	4	12	4.73/.32	4.75/.19	58.75/33.26	88.75/6.08
	total	42	36	3.76/.64	4.12/.56	83.21/15.65	89.72/6.86
Critical Thinking	Re_L	19	11	3.28/.28	3.56/.32	84.74/11.48	88.64/8.09
	Re_M	17	14	3.96/.12	3.96/.12	89.12/7.12	9.00/6.20
	Re_H	6	11	4.72/.27	4.70/.21	61.67/26.20	9.45/6.88
	total	42	36	3.75/.56	4.05/.50	83.21/15.65	89.72/6.86