Research of Computational Thinking Empowering Medical Students

Jia Rui Si, Yu Shan Huang, Liang Yi Yang, Yuan Xue Liu and Jia Wei Liu

May 12, 2023

Research of Computational Thinking Empowering Medical Students

Jiarui Si,Yushan Huang,Liangyi Yang,Yuanxue Liu,Jiawei Liu

Tianjin Medical University, Tianjin, China

sijiarui@tmu.edu.cn

Keywords: MEDICAL STUDENTS, COMPUTATIONAL THINKING, MATLAB, MOOC, BLENDED COURSE TEACHING, ONLINE LEARNING.

Abstract

In this age, Computational thinking has become the third major way of thinking alongside theoretical and experimental thinking. "Computing technology" has been widely used in all aspects of the medical field and has become a necessary supporting tool for the development of modern medicine. For medical students, the ability of computational thinking plays a crucial role in basic research, clinical diagnosis, and treatment in the future. However, medical students learning primarily depends on experimental thinking and lacks computational thinking education. This article proposes three different ways to train students' computational thinking. First, by taking computer courses, students can better understand how computer scientists think about things as well as the basic concepts of computational thinking from a theoretical and practical perspective so as to cultivate students' computational thinking. Secondly, when using the IMindMap, the software drawing mind maps, students can directly progress their thinking in the process of divergent thinking. Third, through modeling and programming, students use mind maps to abstract problems, organize and draw flowcharts of their creations, and then start modeling and completing system design with the help of Scratch software. Then, combined with real and concrete medical cases, MATLAB and Python enable students to apply computational thinking in practice. Finally, the current study proposes a hybrid curriculum based on MOOC+ Small Private Online Course (SPOC) to create a kind of learning environment suitable for computational thinking. So far, two rounds of experiments have been conducted and this approach has shown positive effects. However, there are some issues needed to be addressed in the future.

1 Introduction

In the era of big data, computing affects all kinds of areas of science and engineering [1]. Computer, as a necessary tool, has already permeated everywhere of the medical field. Medical professionals are becoming data creators, decision-makers driven by data, and data researchers[2]. Medical staff only have medical knowledge related to diagnosis and surgery, which is far from meeting the needs of this industry[3]. Therefore, it is necessary for medical students to receive training associated with Computational thinking (CT). As far as teachers are concerned, in order to meet the social requirements for talent capability structure, knowledge structure and comprehensive quality, it is essential to reform teaching methods and train medical students to have such abilities as CT and data processing..

Computing education for life sciences students is not just about teaching programming and using common software, but focuses on enriching life sciences students' curriculum with abstraction, algorithms, and logical thinking, exposing them to the computing "culture." Through collaboration with life scientists and learning from the teaching experience of the authors' team, the following recommendations emerged: (1) taking the time to explicitly reflect on computational thought processes and resisting the temptation of purely hands-on teaching, (2) focusing on discrete concepts rather than continuous concepts, (3) making basic programming a prerequisite so students can spend less time on basic problems[4].

In recent years, CT, together with theoretical thinking and experimental thinking, has become the third important way of thinking and a hot spot in computer education research. Represented by the United States, the birthplace of CT, various teaching methods have been adopted in primary and secondary schools, focusing on cultivating students' CT.

In the United States, Jacqueline Leonard used robotics and game design to improve students' CT skills [5], Irene Lee et al. attempted to combine CT with technology, engineering, and mathematics education [6], and Relkin Emily et al. developed students' CT in everyday life and made an assessment [6]. Hada conducted a study in 2021 to determine the relationship between students' CT and professional interests [8]. In addition to the United States, other developed European countries also emphasize the cultivation of CT. In 2013, the UK's "New Curriculum Plan" made CT an important part of its new information technology curriculum and the UK suggested that high-quality computer course education cultivate students' CT and creativity to understand and change the world. In 2015, Australia included CT as a key part of its new information technology curriculum in the New Curriculum Programme. In 2017, the New Zealand government implemented a new plan to make CT a must-have capability for every New Zealand student, hoping to "connect young people to the future".

In China, although the education of CT started later than the abroad, with the development of science and technology since 2010, CT, as one of the core skills, has gradually attracted attention from domestic educational think tanks and administrative departments. In 2010, the domestic C9 University Alliance emphasized in the "Joint Statement on the Development Strategy of C9 Computer Basic Teaching" that "cultivating students' CT ability is an important, long-term and complex core task of computer teaching in colleges and universities." Subsequently, China's Ministry of Education and the State Council also issued relevant policies mentioning the importance of CT. In 2022, the Ministry of Education included CT as the core competence in the field of information technology in the "Compulsory Education Information Technology Curriculum Standards (2022 Edition)"[9]. CT has been highly valued in China's higher education and scientific research. Medical students need a lot of memory and creative thinking when learning, while medical education pays more attention to the cultivation of experimental thinking and theoretical thinking, resulting in insufficient training of CT. For the basic courses of computer science in Chinese medical universities, students are just often taught the basic working principles of computers, including operating systems, office software and basic network content knowledge. However, writing and running programs is the only way we use computers, which means programming courses are typical for developing students' CT. Among computer languages such as Python, C, and Java, MATLAB and Python can establish a closer relationship with the medical profession.

On the other hand, large-scale online open courses (MOOC) have become an essential auxiliary teaching tool beyond classroom teaching. The advantage of MOOC education is that it is a platform for sharing teaching resources and joining many universities to produce high-quality courses, which allows students to take classes in different schools online and eliminates the limitations of time and space. Chinese university MOOC and Superstar platforms are mainstream online learning platforms, and our university has chosen the superstar platform and applies it to postgraduates' and undergraduates' teaching due to its capability and convenience to operate.. In addition, another online learning education model, Small Private Online Course (SPOC), has also attracted public attention, which means "small-scale restricted online course." It is generally believed that Professor Fox first used this concept. Among them, the meaning of small and private is relative to that of mass and open in MOOC. "Small" means that the number of students is generally tens to hundreds; "Private" refers to students cannot be admitted to the SPOC program until they meet the restrictive entry conditions[10]. MOOCs are more open to social personnel but difficult to manage, while SPOCs are more targeted to students on campus. The advantage of the SPOC model is that students on campus are accessible to it, and only for specific courses. On this platform, teachers can assign learning content, supervise students' learning process, and let students in the same grade fill the class, which is conducive to the development of the curriculum. This study establishes an asynchronous SPOC based on the first-class undergraduate online course "MATLAB Programming" of MOOC in Chinese University. The online teaching of SPOC+MOOC is implemented by using the national first-class course "Python Language Programming" on MOOC platform in China University, which combines with traditional online teaching methods.

At the same time, we use IMindMap software in the classroom to directly cultivate CT. IMindMap is a software developed by Tony Buzan, the founder of mind mapping. It has free lines and freehand drawings. The core method of CT is "construction"[11]. Mind mapping is to help us show the process of thinking construction. At the same time, the software is practical and can be used not only in scientific research but also in daily life. Therefore, you can use CT when using software. In addition, the flexible operation of IMindMap is easy for students because it meets different learning styles among individuals [12]. We use IMindMap in our classes, and students can make mind maps based on their learning and research plans. In the process of diverging thinking, students' CT is trained directly.

In addition, MATLAB and Python can solve medical problems better and directly than other programming languages because of their specialization. However, they are more challenging to master than the modular language Scratch. So we choose the language that is more popular among students to teach. Scratch is one of the most popular visual block programming languages in schools. Students can learn CT skills through Scratch, which can abstractly describe the problem without grammar restrictions. Compared with other programming languages, Scratch uses graphical programming, eliminating complex programming syntax and describing the problem more abstractly. Therefore, students can use Scratch to cultivate computing thinking.

2 Methodology

2.1 Medical students' perception of computational thinking and their adaptability and satisfaction with the curriculum

We conducted three questionnaire surveys on students. The following figure shows the results of the questionnaire survey. The first questionnaire survey was mainly aimed at 59 students in our clinical medicine class of 2018 who have not yet been exposed to computer courses. The main purpose of this questionnaire is to investigate the level of understanding of computational thinking among these students. It is worth mentioning that there was no statistically significant difference in their past grades, interests, age, gender, etc. (P>0.05).

The second questionnaire survey was conducted on 59 medical students from Tianjin Medical University in 2019 who had studied MOOC+SPOCS hybrid courses. The subjects of the third questionnaire survey were 80 medical students from Tianjin Medical University in 2021 who had taken drug sources.

2.2 Approaches to the training of computational thinking

We proposed three methods to train students' computational thinking. Firstly, by taking computer courses, students can understand the basic concepts of computational thinking and the different ways of thinking of computer scientists from both theoretical and practical perspectives, thereby cultivating their computational thinking. Secondly, in the process of thinking, students can directly engage in divergent thinking and logical integration through the mind map created by IMindMap software, while practicing abstraction, simplification and other scientific thinking methods. Thirdly, in modeling and programming exercises, use mind maps to abstract problems, organize the creative content, and draw a flowchart. Then, use Scratch software to start modeling and completing system design. Then, combined with specific medical cases, MATLAB enables students to comprehensively apply computational thinking in practice. Finally, we propose a hybrid course based on MOOC+SPOC to create a learning environment for pervasive computing thinking. The course design concept is as follows (Figure 1).

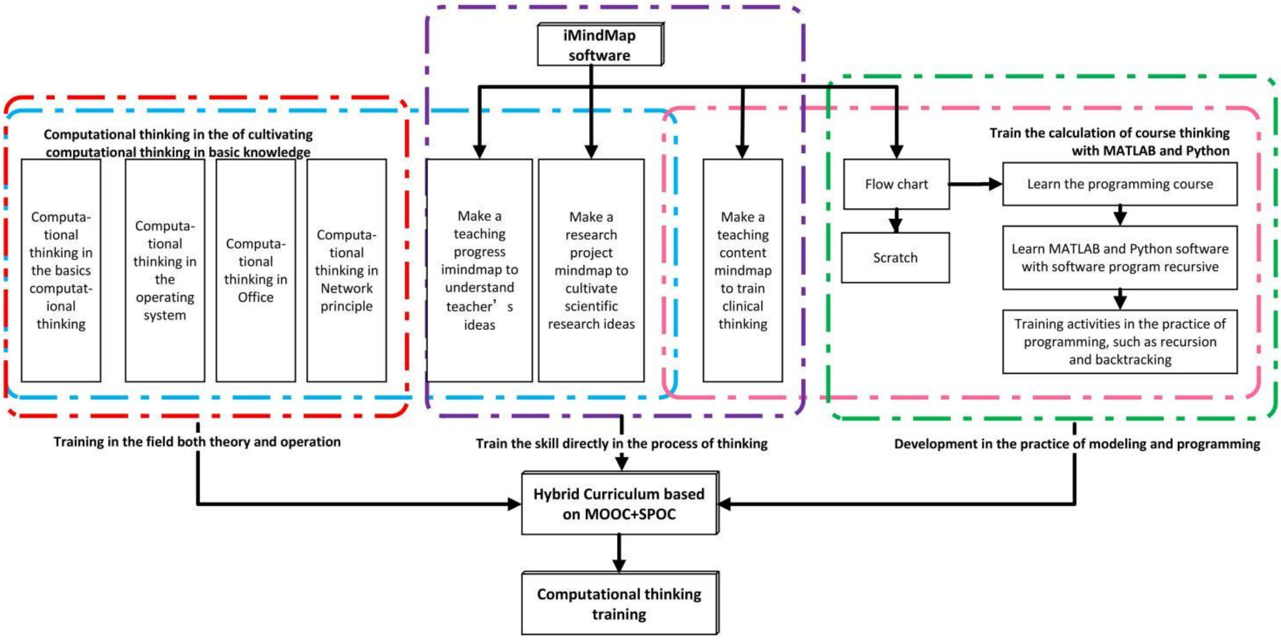

iMindMap software
Computational thinking in the of cultivating computational thinking in basic knowledge
Computa-tional thinking in the basics computat-ional thinking
Computa-tional thinking in the operating system
Computa-tional thinking in Office
Computa-tional thinking in Network principle
Make a teaching progress imindmap to understand teacher's ideas
Make a research project mindmap to cultivate scientific research ideas
Make a teaching content mindmap to train clinical thinking
Train the calculation of course thinking with MATLAB and Python
Flow chart
Scratch
Learn the programming course
Learn MATLAB and Python software with software program recursive
Training activities in the practice of programming, such as recursion and backtracking
Training in the field both theory and operation
Train the skill directly in the process of thinking
Development in the practice of modeling and programming
Hybrid Curriculum based on MOOC+SPOC
Computational thinking training

Figure 1. The course design concept

*2.2.1 The cultivation of computational thinking in basic courses:*For college students, cultivating computational thinking through theory and practice is mainly based on the basic courses of college students. Because computers themselves cannot think actively, they cannot be subjectively assumed to have ideas. Computational thinking refers to human thinking, which is based on a better connection between computers and humans. For example, when studying the history of computer development, students learn the unique ways of thinking of computational scientists. While learning the basic structure and working principles of computers, they also understand the thinking implementation of computer simulation of human brain work. In the learning of operating systems, it is possible to cultivate students' thinking methods in system design. When using computer networks, they can master a computational mindset that simplifies complexity； when conducting information retrieval, they are learning the computational thinking of induction and summarization; when using input devices and Word to input text, they train their numerical encoding thinking ; When learning and mastering office software such as Excel, they can apply their computational thinking of using models to solve problems in the software. In addition, students can also conduct practical operations and computer assembly based on their interests and theoretical knowledge they have learned.

*2.2.2 Train computational thinking through mind mapping:*IMindMap is a software developed by Tony Buzan, the founder of Mind Map. It has free lines and hand drawn functions. The core method of computational thinking is "construction" [13], and mind maps are designed to help us demonstrate the process of thinking construction. At the same time, the software has practicality, which can be used not only in scientific research, but also in daily life. Therefore, computational thinking can be applied when using software.

The author who creates the mind map draws a fundamental point from the initial idea. As thinking and ideas expand, IMindMap can connect and develop like neurons. Each branch represents the emergence and development of an idea. By continuously divergent thinking, a mind map was ultimately completed around the initial basic points. The map making process that simulates divergent thinking in the brain can not only help students focus, but also cultivate their creativity. In addition, the flexible operation of IMindMap is easy for students to operate, as it satisfies different learning methods among different individuals [14]. We used IMindMap in class, where students can create mind maps based on their own content and research plans. During the unfolding process, students' computational thinking was directly trained.

*2.2.3 Training of computational thinking in modeling and programming practices:*Before explaining the specific programming language, instructors let students use IMindMap software to abstract the problem, sort out the creative content, and then draw a flowchart based on this. Then, students used Scratch software to graphically model the problem, and finally wrote a program using MATLAB and Python languages to solve the problem. This step-by-step learning effectively decomposed the process of the program, reduced the difficulty of learning, and cultivated computational thinking throughout the process.

a) Using IMindMap to create a flowchart: For example, when designing a program to calculate BMI, we can first create a mind map as shown in Figure 3, with BMI as the theme and four standards as branches.

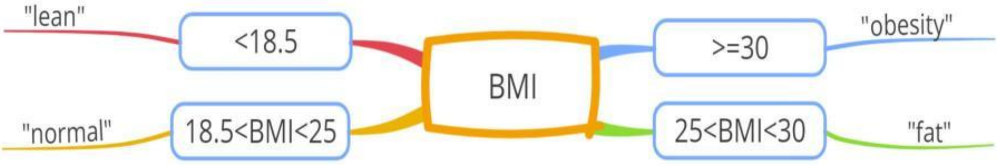
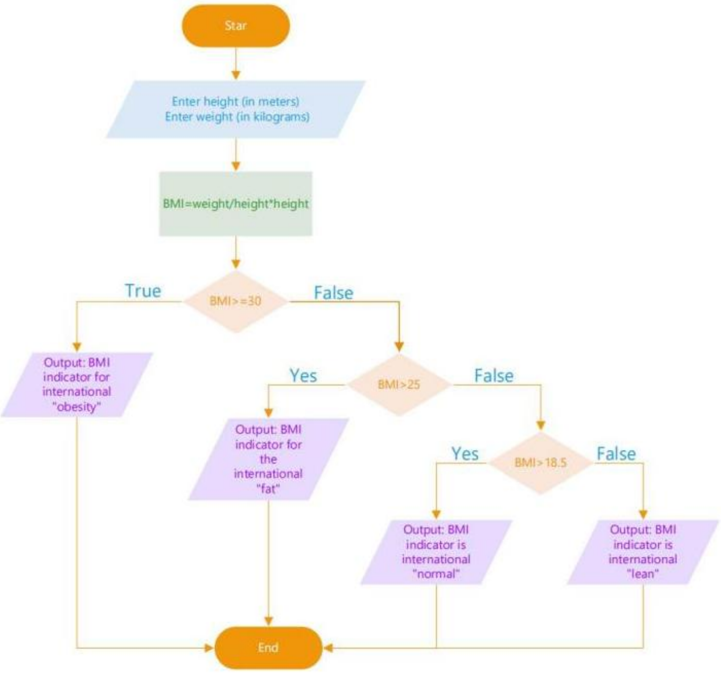


Figure 2. Using IMindMap to create a flowchart

The visualization of programming has improved computer learning [15], thereby enhancing their computational thinking abilities. Simplification, logicality, and visualization of complex problems in mind mapping. By analyzing and representing problems in mind maps, it is easy to create a program flowchart and refine the steps to solve problems. In the above operation case, by organizing the mind map (Figure 2), students can condense the specific steps for calculating BMI values through the flowchart (Figure 3), laying the foundation for further programming.



Star
Enter height (in meters)
Enter weight (in kilograms)
BMI=weight/height*height
True
BMI>=30
False
Output: BMI indicator for international "obesity"
Yes
BMI>25
False
Output: BMI indicator for the international "fat"
Yes
BMI>18.5
False
Output: BMI indicator is international "normal"
Output: BMI indicator is international "lean"
End

Figure 3. Calculating BMI values through the flowchart

b) Modeling with Scratch: Scratch is a graphical and modular basic programming tool developed by the Massachusetts Institute of Technology Media Laboratory [16] in 2007, which can be used to achieve computational thinking training without the need for specific language programming. The use of Scratch can meet all dimensions in the computational thinking framework provided by Professor Brennan and Professor Resnick [17]. In addition, the programming language tends to be professional and domain oriented, which enables the same task to be completed faster than the general computer language. However, because of its more professional nature, it is more difficult to get started. Therefore, we chose the language that was more popular among students for teaching. Scratch is one of the most popular visual block programming languages in schools. K-9 students can learn computational thinking skills through Scratch and consider the progress of their learning. There are other CT skills that have not been captured by Brennan and Resnick's frameworks in the same fashion, including input/output, reading, interpreting, and communicating code, using multimodal media, predictive thinking, and human-computer interaction, which can be considered as possible supplements to their frameworks[18]

On the basis of drawing a flowchart, Scratch can be used to construct module instructions, such as blocks, and model problems in a graphical manner. In addition, this training method helps promote the training of computational thinking by reducing difficulty levels and making it interesting [19.20]. At the same time, grassroots exercises before formal projects can help students better understand [21].

For example, when writing programs to calculate BMI, students can use Scratch to complete language independent modeling based on mind maps and flowcharts (Figure 4).

c) Programming using MATLAB and Python languages: The only way humans can use computers is to write and run programs [22]. Therefore, programming courses are actually typical thinking training courses. Finally, we used MATLAB and Python. Although Python is the most popular language in the era of big data, we believe that MATLAB should be the preferred language for medical students to learn how to program. The reasons are as follows. 1) MATLAB: ① Computational thinking is the complementarity and integration of mathematical thinking and engineering thinking. MATLAB is one of the three major mathematical softwares in the world, and it is widely used in biomedical engineering, which can best reflect the characteristics of computational thinking; ② MATLAB is not only a programming language, but also has a graphical user interface, making it easier to learn and master; ③ MATLAB comes with many toolkits that can be used in the medical field, making it convenient to process various medical data; ④ Neural networks and deep learning toolkits make it easy to implement artificial intelligence algorithms in MATLAB, enabling simulation of human behavior. For example, it is an appropriate tool for virologist to design, stimulate and analyze viral metagenomes [23].2) Python: ① Python is the world's largest single language programming community, with a wide range of third-party programming libraries and a powerful computing ecosystem, which can encapsulate the excellent achievements of other programming languages, reduce the complexity of usage, and is suitable for medical students who have not studied programming professionally; ② Python can easily build TensorFlow or PyTorch, build AI frameworks, and achieve mining in medical big data.

Instructors have offered the course 'Medical Data Processing and MATLAB Implementation'. Firstly, students were given instructions on the basic operations of MATLAB software and the basic methods of programming in MATLAB language, and then targeted modular teaching was provided tailoring to the special needs of medical students. The specific requirements are listed below. For students majoring in medical imaging, they need to process medical images such as CT or FMRI. In order to help them better understand the meaning of medical images, they further studied the image processing toolbox. For students majoring in public health, they need to conduct statistical analysis on large-scale data, but they lacked the ability to use SAS or SPSS. Therefore, they would study statistical toolkits. For clinical medical students, in order to lay the foundation for future use of hospital management systems, they would master the graphical user interface design of MATLAB and the simple design of management information system cases. For students majoring in biomedical engineering who need to analyze EEG and ECG data, it was recommended to use the medical signal processing toolkit. For students in bioinformatics, they need to analyze gene protein sequences and gene chip data. So they also learned medical signal processing toolkits. For students majoring in health service management, they are the main force in medical big data analysis and processing. So they learned neural network toolkits. For nursing students, they would learn the interaction between MATLAB and Word, which would make it easier to record as electronic medical records in the future [24]. At the same time, students could also choose different modules for learning based on their personal interests.

Simultaneously, a course called "Python Processing of Medical Data" was also offered. In this course, the teacher first introduced the concept of computational thinking and IPO writing methods for programming. Through BMI medical cases, the teacher introduced the drawing method of flow charts and the three basic structures of programs (sequence, branch, and loop). Then, based on the flowchart of the selected structure, the teacher introduced the basic construction method of scratch, enabling students to break free from the grammar environment of programming languages and master the basic ideas of program design. After the student successfully ran the scratch code, the teacher would explain the code implementation of the Python language to familiarize the student with Python syntax. In the following courses, we introduced some classic cases, such as the conversion of Fahrenheit temperature to Celsius temperature, the basic usage of sea turtles, the daily improvement of strength, the calculation of basic statistical values, and text word frequency analysis. The teacher explained in detail the Python grammar points involved in each case, allowing students to master as many grammar points of the Python language as possible in the shortest possible time, as well as the installation and use of third-party libraries. Finally, students were asked to collect medical application cases of Python to broaden their horizons.

3 Results

According to the survey results , in the first questionnaire survey, 73% of students only heard of computational thinking (a). After seeing the clear definition of "computational thinking" in the questionnaire, 54% of students believed that computational thinking is meaningful to them to some extent (b), and 75% of students expressed a desire to integrate computational thinking with their majors (c). We found that most students have insufficient understanding of computational thinking and are not aware of its application in professional learning. In the second questionnaire survey, 79.25% of students were able to adapt to the progress of online teaching (d), and over 75% of students were satisfied with the effectiveness of online teaching (e). In the third questionnaire survey, over 90% of students were willing to learn Scratch and believed that MATLAB and Python were beneficial in medical data processing. More than 50% of people like both software (f, g, h, i) at the same time.

In order to verify the effectiveness of the training, we used the medical application course "Medical Data Processing and MATLAB Implementation" after the basic course of "Computer Fundamentals for College Students" as the basis. We would compare different grades of students who adopted these training methods, including 31 and 33 students from two classes of clinical majors in 2018, with 249 students from all grades of the school (including "5+3" clinical, anesthesiology, management, and other majors).

According to the computational thinking of Professor Brennan and Professor Resnick, a three-dimensional measurement framework has been established that includes three dimensions: concept, practice, and perspective. This framework corresponds to learners' mastery of concepts such as "loops", "operators", and "sequences" in their applications, as well as their practical use of concepts such as abstraction, modularity, and testing. It also corresponds to the perspective and questioning brought by learners when combining their own viewpoints with expression and connection. The above three parts are specifically implemented in the course assessment, corresponding to the three modules in the exam, namely multiple-choice questions, short answer questions, and true/false questions. The specific training effect analysis is as follows.

Table 1. The results of specific training effect analysis

Object	Selection (CT concepts)	Short answer (CT practices)	Judge (CT perspectives)
Class 1	42.65	7.22	36.12
Class 2	40.42	8.06	35.39
The average scores	38.52	7.56	33.55

Firstly, the indefinite choice question corresponded to the "concept" part of computational thinking. Single and multiple-choice questions were used, such as "What is the essence of computational thinking", "What are the symbols of conjugate transposition operators", and "What is the main focus of the definition of computational thinking" to test students' mastery of relevant concepts. In the multiple-choice questions with a total score of 50, the average score for Class 1 was 42.65, the average score for Class 2 was 40.42, and the average score for the entire grade was 38.52 (Table 1). For the mastery of concepts related to computational thinking and programming, the training of computational thinking in both classes was significantly higher than the average level.

Secondly, the short answer questions corresponded to the "practical" part of computational thinking. The given program was used to test students' practical programming skills. In this case, students were required to write down the running results of the program and fill in the code and other tables in the blank space. In the short answer questions with a total score of 10, Class 1 averaged 7.22 points, Class 2 averaged 8.06 points, and the grade averaged 7.56 points. (Table 1) For the practice of computational thinking, the grades of the second class trained in computational thinking were significantly higher than the average level, while the grades of the first class were slightly lower than the average level.

Thirdly, determining the "perspective" part of the problem corresponded to computational thinking.A given expression program was used to test their expressive and self-questioning abilities, requiring them to judge the truth of a given proposition. For example, compared to mathematics and physics, abstraction in computational thinking is simpler Command B=0,10100 ' will generate a matrix of 1 * 10. Among the questions with a total score of

40 points, the average score of Class 1 was 36.12 points, the average score of Class 2 was 35.39 points, and the average score was 33.55 points. (Table 1) In terms of using computational thinking for expression and questioning, the performance of the two trained classes was significantly better than the average level.

Finally, based on the comprehensive analysis results, the average scores of Class 1 and Class 2 were 86 and 83.8 (out of 100), respectively, which were higher than the 81.8 scores of the entire grade. In the 100-point test, the average score of Class 1 was 86, while the average score of Class 2 was 83.8, which was higher than the average score of 81.8 for the entire grade. The training class achieved above average results in three aspects of computational thinking. Although Class 1 was slightly lower in the "practical" part of computational thinking than grade, it is much higher in the "conceptual" and "perspective" parts. Most importantly, the overall level was still above the average. In summary, students who had followed the above training computational thinking method performed better and possessed more computer skills than those who had not followed the method. Teaching practice has shown that this training method has a good effect on improving students' computational thinking.

This study also found a relationship between the time medical students spent watching videos and their classroom exams and final grades. After analyzing the data results, we found that there was no linear correlation between scores and the time spent watching the video. Most students watched course videos for less than 10 hours, while some students watched for much longer than 8 hours. However, no matter how long they spent on the video, only a few people achieved good results. Therefore, it cannot be inferred that the length of video viewing time is directly related to the score. In addition, when studying the relationship between the number of times medical students watch videos and the length of time they watch videos during online learning, we found that very few students repeatedly watched videos during the video learning process. During the investigation process, most medical students did indeed complete a large number of video viewing tasks, but the overall viewing time was not long. This indicates that in the process of video learning, students tended to choose short videos to quickly complete learning tasks.

4 Conclusion

Computational thinking is a universal attitude and skill. Besides scientists, ordinary people can also be enthusiastic about learning and using it. Computational thinking will help people engage in many fields, including medicine and other types of science, such as engineering and even art.

This article aims to address the problems in the cultivation of computational thinking among medical college students. Based on supplementing and improving the concepts and methods of computational thinking, it reflects on computer course teaching from three aspects and proposes suggestions and ideas for reforming the classroom. The training of computational thinking requires abstract and universal methodological generalizations, as it is an abstract and concrete process. In addition, it also needs to target individual projects for students. However, the training process of computational thinking is long-term.

In addition, through the analysis of learners' online learning behavior, this study has found some problems with this teaching standard. Teachers can only partially understand students from the data, which is a major difference between the new online teaching and traditional LBL teaching, leading to teachers' negative treatment of students. The lack of self-directed learning ability can affect the learning efficiency of participants. Analysis of students' online learning behavior found that they rarely watch videos repeatedly during autonomous learning, and they intentionally choose to watch shorter videos. Despite spending a lot of time watching course videos, some students still achieved very poor grades. In fact, this is because they have not really studied this learning materials. Students lack autonomy in learning, leading to the phenomenon of muddling along, making video learning impractical.

In order to strengthen the management of students' online learning, teachers should pay attention to repeated checks of homework [25]. Improving video quality and the attractiveness of MOOC+SPOC platforms, as well as some technical measures, will help ensure comprehensive monitoring of video viewing. Teachers can provide technical guidance after class, which is beneficial for students to complete learning tasks and enrich their knowledge.

After two years of exploration, a teaching model has been established. It can enhance the computational thinking of medical students. However, there are still some issues with this model. The author will continue to track the quality of students cultivated through this training method.

It is said that education is a bronze medal, ability is a silver medal, and connections are a gold medal. So, what is an ace? The answer is to think. Because computers are everywhere today, students who learn computational thinking and apply it to the medical industry will be invincible!

5 Acknowledgements

The authors acknowledge support from teaching reform research of Tianjin Medical University.

The authors declare no competing or competing interests. The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

6 References

[1] Wing, J. Computational thinking. Commun. ACM 49, 3 (Mar. 2006), 33–35.

[2] Saqr Mohammed,Tedre Matti. Should we teach computational thinking and big data principles to medical students? [J]. International journal of health sciences,2019,13(4).

[3] Yao B. Research and exploration of computer basic teaching in medical vocational colleges [J]. China Information Technology Edusation, 2015(7): 131, 143.

[4] Hu SS,Li P.Discussion on Wisdom Education of Medical Colleges in the Big Data Era [J]. JOURNAL OF MEDICAL INFORMATICS,2017,38(05):85-88.

[5] Jacqueline Leonard, Alan Buss, Ruben Gamboa, Monica Mitchell, Olatokunbo S. Fashola, Tarcia Hubert & Sultan Almughyirah. Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills [J]. JOURNAL OF SCIENCE EDUCATION AND TECHNOLOGY 860–876(2016). Doi: 10.1007/s10956-016-9628-2.

[6] Irene Lee, Shuchi Grover, Fred Martin, Sarita Pillai & Joyce Malyn-Smith. Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education [J]. JOURNAL OF SCIENCE EDUCATION AND TECH-NOLOGY, 1–8(2020). Doi: 10.1007/s10956-019-09803-w.

[7] Relkin Emily, de Ruiter, Laura, Bers, Marina Umaschi. TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education [J]. JOURNAL OF SCIENCE EDUCATION AND TECHNOLOGY, 482-498 (2020). Doi: 10.1007/s10956-020-09831-x.

[8] Hava Kevser, Unlu Zeynep Koyunlu. Investigation of the Relationship Between Middle School Students' Computational Thinking Skills and their STEM Career Interest and Attitudes Toward Inquiry[J]. JOURNAL OF SCIENCE EDUCATION AND TECHNOLOGY, 2021. Doi: 10.1007/S10956-020-09892-Y.

[9] Chen XZ,Zhang HL,Yang Y,Comparison of the research context and practice of computational thinking education at home and abroa[J].Journal of Comparative Education,2023,343(01):148-161

[10]Wang XY.An Empirical Study of SPOC Mixed Teaching Model in English Reading Teaching in Senior High School[D]. Harbin Normal University,2022.DOI:10.27064/d.cnki.ghasu.2022.001383.

[11] Lu HQ, He QM, Xu JC. Teaching content design of "university computer foundation" course based on computational thinking [J]. Teaching in Chinese universities, 2012(09):55-58.

[12] Wan Noor Hazlina Wan Jusoh, Suraya Ahmad. IMindMap as an innovative tool in teaching and learning accounting: an exploratory study. Interactive Technology and Smart Education.2016, 13(1):71-82. Doi: 10.1108/ITSE-05-2015-0012.

[13] Lu HQ, He QM, Xu JC. Teaching content design of "university computer foundation" course based on computational thinking [J]. Teaching in Chinese universities, 2012(09):55-58.

[14] Wan Noor Hazlina Wan Jusoh, Suraya Ahmad. IMindMap as an innovative tool in teaching and learning accounting: an exploratory study. Interactive Technology and Smart Education.2016, 13(1):71-82. Doi: 10.1108/ITSE-05-2015-0012.

[15] Eben B. Witherspoon, Ross M. Higashi, Christian D. Schunn, et al. Developing Com-putational Thinking through a Virtual Robotics Programming Curriculum. ACM Trans-actions on Computing Education (TOCE).2017, 18(1):1-20.

[16] LeChen Zhang,Jalal Nouri. A systematic review of learning computational thinking through Scratch in K-9[J]. Computers & Education,2019,141. Doi: 10.1016/j.compedu.2019.103607.

[17] Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American educational research association (pp. 1–25). (Vancouver, Canada).

[18] LeChen Zhang,Jalal Nouri.A systematic review of learning computational thinking through Scratch in K-9[J]. Computers & Education, 2019(141):103607-103607.

[19] Wang Danli, Wang Tingting, Liu Zhen. A tangible programming tool for children to cultivate computational thinking. [J]. TheScientificWorldJournal,2014,2014. Doi: 10.1155/2014/428080.

[20] Weinan Zhao, Valerie J. Shute. Can playing a video game foster computational think-ing skills? [J]. Computers &amp; Education,2019,141. Doi: 10.1016/j.compedu.2019.103633.

[21] Amir Rubinstein, Benny Chor. Computational Thinking in Life Science Education. PLOS Computational Biology.2014, 10(11).

[22] Chen GL, Dong RS. Computational thinking and basic computer education in universi-ties [J]. Teaching in Chinese universities, 2011, (1):7-11,32.

[23] Martin Norling, Oskar E. Karlsson-Lindsjö, Hadrien Gourlé, et al. MetLab: An In Silico Experimental Design, Simulation and Analysis Tool for Viral Metagenomics Studies. Plos One.2016, 11(8). Doi: 10.1371/journal.pone.0160334.

[24] Xu DY, Ma J. Application of Computational Thinking in the Teaching Reform of Database Basis in Medical Colleges [J]. The electronic world, 2013(12):152.

[25]Xiang,Q, Yu R.; Ding Y. etc. Exploration on the New Mode of Traditional Experiment Teaching and Turning Classroom in MOOC Background. Science and Technology Innovation Herald. 2017,14(01):214-217.