

Assessing novice programmers' performance in programming exams via computer-based test.

Master Thesis
Educational Science and Technology
University of Twente

Author:
Deniz Gursoy

First Advisor:
Lars Bollen, Dr.

Second Advisor:
Hannie Gijlers, Dr.

Enschede, June 2016

I dedicate this to my parents

Abstract

Problem: Students who are registered for the computer science programs are having some difficulty in answering the questions in programming exams that they are supposed to write some code on paper. Students are accustomed to write codes into code editors. In paper-based tests, students cannot take advantages of code editors such as syntax highlighting, automatic indentation and code autocompleting. Students spent extra time to check codes and add the punctuations marks in the correct places. Students also face the difficulty of adding new codes between the lines already written. They also have difficulty in tracing the code since they do not take advantage of syntax highlighting. When students take CBT in a programming course, they are supposed to finish the test earlier because they take advantage of a code editor. Since they have extra time during the exam, students check their answer; thus, CBT might create significant exam score difference between students who take CBT and those who take PBT. To overcome problems that students face in paper-based programming exam, students are introduced with a computer-based test that allows students to write their answers to a code editor.

Method: The study is a mixed method study consisting of quantitative and qualitative parts. The qualitative part is used to clarify the result in the quantitative part. The quantitative part of this study consists of two exams taken by 44 students (28 male, 16 female) at the Middle East Technical University in an experimental setting. The quantitative part also includes data coming from a TAM questionnaire with variables Perceived Ease of Use, Perceived usefulness and Attitude towards use constructs. Quantitative part tries to find: Which group performs better in the tests and completes it earlier, and what are the correlations between TAM variables? It is also checked if exam score and time affect perceived ease of use of the computer-based test. The qualitative part of the study consists of semi-structured interviews conducted with four participants (three males and one female) at the University of Twente. The qualitative part aimed to find the factors affecting students' attitude toward the computer-based test. Attitude is measured to show in the literature whether students accept computer-based test or not.

Data analyses: Independent samples t-tests were used to compare the exam score and time spent in the exam between control and experimental group. Reliability of the exams was checked by improved Split-half reliability. Reliability of TAM questionnaire was checked by Cronbach's alpha value. Correlations between TAM variables were analyzed by Spearman's Rho method. Interviews were recorded and then transcribed. Transcribed interviews were analyzed by finding the descriptive codes and finding themes from those codes.

Results: Based on the result of the quantitative analysis, there is no statistically significant exam score difference between students who took CBT and PBT. Students who took CBT completed the exam in significantly less time. Analyses of correlation among TAM variables show that perceived ease of use positively and significantly correlates to perceived usefulness and attitude towards; however, perceived usefulness does not significantly correlate to attitude towards the use of CBT. Exam score and time spent in the exam do not correlate to the perceived ease of use of the CBT. Results of the qualitative analysis show that code editor, the design of the software and pictures in questions are the factors that affect students' attitude towards CBT.

Keywords: code-editor, assessment, programming, TAM

Table of Contents

1. Introduction.....	5
2. Conceptual Framework.....	6
2.1. Implementaions of CBT and its generations.....	6
2.2. Computer-based tests and Assessment in Computer Science courses	6
2.3. Technology Acceptance Model	7
2.4. Research question and hypotheses	8
3. Method	9
3.1. Design.....	9
3.2. Participants	9
3.3. Instrumentation	10
3.4. Development of the computer-based test.....	12
3.4.1. Theoretical guidelines to development of computer-based test.....	12
3.4.2. Technical information about the computer-based test.....	12
3.4.3. Security precautions	13
3.5. Procedure	13
3.6. Data analysis	14
4. Results	15
5. Discussion.....	18
6. Conclusion	20
7. References.....	22
Appendix A: Exam Questions	25
Appendix B: TAM Questionnaire.....	36
Appendix C: Overview of data analysis	38
Appendix D: Coding Scheme.....	39
Appendix E: Screenshots of the software.....	40

1. Introduction

Programming courses are important courses of academic programs related to engineering, computer science, and statistics. In programming courses, students use Integrated Development Environments (IDEs), “which incorporate a wide tool-chain of editors, compilers, runtime environments, debugging and documentation tools” (Fylaktopoulos et al., 2015, p. 43), to do their programming assignments. They are accustomed to use IDEs while writing their codes. In the entire computer programming courses, students are not supposed to write any code on paper, but they have to write codes on the paper in the exams. Students are not accustomed to write code pieces on paper. Instead of a paper-based test (PBT), a computer-based test (CBT) should be used to tackle this problem.

Students who are registered for the computer science programs do many programming assignments as requirements of the courses by using different code editors and compilers. They type their code in code editors, and they become accustomed to see and type the codes in code editors. However, this is not the case in the paper-based programming exams. Students face many problems when they take paper-based programming exams. Therefore, in a paper-based programming exam, “students usually spent some time on checking answers” (Čisar et al., 2016, p. 151).

Students use “,” and “;” punctuation marks at correct places when they type codes in a code editor, so they are accustomed to write those punctuation marks automatically in a code editor. Students miss those punctuation marks when they write codes on paper. Students also have difficulty in adding “{” and “}.” If the exam is paper-based, they forget to add “}” or spend more time to check whether “{” and “}” are at correct places. Another challenge in the PBT is that students cannot add new codes between existing lines. If they want to add a new line between two lines, they have to erase the codes that they write before and rewrite those again after they make the new line. Students are also deprived of advantages of using syntax highlighter. A syntax highlighter gives visual feedback by changing the colors of specific code pieces when students; for example, define a variable, or import a class or library. Syntax highlighting makes tracing the written code easier, and it also informs students that they used the correct code. In a PBT, students either spend extra time to check whether the codes are correct or might not realize that they make a mistake while writing code. Furthermore, students are accustomed to use auto-completion feature of IDEs, which allows them to write the codes faster, via specific shortcuts assigned to shortcuts. In a PBT, students also do not have the possibility of using the auto-completion feature of IDEs. Additionally, IDEs make automatic indentations when students write certain pieces of codes. That makes students type the codes faster. Students also feel the lack of auto-indentation in PBTs.

A possible solution for the problems above mentioned is to prepare a CBT, which has a code editor allowing students to provide answers easily to programming questions. The code editor in the CBT should have the same properties of the code editor that student are accustomed to use in their course. Students at the University of Twente and the Middle East Technical University use Eclipse software as IDE. The CBT used in the study has a code editor resembling the Eclipse.

In general, students who are enrolled in programming courses have disadvantages in paper-based programming exams. Students should take CBTs instead of PBTs. In the study, participants from the University of Twente and the Middle East Technical University took CBTs and results are reported.

This research contributes to the literature in many ways. Firstly, Čisar et al. (2016) suggest that software delivering the tests should provide some form of feedback to students. The code editor, which software delivering the test uses, gives some visual feedback by changing the colors of the code or highlighting some parts of the codes. Secondly, students have not written their answers for open-ended questions to a code editor in computer-base programming exams before. One of the main objectives of the study is to find out what students' attitudes towards CBT with a code editor are. Thirdly, Özalp and Çağıltay (2010) say that, especially in engineering education, more data concerning the equivalency of PBT and CBT are necessary. This study contributes to the literature by adding a new result of the comparison of CBT and PBT in the engineering field. If students' scores on the CBT were higher than those of students who took the PBT, this study would cast

doubts on the delivery of programming exams and prevent instructors at universities from using PBTs for programming courses.

2. Conceptual Framework

2.1. Implementations of CBT and its generations

“In the early 1970s, the US military and clinical psychologists pioneered the development of computer-based tests. Initially, psychologists saw computerized assessments as a method of controlling test variability, eliminating examiner bias, and increasing efficiency” (Russell et al., 2003, p. 280). As the technology improves and testing methodology changes, computer-based testing also evolves. Bunderson et al. (1989) separated CBTs into following four generations:

- Generation 1, Computerized testing (CT): administering conventional tests by computer
- Generation 2, Computerized adaptive testing (CAT): tailoring the difficulty or contents of the next piece presented or an aspect of the timing of the next item on the basis of examinees' responses
- Generation 3, Continuous measurement (CM): using calibrated measures embedded in a curriculum to continuously and unobtrusively estimate dynamic changes in the student's achievement trajectory and profile as a learner
- Generation 4, intelligent measurement (IM): producing intelligent scoring, interpretation of individual profiles, and advice to learners and teachers, by means of knowledge bases and inferencing procedures. (p.401)

Nowadays, computer-based testing is one of the most common forms of testing since the 1990s (Educational Testing Service, 1992). According to the Fair Test: National Center for Fair and Open Testing (2007), “over 100 million standardized tests are administered annually in schools. The states will have spent \$330 million on standardized achievement tests in 2000, and many individual students will pay for other examinations, such as the Scholastic Aptitude Test (SAT) and the Graduate Record Examination (GRE)”(p.14).

2.2. Computer-based tests and Assessment in Computer Science courses

Computer-based tests are being developed and used in many countries today because CBTs have some benefits such as “rapid access to test results and feedback, ability to re-score or adjust answers on exams when needed, availability of longitudinal data for long-term performance assessment, and reduction of cheating potential”(Pawasauskas et al., 2014). Apart from these main advantages, CBTs also have other following advantages:

- a) Shorter testing time for each student,
- b) a better student-test fit with adaptive tests,
- c) much higher precision in the measurement, especially at high and low achievement levels,
- d) a more enjoyable and better testing experience for the students,
- e) less stress and pressure on all concerned, as the tests will not all occur at the same time but will be administered over a period of time each year,
- j) Testing using a medium (computers) that is becoming increasingly dominant in education,
- g) The reuse of test item,
- h) Cheaper and quicker coding of test responses, and
- i) Better information about the student group, the schools, school districts and the whole educational system. (Björnsson 2008, p. 11)

Despite the various advantages, CBTs pose some disadvantages to students such as failure in the system delivering the test (Aybek et al., 2014), requirement of computer interaction and skills (Aybek et al. 2014), students' absence of basic test strategies (Natal, 1998), and extra cost to examinees (Bugbee & Bernt, 1990). CBTs also pose disadvantages of a likelihood of failures in security, hardware and software to educators (Natal, 1998).

There are two different kinds of CBTs available today. These are linear and adaptive CBTs. McFadden et al. (2001) say

A linear computerized test is a series of items presented one at a time, in the same order, to all students. Adaptive tests are individualized and account for differing abilities by changing the order and total number of items presented. The report concluded that numerous studies comparing CATs with traditional paper- and-pencil tests in the same subjects show that the two methods rank-order students in about the same way in less time. The number of test items

needed for a given level of precision can be reduced by more than half. Other advantages of CATs include year-round testing, longer time per question, and immediate feedback or notice of standing. Virtues of CAT can be extolled, because it can be tailored to district or state educational goals, target each student's proficiency level, a 'bank' of thousands of questions in each subject matter area can be used, which improves test security, new students can be tested quickly, and all students can be examined repeatedly throughout the school year. (p. 44)

In this study, a linear CBT was used because adaptive CBT requires large question pools about topics related to programming courses, which does not exist in public, and comparison of students' exam scores can be easily done with the data coming from the linear test because students solve the same questions. Therefore, scores in the linear test are more reliable.

The use of CBTs in exams also brought some issues to the discussion. Most important of these issues is the equivalence of CBTs and PBTs. The equivalence of CBT and PBT is an ongoing debate in the literature. The equivalence is tested in many different domains, but there is no consensus on the equivalency of CBT and PBT in the literature. For example, Anakwe (2008) who examined the issue in an accounting class and Karay et al. (2015) who studied this issue in medical school course found no significant difference between scores of CBT and PBT; however, Lee and Weerakoon (2001) and Russell (1999) found that, students who took PBT perform better than those who took CBT. Conversely, Pomplun et al. (2002) found that students perform better on CBT than PBT. Computer-based tests also used in engineering courses. For example, Özalp and Çağiltay (2010) tested the equivalency of PBT and CBT in a thermochemistry course, and they concluded that students' performances on the CBT and PBT do not differ significantly. Looking at the diverse and partly contradicting results in the literature, it is not clear whether the test domains (e.g. engineering, accounting, medicine) affect the results or not. Therefore more research is needed in this area. Table-1 summarizes other research regarding the equivalency of CBT and PBT.

Table-1.

Other articles investigated the equivalency of computer-based test and paper-based test.

Researchers	Domain	Results of comparison
Akdemir & Oguz (2008)	Educational Science	Equivalent
Jeong (2014)	Language	Higher in CBT
Jeong (2014)	Science	Equivalent
Cerillo and Davis (2004)	Science	Higher in PBT
Bayazit & Aşkar (2012)	Educational Science	Equivalent
Frein (2011)	Social Science	Equivalent

Assessment in Computer Science, which is an engineering related field, is done in many ways. Grading in programming courses is being done based on students' score on midterms, finals and programming assignments. Instructors of programming courses prepare assignments. Assignments in programming courses are graded by automated assessment systems, which Cheang et al. (2003) define as systems "that students can electronically submit their programming assignments and receive instant feedback"(p.121). However, these automated are not used in exams, which contain different kinds of questions such as multiple choice and open-ended questions.

Recently, Čisar et al. (2016) developed an adaptive CBT, which delivers only multiple-choices programming questions, to assess students' knowledge in a programming course. They presented programming test to students in two different conditions. Some students took programming exam having 20 questions on computers, and others took the traditional paper-based exam. According to results of the study, students who took programming test on the computer performed significantly higher than students who took the paper-based exam. Also, students who took CBT completed the test in significantly less time.

2.3. Technology Acceptance Model

With the use of new technologies and computer-based tests, it's important to determine people's attitudes towards new technologies to see if the technology is useful and accepted by the user, and to develop the new technologies in the more appealing way. Davis' (1989) Technology Acceptance Model (TAM) is used to determine people's attitude towards the new technology. The goal of TAM

is to predict whether new technology will be acceptable to the users by constructs “Attitude toward using” (A), “Perceived usefulness” (PU) and “Perceived ease of use”(PEU). PU is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" (Davis, 1989, p. 82). PEU, in contrast, refers to "the degree to which a person believes that using a particular system would be free of effort" (Davis, 1989, p. 82). Davis, Bagozzi and Warshaw (1989) postulate that PU and PEU influence A. Davis, Bagozzi and Warshaw (1989) say, “According to TAM, A is jointly determined by PU and PEU”. If PEU and PU of a system increase, the probability of using new technology increases. Furthermore, Davis et al. (1989) and Davis (1993) also postulated that PEU has a direct impact upon PU, but not vice versa. Furthermore, TAM also assumes that “some external variables affect perceived usefulness and perceived ease of use, which also mediate the effect of external variables on attitude to use” (Khader et al., 2014, p.6). Čisar et al. (2016), who measured computer science students’ programming knowledge via an adaptive CBT, reported that students who took programming tests on computer performed better and spent less time in the exam in comparison with students who took the test in the traditional way. Based on their result, external variables “Exam score” (S) and “Time spent” (T) are introduced in this research. Since 66% of those students who took computer-based programming test said that “taking the test on a computer was a much more pleasant experience than a conventional test” (Čisar et al., 2016, p. 151), it is assumed that S and T affect PEU of computer-based programming test. It is postulated that higher S increases PEU, and lower T increases PEU, but not vice versa. Figure-1 summarizes the relationship among TAM’s variables defined above.

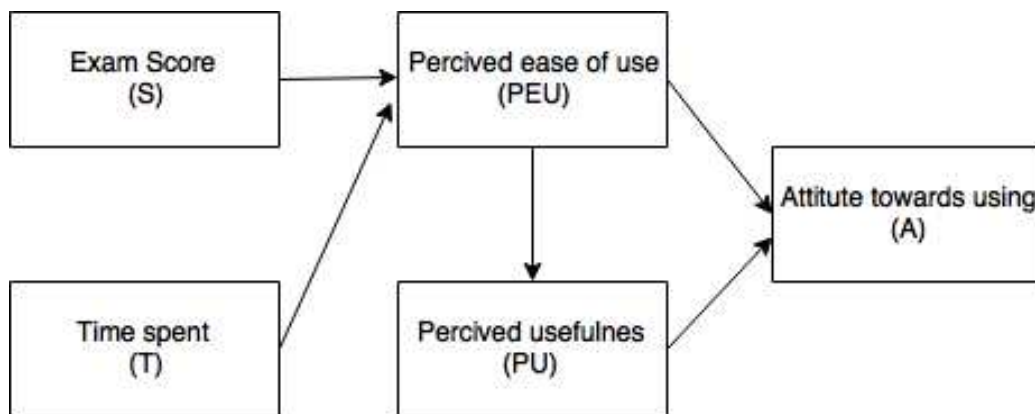


Figure-1. Relationship between TAM’s variables: Exam Score (S), Time spent (T), Perceived ease of use (PEU), Perceived usefulness (PU) and Attitude towards using (A).

2.4. Research question and hypotheses

Based on the conceptual framework and the problem statement part, the following research question and hypotheses were constructed. The research question was formulated based on the problem statement part. The first and second hypotheses were formulated based on the results reported by Čisar et al. (2016) in the conceptual framework, which state that students who took CBT scored significantly higher, and completed the CBT in significantly less time. The third, fourth, fifth, sixth and the seventh hypotheses were formulated based on the TAM part in the conceptual framework. The TAM part in the conceptual framework posits following: Systems, which are perceived as easy to use, are also perceived as useful and result in users’ favorable attitudes toward using. Systems, which are perceived as useful, also result in users’ favorable attitudes toward using. Students who scored higher in the CBT of Čisar et al. (2016) also found the CBT easy to use. Lastly, students who completed the CBT of Čisar et al. (2016) in significantly less time reported that CBT was easy to use.

Research Question

1-What are factors that affect students’ attitudes towards CBT?

Hypotheses

- 1- The mean score of the students who take the CBT will be statistically higher than that of students who take the PBT.
- 2- Students who take the CBT will complete the test in statistically significant less time.
- 3- PEU positively correlates to PU of CBT.
- 4- PEU positively correlates to A of CBT.
- 5- PU positively correlates to A of CBT.
- 6- S positively correlates to PEU of CBT.
- 7- T negatively correlates to PEU of CBT.

3. Method

3.1. Design

This study is an explanatory sequential mixed methods study consisting of both quantitative and qualitative data. Creswell (2012) describes mixed-method as “ procedure for collecting, analyzing, and ‘mixing’ both quantitative and qualitative methods in a single study or a series of studies to understand a research problem” (p. 535). For the study, the quantitative data were collected and analyzed to come to a conclusion about whether CBT creates significant exam score differences in computer science domain in comparison with the PBT. After the quantitative analysis, factors that affect students’ attitudes towards the CBT were investigated with a qualitative study.

The quantitative part of the research aimed to test the hypotheses. The quantitative part is an experimental study and confirmatory in nature. Students who participated in quantitative part were divided into control and experimental groups. Students in the control group took the paper-based versions of the tests in Appendix-A. Students in the experimental group took the computer-based version of the tests in Appendix-A.

Qualitative data were collected to answer the first research question after the quantitative phase of the study was completed. The qualitative part aimed to find the factors affecting students’ attitudes towards computer-based programming test. The main objective was to find what makes students feel comfortable or uncomfortable while taking the computer-based test, and what factors make using computer-based test difficult for them. Hence, the qualitative part was exploratory in nature. Four of the students who took the CBT were interviewed, and questions related to their interaction with CBT were asked.

3.2. Participants

Students enrolled in Computer Science related programs in higher education learn and use many programming languages. They are accustomed to type their code into an IDE with specific programming features while they are coding. They encounter a lot of problems in paper-based programming exams, so participants of this study were selected from students in higher education. In the study, the number of possible participants was limited, which led to convenience sampling. Many universities in Turkey were informed about the study and only professor teaching Programming Languages course, offered in the second year of Computer Education and Instructional Technology (CEIT) Department at Middle East Technical University (METU), accepted to participate in the study. There were 44 students enrolled in the Programming Languages course. Of the participants at METU, 28 were male with the average age of 22.14 (S.D=1.43) and 16 were female with the average age of 21.75 (S.D=1.48). Participants at METU had prior programming experience in the programming language C. Second year students at METU are taught object-oriented programming in the programming language C++. They take a programming test every week before they attend laboratory sessions. All students enrolled in the course took the test used in the study. Therefore, 22 students were assigned to the experimental group, which took CBTs, and other 22 students were

assigned to control group, which took the PBTs. Personal background variables of the students who participated in the quantitative part are summarized in the Table-2.

Table-2.

Overview of personal background variables of students at METU who participated in the quantitative part of the study.

Variable	Mean	Standard Deviation	Categories	Percentage
Gender			Male	63.66%
			Female	36.4%
Age	22	1.44		
Number of programming course taken so far	1.13	.87		
Repeating the course			Yes	93.2%
			No	6.8%
Job status			Working	93.2%
			Not working	6.8%
Prior CBT experience			Having	84.1%
			Not having	15.9%

The qualitative data were collected with students at the University of Twente in Netherlands. Professor teaching Software Systems course at the University of Twente accepted to participate in the study. Students who were enrolled in the Software Systems course were contacted via e-mail. Only volunteer students were allowed to participate in the study. 140 students were enrolled in the Software Systems course. Participating rate to the qualitative part was 6%. Only nine students from the University of Twente took the CBT. They are first-year students at the University of Twente. These students also have prior programming experience. In the Software Systems course, students master the programming language JAVA. Four of students (three males and one female) who took the CBT were interviewed.

The results of the quantitative part of the study can only be informative about students at METU because no students from any other university did participate in the study. Moreover, entry level of students at METU is different from the other universities in Turkey. For these reasons, results of the quantitative part can only be generalized to students enrolled in CEIT department at METU. Results of the qualitative part give information about Computer Science field. Participants of the qualitative part are from Netherlands, which has a different educational system than other countries. Results of the qualitative part can be generalized to only Computer Science field in the Netherlands.

3.3. Instrumentation

This study contains both quantitative and qualitative data. The quantitative data were collected through instruments such as software, two exams, a TAM questionnaire. The qualitative data were collected by semi-structured interviews.

Software and time measurement: The times spent by students taking PBT were written down on a piece of paper by the researcher as soon as students gave their exam papers to the researcher. The software recorded the time spent by students taking CBT automatically when they clicked on the Finalize button on the top right corner of the interface of the software. An interface of the software can be seen in the Figure-2. Other pictures of the software with high resolutions were included in Appendix-E.

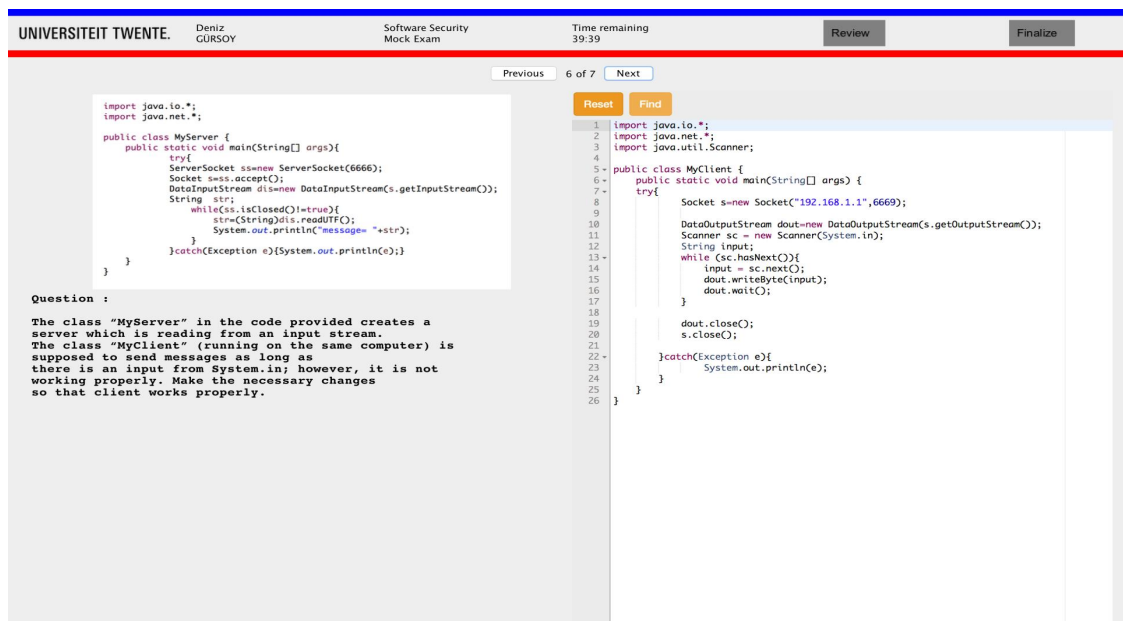


Figure-2. A screenshot of the software interface is shown in the picture. The picture shows how an open-ended question appears on the screen. The text of the question and the image are presented on the left side. Code editor used in the study can be seen on the right side.

Exams: Three different exams were used in the study. The first and second exams in the Appendix-A were used in quantitative part of the study at METU. The exams, which students at METU took, contain questions about writing a C++ class and data encapsulation in object-oriented programming. These two exams have one open-ended question and three multiple-choice questions. The third exam in the Appendix-A that was used in the qualitative part of the study has questions related to topics that are covered in Software System course namely: threads, network systems, and recursion. The third exam contains four multiple-choice questions and three open-ended. All the exams used in the study contain at least one open-ended question because open-ended questions are the questions, which students use the IDE provided to them. Students were also asked some multiple choice questions because the tests were supposed to resemble their courses' exams that contain not only open-ended questions but also multiple choice questions. Content validities of all questions were discussed with the instructors of the courses. Questions were modified according to feedback given by instructors. Grading of the exams was done on a scale from 0 to 100.

Questionnaire: Demographic information and data regarding students' attitudes were collected by a questionnaire. All students participating in the study filled the demographic part of the questionnaire in Appendix-B. The demographic data collected with the questionnaire include students' prior experience with CBT, their job status, and their prior programming experience. The demographic information was collected to explain the differences in the results of the study if any. For example, if students who had CBT before scored higher than those who had not, then it would have been concluded that having prior CBT experience has an influence on students' achievement. Students who took the computer-based test filled the full Davis' (1989) TAM questionnaire, which can be seen in Appendix-B. This questionnaire has 15 Likert-type items rated on a scale from 1 to 7. Students who took the PBTs filled TAM questionnaire with only last three statements in the questionnaire in Appendix-B to predict their attitude towards using CBT.

Interviews: The qualitative data were collected through semi-structured interviews. Students were asked four questions regarding the use of the computer-based test. To illustrate, students were asked questions like "How did you feel while providing answers to open-ended questions?" These questions were prepared with the assistance of the supervisor, who checked questions' validity and reliability.

Testing of the instruments: A pilot study was done with students who are enrolled in Computer Science in Turkey to test the software and the exams. 10 students participated in the pilot study. Students who took part in the pilot study were requested to solve the third test in Appendix-A

and report how much time is needed to complete the test, and whether the questions were written and presented in a good way. They were also asked for suggestions to improve the usability of the software. The pilot study also aimed to detect flaws in the software to prevent probable failures during data collection. After the pilot study, exam duration and questions were adjusted based on the results of the pilot study. Suggestions and minor problems such as hiding characters in the password field in the login screen were noted, and necessary adjustments were done.

The table in the Appendix-C shows how instruments were used to answer the research questions and hypotheses.

3.4. Development of the computer-based test

3.4.1. Theoretical guidelines to development of computer-based test

Computer-based tests can be used interchangeably with paper-based tests only if students taking the computer-based test are not at a disadvantage. Russell et al. (2003) say that there are three factors affecting students' performance in computer-based testing: (a) ability to review and revise responses, (b) presentation of graphics and text on computer screens, and (c) prior experience working with computers. During the development of the software, these three factors were taken into consideration. To meet the first criterion, a review button is presented on the screen so that students can view whether they answered the questions or which questions they have not answered yet. Previous and next buttons are also presented on the screen so students can freely solve or skip a question that they want to. Students are allowed to use these buttons as long as they have time left in the exam. Before exam finishes, students can go back to questions and change their answers. If exam finishes, the software automatically saves the answers given and does not let students change their answers. To meet the second criterion, software presented the graphics in high quality. It does not change the resolution of any image. If the size of the image is larger than the size of the place provided for the image, scrollbars appear on the screen so that students can see the whole image. The third criterion is assumed as met since participants in the study are Computer Science students and they are acquainted with the computers. During the test, they used the computers that they used in daily live, and most importantly they used their keyboards, so students did not any problem while using typing codes.

3.4.2. Technical information about the computer-based test

The software that delivers the test was developed in Java, which is an object-oriented programming language. The software consists of two parts, namely client-side and server-side as shown in the Figure-3. The tests are delivered to examinees in the client-side. The client-side of the software can present three types of question, which are multiple-choice questions, open-ended questions, and open-ended programming questions, by using Graphical User Interface. Clients get connected to remote server by using a technology called Remote Method Invocation (RMI) through the Internet. RMI server running on the server-side responds the requests coming from RMI clients. RMI server uses a database run in a MYSQL server to retrieve or send data such as user information, exam information, and exam questions and to save the answers examinees provide. The code editors, which examinees use when they answer an open-ended programming question, are displayed via web-engine run by the graphical user interface, which connects to the web server on the server-side. The code editor uses Ace Editor, which is a web-based editor and provides wide range of code editor from different IDEs. The code editor included in the software uses the Eclipse IDE's code editor. The code editor provides auto-completion, syntax highlighting and auto indentation features of Eclipse IDE. The web server uses the MYSQL server to display the code editor with pre-set value and to save the answer examinees provide to open-ended programming questions. The software requires Internet access as long as examinees do not finish their exam by clicking on the Finalize button in the graphical user interface. Figure-3 visualizes how software works and shows the connections between components.

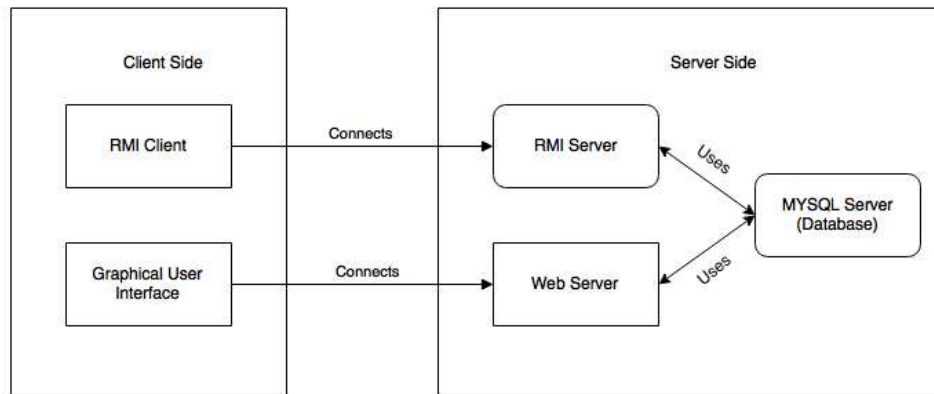


Figure-3. Picture shows the components of the software, consisting of the client-side and the server-side. Client-side connects to server-side by using the Internet connection. Client-side shows the parts used to deliver the tests to end-users like students. The server-site shows basically how the software process information.

3.4.3. Security precautions

Students who take PBTs are not allowed to use any device such as computers and mobile phone during the exam. However, students who take the CBT use internet-connected computers while answering the questions. To provide a fair testing environment, students who take CBT should not have any access to other applications and the Internet. Therefore, the software delivers the test in full-screen mode. The full-screen mode does not allow students to switch between applications and execute any other application, so students who take CBT do not have any chance to use the Internet to search for answers or message each other, or to open a compiler to test if the answers they provided are correct or not.

To ensure the security of the data, students needed to authenticate the software. The International Test Commission (2006) groups the software's security application in Controlled Mode and defines the Controlled Mode as "no direct human supervision of the assessment session is involved, but the test is made available only to known test-takers. Internet tests will require test-takers to obtain a login username and password. These often are designed to operate on a one-time-only basis"(p.144). To authenticate, students are given private usernames and passwords. Students' data are also stored in the database by the usernames. Therefore, students' answers and time spent in the exam are stored correctly in the database. Students' names also are searched in the database by using the usernames given to students and displayed on the screen, so students know that nothing is going wrong and they are using the correct username or computer. Displaying name of the students is also important to catch students who attempt cheating in a case; for example, students change the seats during the test or someone else takes the test instead of students who is supposed to take the test.

3.5. Procedure

The study required human test subject so the first step was to get the confirmation from ethical committee at the University of Twente. Necessary documents and forms were submitted to ethical committee at the University of Twente. After the ethical committee at the University of Twente approved the study, data collection started with collection of the quantitative data. The quantitative data were collected at METU in the Programming Languages course offered in the CEIT department. Students who are registered for the course also attend weekly laboratory sessions and take an exam before they do the lab activity. The laboratory sessions are done in two different days: Monday and Wednesday. Students who attend the laboratories in different days took equivalent exams with different questions because there was a possibility of cheating. Students who took the exam on Monday might have given the exam questions students who were going to take the exam on Wednesday. Therefore, two different exams were used in the quantitative part of the study. Students who attended the laboratory on Monday took the first exam in the Appendix-A. Students who attended laboratory on Wednesday took the second exam in the Appendix-A. Students, who

participated in the quantitative part, were randomly assigned to conditions namely: control and experimental group. Gender was used as a blocking variable in the random assignment. An equal number of the female and male students were assigned to control and experiment group. Students in the control group took the PBTs. Students in the experimental group took the CBTs. There was one supervisor in the classes where students took the exams so that students could not cheat. Students in both conditions started the exam at the same time and in the same place. The software blocked students from switching between applications. Students used only the software's interface delivering the test during the entire exam duration. Students were given 20 minutes to answer all questions in the exams. Preventing cheating or talking during the exam was important because the independence of the observations assumption should be met in this study so we can run the statistical test and get significant results. All students who participate in the study filled the questionnaire. Students who took the CBTs filled the TAM questionnaire immediately after the exam. Students were not informed about their exam scores while filling the TAM questionnaire. Students' performances in the exam were graded in a week, and then all students who participated in the study took feedback about their exam performances.

Qualitative data were collected from students at the University of Twente. Students at the University of Twente were contacted through the Blackboard site of UT. They asked if they would participate in the study. Students who wanted to participate in the study registered through a Google Form. Google form was used because it allowed students to register easily and whenever they wanted. Each student who registered to study received an e-mail informing about the study. Date of the exam and location where the students would take the exam were decided after a discussion with volunteering students. On the exam date, students who showed up in the exam place took the third test in the Appendix-A. Students' exams were graded in two days. Then, detailed feedbacks about their performance were provided to all participants. Four students were selected for interview based on their exam results. Students who scored highest participated in the qualitative part. The qualitative data were collected through semi-structured interviews. Students were asked open-ended questions regarding their interaction with computer-based programming test. Interviews lasted 15 minutes.

After the data analysis had been completed, all students who participated in the study were informed about the result of the study.

3.6. Data analysis

Data analysis began with quantitative data analyses. One of the concerns in this study is the reliability of the exams. Students' exam scores were entered into SPSS software. Exams' reliability was checked with improved split-half reliability method, which checks the Pearson correlation coefficients among high, moderate and low achievers in the exams. After all students' exam performances were graded, an independent-samples t-test was run to compare the mean scores of the control and experimental group to test the first hypothesis. Reliability of the TAM questionnaire was checked by using Cronbach's alpha value. Spearman's rho rank order correlation method was used to calculate the correlations among the variables PEU, PU, S, T and A to test third, fourth, fifth sixth and seventh hypotheses. Time students spent in the exam was compared by an independent-samples t-test to determine which group spent less time in the exams, and to test the second hypothesis.

Secondly, the qualitative data were collected through interviews. Interviews were recorded. All records were transcribed by listening to the record, and writing words on the records into a text file in a computer. No software was used in the transcription process. At the end of transcription, four different text documents were obtained. After the transcriptions had been done, text data were analyzed by coding. Different descriptive codes, which summarize the primary topic of the statements of students in test documents, were produced. Themes and the subgroups were created from the codes, which refer to the same domain. A coding scheme was created from using these themes and groups. The coding scheme used in qualitative analyses is shown in the Appendix-D. The qualitative data analysis aimed to answer the first research question.

The table in the Appendix-C provides an overview of by which instruments data were collected, how data were measured, which source data came from, and how data was used to answer research questions and hypothesis.

4. Results

In this section, results of the study were reported in the order that analyses were done. All separate analyses were reported under different titles. Reliabilities of the exams and the TAM questionnaire were reported in the beginning. Subsequently, the responses of the student who filled TAM questionnaire were reported in a table. All the results regarding the hypotheses were written under titles stating the hypotheses. Finally, results of the qualitative analysis were reported at the end of this section.

Reliability of exams

Reliability of the exams was calculated by improved Split-half reliability method. In reliability calculation, first and second exams in Appendix-A are considered equivalent. Reliability was calculated as following: students' scores were sorted in ascending order. Scores were divided into three groups namely: low, moderate and high achievers. Exam questions were also divided into two groups namely: easy and difficult questions. The difficulty of each exam questions was calculated by counting the numbers of the correct answers given to each question. Three different Pearson-product moment correlation coefficients between students' scores on easy and difficult questions were calculated among low, moderate and high achievers. Correlations were shown in the table Table-3.

Table-3.

Correlation between easy and difficult questions based on groups of low, moderate and high achievers.

Group	Correlation coefficient	Significance	n
Low achievers	-.385	.174	14
Moderate achievers	-.810**	.000	15
High achievers	.122	.665	15

***Correlation is significant at the .01 level.*

Reliability of the TAM questionnaire

Reliabilities of three TAM constructs PEU, PU and A in the questionnaire were determined by Cronbach's alpha values. Cronbach's alpha values for the six perceived ease of use statements, six perceived usefulness statements and three attitudes towards use statements in 22 filled TAM questionnaire were .94, .60 and .78, respectively. However, the reliability of perceived usefulness construct was relatively low in comparison with those of other two constructs. Last three perceived ease of use statements might be misleading, or students might have filled their answers by considering their answers to the questions but not their interaction with the computer-based test. Cronbach's alpha value for perceived usefulness construct was recalculated after last three statements were removed. The Cronbach's alpha value increased from .60 to .77 and the correlation between PEU and PU became significant after the last three statements were taken out of the calculation. Table-4 shows the Cronbach's alpha values of the TAM constructs.

Table-4.

Reliability of the TAM constructs.

Construct	Alpha value	Number of statements
Perceived ease of use (PEU)	.94	6
Perceived usefulness (PU)	.77	3
Attitude towards use (A)	.78	3

The responses of the student who filled the TAM questionnaire

In the study, students who took the PBTs filled the TAM questionnaire with the only construct of attitude towards the use of the CBT to see whether they think that using CBT in programming exams is a good idea or not. Percentage of the given answer by all students can be seen the in Table-5.

Table-5.*Percentage of responses given by students who filled the TAM questionnaire's attitude statements*

Statement	Categories	PBT	CBT
I dislike the idea of computer-based test in programming exams.	Strongly disagree	31,8%	18%
	Disagree	36,4%	27,3%
	Partly disagree	18%	4,5%
	Neutral	9%	27,3%
	Partly agree	0%	9%
	Agree	0%	9%
	Strongly agree	4,5%	4,5%
I have a generally favorable attitude toward using computer-based test in programming exams	Strongly disagree	0%	0%
	Disagree	0%	0%
	Partly disagree	9%	4,5%
	Neutral	13,6%	31,8%
	Partly agree	4,5%	22%
	Agree	45,5%	27,3%
	Strongly agree	27,3%	13,6%
I believe it is a good idea to use computer-based test in the next programming exams.	Strongly disagree	4,5%	4,5%
	Disagree	9%	0%
	Partly disagree	0%	4,5%
	Neutral	13,6%	18%
	Partly agree	9%	22%
	Agree	36,4%	22%
	Strongly agree	27,3%	27,3%

Hypothesis-1: The mean score of the students who take the CBT will be statistically higher than that of students who take the PBT.

An independent-samples t test was conducted to evaluate the hypothesis that the mean score of the students who take the CBT will be statistically higher than that of students who take the PBT. The test was not significant, $t(42) = 1.44, p = .15$. Students who took PBT ($M = 62.64, SD = 20.75$) had higher average than those who took CBT ($M = 53.74, SD = 20.38$). The 95% confidence interval for the difference in means was ranging from -3.65 to 21.42.

Table-6.*T-test results comparing exam scores of students on conditions CBT and PBT.*

Condition	n	Mean	SD	t-cal	t-crit	df	p	Decision
CBT	22	62.64	20.75	1.44	1.98	42	.153	Not Rejected
PBT	22	53.72	20.38					

Hypothesis-2: Students who take the CBT will complete the test in statistically significant less time.

An independent-samples t test was conducted to evaluate the hypothesis that students who take the CBT will complete the test in statistically significant less time. The test was significant, $t(42) = 3.22, p = .00$, and the results were supporting the research hypothesis. Students who took CBT ($M = 815.22, SD = 190.70$) completed the test in less time than those who took PBT ($M = 989.73, SD = 167.24$). The 95% confidence interval for the difference in time was ranging from 65.37 to 283.64.

Table-7.*T-test results comparing completion time of students on conditions CBT and PBT.*

Condition	n	Mean	SD	t-cal	t-crit	df	p	Decision
CBT	22	815.22	190.71	3.22	1.98	42	.002	Rejected

Hypothesis-3: PEU positively correlates to PU of CBT.

A Spearman's rank-order correlation was run to determine the relationship between TAM variables PEU and PU. There was moderate positive correlation between PUE and PU, which was statistically significant ($r_s(22) = .560, p = .007$) (Hinkle, 2003).

Hypothesis-4: PEU positively correlates to A of CBT.

A Spearman's rank-order correlation was run to determine the relationship between TAM variables PEU and A. There was moderate positive correlation between PEU and A, which was statistically significant ($r_s(22) = .560, p = .007$) (Hinkle, 2003).

Hypothesis-5: PU positively correlates to A of CBT.

A Spearman's rank-order correlation was run to determine the relationship between TAM variables PU and A. There was low positive correlation between PU and A, which was not statistically significant ($r_s(22) = .383, p = .079$) (Hinkle, 2003).

Hypothesis-6: S positively correlates to PEU of CBT.

A Spearman's rank-order correlation was run to determine the relationship between TAM variables S and PEU. There was no correlation between S and PEU, which was not statistically significant ($r_s(22) = .076, p = .737$) (Hinkle, 2003).

Hypothesis-7: T negatively correlates to PEU of CBT.

A Spearman's rank-order correlation was run to determine the relationship between TAM variables T and PEU. There was no correlation between T and PEU, which was not statistically significant ($r_s(22) = -.137, p = .543$) (Hinkle, 2003).

Based on the Spearman's rank-order correlation analysis, correlation among TAM variables can be visualized as in the Figure-4.

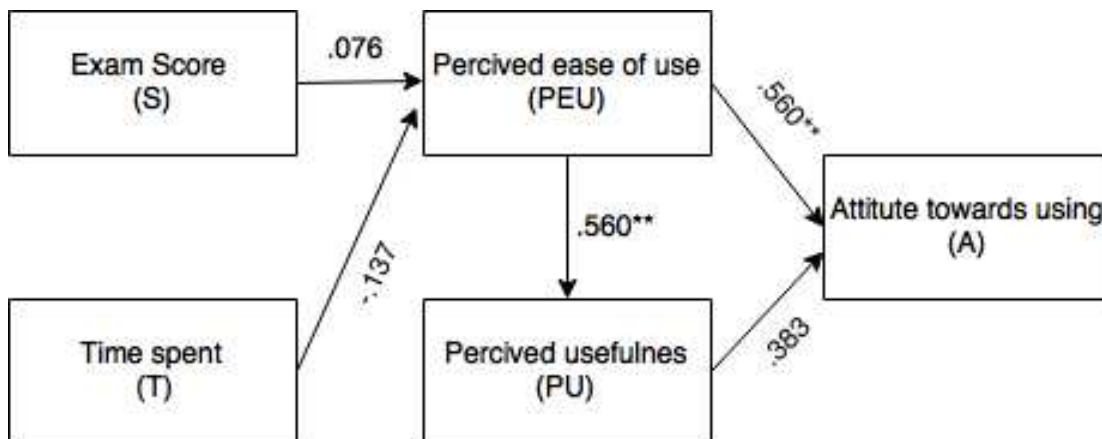


Figure-4. Spearman's rho ranked order correlation coefficients between TAM variables: Exam score (S), Time spent (T), Perceived ease of use (PEU), Perceived usefulness (PU), Attitude towards using (A).

** Significant at p value < .01

Research Question-1: What are factors that affect students' attitudes towards CBT?

To answer the research question, four students from the University of Twente were interviewed, and they were asked questions related to their interaction with CBT. Based on the analysis of the interviews following factors appear to be affecting students' attitudes towards CBT.

1. Code editor:

Most interview respondents indicated that it was a pleasant experience to take the test on the computer and use the code editor in the exam. They felt familiar with the code editor because it was like the one they use in their course. They report why they liked the code editor as following:

- *There are a lot of other features that are more influencing my results than only color highlighting. For example, you can erase line and bring back your lines. You do not have to erase your whole answer on paper. You can just use backspace and indent automatically. The ways code is written, nicely formatted. You do not have your ugly handwriting. You can be really quick. Things like that really improved my performance.*
- *In paper-based exam, I had to correct very much and put something in something it looks not so good. It is easy if you put it on the computer. Also highlighting was also quite handy. It just looks more ordered.*
- *It is really nice to see that it is highlighted it was very familiar. I liked it.*
- *It feels quite familiar. That is a lot easier than writing it down by pen. It is easier mostly because of the changes, you just stop writing on paper if you forget one thing in the beginning, you have to write whole thing again. You can just hit enter few times in computer-based test and type what you need.*

2. Design of the software:

Interview respondents identified some features of the software that can be improved and would increase student's concentration and performance during the exam such as location review button, which they never used in the exam, presentation of questions and choices, and background color of software. Comments regarding those features are following:

- *I did not know what review button was for because it was next to finalize button I might think it was review by teacher or something like that. It should be somewhere else but not next to the finalize button which I do not want to press before I am done.*
- *Questions jump up and down if you press the next button. Always bring it to the left side or right side so you know where you focus for the questions.*
- *Grey background is a kind of harder to concentrate.*

3. Pictures in questions:

There is no consensus among interview respondents about the use of colored pictures that show highlighted codes, in the software. Specific comments regarding the images included the following:

- *They are colored. It is definitely easier than paper-based test. In paper-based test, there are not colors. They are printed black and white. I think that was easier.*
- *Pictures were fine. It really did not matter. There was no problem in the pictures.*

5. Discussion

Discussions of the reliability of the exams

Reliability of the exams was checked by improved Split-half reliability method. In a reliable test, high achievers are supposed to have high scores on both easy and difficult questions. Additionally, moderate and low achievers are supposed to have equivalent scores on both easy and difficult questions. To check whether high achievers have high score on both easy and difficult questions, a Pearson correlation coefficient was calculated between scores of high achievers' scores on both easy and difficult questions and shown in the Table-3. There is no correlation between scores of high achievers' scores on both easy and difficult questions and it is concluded that high achievers do not get high scores on both easy and difficult questions. To check whether moderate and low achievers have equivalent scores on both easy and difficult questions, two Pearson correlation coefficients were calculated between scores of moderate and low achievers' scores on both easy and difficult questions and shown in the Table-3. There is moderate negative correlation between scores of moderate and low achievers' scores on both easy and difficult questions and it is concluded that moderate and low achievers do not get equivalent scores on both easy and difficult questions.

Therefore, it was concluded that exams used in the study are not reliable. The exams should not be used in future.

Discussions of reliability of the TAM questionnaire

Cronbach's alpha value is used to determine the internal consistency of the set of items in the each TAM constructs in and values can be seen in Table-4. According to George and Mallery (2003), Reliability of perceived ease of use construct in the TAM questionnaire with alpha value .94 is considered excellent. Reliability of perceived usefulness construct in the TAM questionnaire with alpha value .77 is considered questionable. Reliability of attitude towards use construct in the TAM questionnaire with alpha value .78 is considered acceptable. There is no construct in the TAM questionnaire, which has poor reliability coefficient. Therefore, the TAM questionnaire is a reliable source that can be used in the study.

Discussion of the responses of the student who filled the TAM questionnaire

All students participated in the study filled the TAM questionnaires' attitude towards using statements. The percentages of the responses given by the students are presented in the Table-5. As shown in the Table-5, 82% of the students who took PBT reported that they like the idea of computer-based programming test. 77% of those had a favorable attitude towards computer-based programming test, and 72% of those believed it is a good idea to use CBT in the next programming exams. Additionally, 50% of the students who took CBT reported that they like the idea of computer-based programming test. 62% of those had a favorable attitude towards computer-based programming test, and 71% of those believed it is a good idea to use CBT in the next programming exams. In short, the overwhelming majority of the students show a favorable attitude towards computer-based programming test and that indicates the need for using a computer-based test in future programming course's exams.

Discussion of results of hypotheses

An independent-samples *t*-test was conducted to test if there is significant mean score difference between students who took CBT and PBT and the result was presented in Table-6. Based on the T-test result, it is concluded that:

- There is no significant mean score difference between students who took CBT and those who took PBT.
- CBT can be used interchangeably with PBT without any concern of equivalency of exam score in programming related courses.

Another independent-samples *t*-test was conducted to test if students who take the CBT will complete the test in statistically significant less time, and the result was presented in Table-7. Based on the T-test result, following is concluded that:

- There is a significant time difference between students who took CBT and those who took PBT with the certainty of 95% and a risk of 5%.
- Students who take CBT face fewer problems in writing code during the exam in comparison with students who took PBT; thus, complete the test earlier.

Correlations among TAM variables are calculated by running Spearman's rank-order correlation test. Based on the Spearman's rank-order correlation results, following is concluded that:

- There is a medium and significant correlation between PEU and A of CBT. Therefore, students who find the software easy to use also tend to use CBT in future programming exam.
- There is the medium and significant correlation between PEU and PU of CBT. Students who consider CBT as easy to use also think that CBT is useful.
- There is low and not significant positive correlation between PU and A of CBT. Students who find CBT useful also tend to use CBT in future. Correlation is not high because students might have encountered some usability problems in the design of the software, which may prevent students from using the software in future programming exams.

- S does not relate to students' PEU of CBT. Both students who scored high and students who scored low do not judge the ease of use of CBT based on their score in the exam.
- There is a low and not significant correlation between T and PEU of CBT, and the correlation is negative. That means students who spent less time in exam might consider CBT as easy to use.

Discussion of qualitative analysis

Qualitative analyses were made to determine the factors that affect students' attitudes towards the CBT. Students reported that code editor facilitated adding new lines and changing the answers. They also reported that codes written into the code editor look more organized in comparison with codes written on paper. Respondents said it was a pleasant experience to take the test on the computer and use a code editor in the exam. In short, students' interactions with the CBT in programming course confirm the problem and show the need for delivering programming test via CBT.

Furthermore, it is also concluded from interviews that software design plays a key role in determining students' attitudes towards the CBT. Firstly, presentation of the questions in the software must be consistent. Places where the texts appear, where the images appear, and where the choices are listed should not change from question to question. Students should not see parts of questions moving when they switch between questions. Seeing the places of text, images or choices moving in the screen distracts students during the test. Students favor consistent design more. Secondly, frequently used buttons should not be located close to buttons that students do not want to click on them as long as they are not done with them exam such as the Finalize button. For example, in this case, the review button was not used by students and therefore, it is need to be placed next to previous and next buttons, which are frequently used by students.

Lastly, the use of colored images, which shows highlighted codes, in the software make students trace the code more quickly. In the study, students did not make any negative comment on the use of images. The use colored of images provides advantages in tracing the coded as long as images are displayed in high resolutions. If images are presented with low resolutions, students do not show the favorable attitude towards CBT.

6. Conclusion

This study aimed to show what is the impact of code editor on students' exam performance. Results of the analysis indicate that students show the favorable attitude towards the use of CBT in the programming course. Use of CBT in programming exams does not create significant score difference in comparison with the use of PBT. Students who took the CBT finished the test statistically in less time than those who took the PBT. Students describe taking CBT in programming course as a pleasant experience. The research shows the need for using CBT in programming courses.

Finding sample was one of the challenges in the study. Professors, we contacted, who taught programming courses at universities did not want to participate in the study for many reasons. Many of them did not use any exams at all in their courses because they grade their students based on students' assignments and projects. Some professors had already done the exams when we contacted them. Some professors were cautious about loss of the exam data, and they did not want to use the software in the final or midterm exams. For these reasons, one of the limitations of this study was not to use the software in the midterm and final exams. In this study, data was collected in quizzes, which students take before they begin their laboratory sessions. Future research should focus on the use of the software during a semester, which students take a midterm and final exams, in an experimental setting.

The lack of sample also reduced the scope of the study. The role of gender is a currently discussed factor in the computer science field. Since there were not enough females participants, the achievement difference between female and male participants was not checked in the study.

Another limitation of the study was not to use the same exam in the quantitative part of the study. The quantitative data collected with two equivalent exams with different questions to prevent

cheating. That might have affected the results of the study. There was no significant score difference between students who took the CBT and PBT. That might be because of the use of two exams, which were assumed equivalent. The study should be repeated with data collected in same exam.

Code editor used in the study was only able to evaluate code by highlighting keywords in the code. Future researchers should include a code editor, which can evaluate the code and gives errors if there is any logical error in the code. Code editor should be able to list all the class' methods when students invoke them with "." mark from an object of that class.

This research was conducted with students in higher education. Giving visual feedback regarding the code written by students might be instructive and useful in vocational high schools. Students who take their test from this software might learn more than those who take a traditional exam at the end of a semester. Two or more vocational high school can be selected for a future research and their students' performance can be compared at the end of a semester by taking their prior knowledge into consideration in an experimental setting to see whether using code editor contributes vocational high school students learning or not.

7. References

- Akdemir, O., & Oguz, A. (2008). Computer-based testing: An alternative for the assessment of Turkish undergraduate students. *Computers and Education, 51*(3), 1198–1204.
- Anakwe, B. (2008). Comparison of Student Performance in Paper-Based Versus Computer-Based Testing. *Journal of Education for Business, 84*(1), 13–17.
- Aybek, E. C., Şahin, D. B., Eriş, H. M., Şimşek, A. S., & Köse, M. (2014). Meta-analysis of comparative studies of student achievement on paper-pencil and computer-based test. *Asian Journal of Instruction, 2*(2), 18–26.
- Bayazit, A., & Aşkar, P. (2012). Performance and duration differences between online and paper-pencil tests. *Asia Pacific Education Review, 13*(2), 219–226.
- Björnsson, J.K., 2008. Changing Icelandic national testing from traditional paper and pencil based tests to computer based assessment: some background, challenges and problems to overcome. In: F.S. Scheuermann and A.G. Pereira (Eds.), *Toward a research agenda on computer based assessment: challenges and needs for European educational measurement*. Ispra (VA), Italy: European Commission.
- Bugbee J., A. C., & Bernt, F. M. (1990). Testing by computer: findings in six years of use 1982-1988. *Journal of Research on Computing in Education, 23*(1), 87-101.
- Bunderson, V. C., Inouye, D. K., & Olsen, J. B. (1989). *The four generations of computerized educational measurement*. In R. L. Linn (Ed.), *Educational measurement: Third edition* (pp. 367- 407). New York: Macmillan.
- Cerillo, T.L., & Davis, J.A. (2004). *Comparison of paper- based and computer based administrations of high- stakes, high-school graduation tests*. Paper presented at the Annual Meeting of the American Education Research Association, San Diego, CA.
- Cheang, B., Kurnia, A., Lim, A., & Oon, W. C. (2003). On automated grading of programming assignments in an academic institution. *Computers and Education, 41*(2), 121–131.
- Čisar, M. S., Čisar, P., & Pinter, R. (2016). Evaluation of knowledge in Object Oriented Programming course with computer adaptive tests. *Computers & Education, 92-93*, 142–160.
- Creswell, J. W. (2012). *Educational research: planning, conducting, and evaluating quantitative and qualitative research*. Boston: Pearson Education.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319-340.
- Davis, F. D. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioral impacts. *International Journal of Man–Machine Studies, 38*, 475–487.
- Davis, F. D., Bagozzi, R., & Warshaw, P. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management Science, 35*, 982-1003.

- Educational Testing Service. (1992). *Computer-based testing at ETS 1991–1992*. Princeton, NJ: Author.
- Fair Test: The National Center for Fair and Open Testing. (2007, August 20). *Computerized Testing: More Questions than Answers*. Retrieved from <http://www.fairtest.org/facts/computer.htm>
- Frein, S. T. (2011). Comparing In-Class and Out-of-Class Computer-Based Tests to Traditional Paper-and-Pencil Tests in Introductory Psychology Courses. *Teaching of Psychology, 38*(4), 282–287.
- Fylaktopoulos, G., Goumas, G., Skolarikis, M., Sotiropoulos, A., Athanasiadis, D., & Maglogiannis, I. (2015). CIRANO: An Integrated Programming Environment for Multi-tier Cloud Based Applications. *Procedia Computer Science, 68*, 42–52.
- Hinkle, D. E., Wiersma, W., & Jurs, S. G. (2003). *Applied statistics for the behavioral sciences*. Boston, Mass: Houghton Mifflin.
- Jeong, H. (2014). A comparative study of scores on computer-based tests and paper-based tests. *Behaviour & Information Technology, 33*(4), 410–422.
- Karay, Y., Schaubert, S. K., Stosch, C., & Schüttpelz-Brauns, K. (2015). Computer Versus Paper—Does It Make Any Difference in Test Performance?. *Teaching and Learning in Medicine, 27*(1), 57–62.
- Khader, A., & Almasri, M. (2014). The Influence On Mobile Learning Based On Technology Acceptance Model (Tam), Mobile Readiness (Mr) And Perceived Interaction (Pi) For Higher Education Students. *International Journal of Technical Research and Applications, 2*(1), 5–11.
- Lee, G., & Weerakoon, P. (2001). The role of computer-aided assessment in health professional education: a comparison of student performance in computer-based and paper-and-pen multiple-choice tests. *Medical Teacher, 23*(2), 152-157.
- McFadden, C. A., Marsh II, E. G., & Price, B. J. (2001). Computer Testing in Education, *Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research, 18*(2), 43-60.
- Natal, D. (1998). *On-line Assessment: What, Why, How*. Paper presented at the Technology Education Conference, Santa Clara, California. May 6, 1998. 1-23.
- Özalp, Y. Ş., & Çağiltay, N. E. (2010, April 14-16). *Paper-Based versus Computer-Based Testing in Engineering Education*. Paper presented at IEEE EDUCON 2010 Conference: The Future of Global Learning Engineering Education, Madrid. doi:10.1109/EDUCON.2010.5492397
- Pawasauskas, J., Matson, K. L., & Youssef, R. (2014). Transitioning to computer-based testing. *Currents in Pharmacy Teaching and Learning, 6*(2), 289–297.
- Pomplun, M., Frey, S., & Becker, D. F. (2002). The score equivalence of paper-and-pencil and computerized versions of a speeded test of reading comprehension. *Educational and Psychological Measurement, 62*, 337-354.
- Russell, M. (1999). Testing on computers: A follow-up study comparing performance on computer and on paper. *Education Policy Analysis Archives, 7*, 20.

Russell, M., Goldberg, A., & O'connor, K. (2003). Computer-based Testing and Validity: a look back into the future. *Assessment in Education: Principles, Policy & Practice*, 10(3), 279–293.

The International Test Commission. (2006). International Guidelines on Computer-Based and Internet-Delivered Testing. *International Journal of Testing*, 6(2), 143–171.

Appendix A: Exam Questions

Exam-1

Question-1

Define a class to represent Cylinder with height and radius.(take pi value as 3)

It will have

- A Constructor to set the height and radius data members to the parameters it receives
- Member function (get methods) to return each data member
- Member function to compute and return area of the Cylinder
- Create an instance of the Cylinder object to test each member function by calling them

```
#include<iostream>
using namespace std;
```

```
int main(){
```

```
};
```

Question-2

A class is considered a template of _____.

- A. An object
- B. A variable
- C. A value
- D. An attribute

Question-3

A member function is defined _____.

- A. Outside the class
- B. inside the class
- C. Either a or b
- D. none of these

Question-4

A program can access the private members of a class _____.

- A. directly
- B. only through other private members of the class
- C. only through other public members of the class
- D. none of above – a program cannot access the private members of a class in any way

Question-5

A function that is called automatically each time an object is destroyed is a _____.

- A. Destructor
- B. Destroyer
- C. Remover
- D. Terminator

Exam-2

1-Define a class to represent Rectangle with height and width. It will have

- A Constructor to set the height and width data members to the parameters it receives
- Member function (get methods) to return each data member
- Member function to compute and return area of the rectangle
- Create an instance of the Rectangle object to test each member function by calling them

```
#include<iostream>  
using namespace std;
```

```
int main()  
{
```

```
};
```

2.	What is the difference between an object and a class? Please select the best answer.
<input type="radio"/>	A. An object is an extension of the class construct whose default access privilege is public.
<input type="radio"/>	B. The term <i>object</i> is just another way of referring to the public data members of a class.
<input type="radio"/>	C. An object is an initialized class variable.
<input type="radio"/>	D. A class is an initialized object variable.

3.	An object is _____ of a class. Please select the best answer.
<input type="radio"/>	A. an instance
<input type="radio"/>	B. an interface
<input type="radio"/>	C. an encapsulation
<input type="radio"/>	D. a member function

4.	To initialize the data members of a class, Which of the following is used?
<input type="radio"/>	A. Constructor
<input type="radio"/>	B. Destructor
<input type="radio"/>	C. Data members are automatically initialized
<input type="radio"/>	D. All the previous answers are incorrect

5.	Is the following statement true or false? “A struct can have member functions.”
<input type="radio"/>	True
<input type="radio"/>	False

Exam-3

Question 1:

Which statement is true for the following code?

```
public class Rprcraven{
    public static void main(String argv[]){
        Pmcraven pm1 = new Pmcraven("One");
        pm1.run();
        Pmcraven pm2 = new Pmcraven("Two");
        pm2.run();
    }
}
```

```
class Pmcraven extends Thread{
    private String sTname="";
    Pmcraven(String s){
        sTname = s;
    }
    public void run(){
        for(int i =0; i < 2 ; i++){
            try{
                sleep(1000);
            }catch(InterruptedException e){}
            System.out.println(sTname);
        }
    }
}
```

- A) Compile time error, class Rprcraven does not import java.lang.Thread
- B) One One Two Two
- C) One Two One Two
- D) Compilation but no output at runtime

Question 2:

Given:

```
1. public class MyRunnable implements Runnable {
2.     public void run(){
3.         System.out.println("running...");
4.     }
5.     public static void main (String[] args){
6.         //insert code here
7.     }
8. }
```

Which of these , inserted independently at line 6, will create and start a new thread?
(Choose all that apply.)

- A) new Runnable(MyRunnable).start();
- B) new Thread(MyRunnable).run();
- C) new Thread(new MyRunnable()).start();
- D) new MyRunnable().start();

Question 3:

Consider these methods.

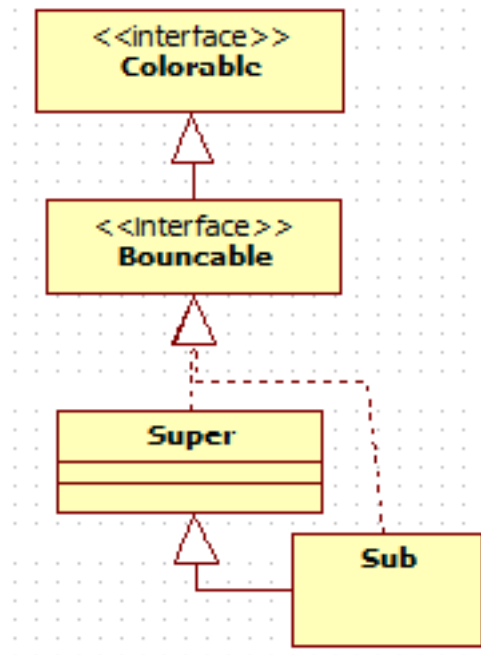
- ⇒ run()
- ⇒ start()
- ⇒ join()
- ⇒ sleep()
- ⇒ currentThread()

How many of them declare a checked exception? (1 correct answer)

- A) One.
- B) Two.
- C) Three.
- D) All.

Question 4:

Class diagram



Tester class

```
interface Colorable { }
```

```
interface Bouncable extends Colorable { }
```

```
class Super implements Bouncable { }
```

```
class Sub extends Super implements Bouncable { }
```

```
public class Tester {
```

```
    public static void main(String[] args) {
```

```
        System.out.println(new Sub() instanceof Super); //line 1
```

```
        System.out.println(new Sub() instanceof Bouncable); //line 2
```

```
        System.out.println(new Sub() instanceof Colorable); //line 3
```

```
        System.out.println(new Super() instanceof Sub); //line 4
```

```
        System.out.println(new Super() instanceof Colorable); //line 5
```

```
    }
```

```
}
```

The class diagram above shows the dependencies among classes **Colorable**, **Bouncable**, **Super** and **Sub**. Based on this class diagram, which lines of the main method of the **Tester** class above will evaluate to true?

- A) All lines will evaluate to true
- B) Line 4 evaluates to false, all other lines evaluate to true
- C) Only line 1 and 2 will evaluate to true

Question 5:

```
public class Movie {  
  
    private String name;  
    private int rank;  
    private int year;  
    private double ranking;  
  
    public void setName(String name){  
        this.name=name;  
    }  
    public String getName(){  
        return name;  
    }  
    public void setRank(int rank){  
        this.rank=rank;  
    }  
    public int getRank(){  
        return rank;  
    }  
    public void setYear(int year){  
        this.year=year;  
    }  
    public int getYear(){  
        return year;  
    }  
    public void setRanking(double ranking){  
        this.ranking=ranking;  
    }  
    public double getRanking(){  
        return ranking;  
    }  
}
```

The Top 250 films in IMDB (Internet Movie Database) are listed in the file input.txt in the format shown below. In this file, each line contains information about a movie separated with “\”.

For example:

Line 1 is : 1\The Shawshank Redemption\1994\9.2

```
Rank of the movie: 1  
Name of the movie: The Shawshank Redemption  
Released year: 1994  
Ranking of the movie: 9.2
```

Complete the code provided so that all the movies are stored in a list and movies that are released in a selected year are printed to the console. Use the class provided to store the movie information. (See the next page)


```

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {
        //Create a list that can store Movie objects

        try{
            InputStream ips=new FileInputStream("input.txt");
            InputStreamReader ipsr=new InputStreamReader(ips);
            BufferedReader br=new BufferedReader(ipsr);
            String line;
            Movie temp;
            // write the code that read the file line by line to while loop
            while (
                String[] splitted = line.split("\\\\");
                temp= new Movie();
                // assign the rank, name, year, ranking read to temp object
                // and add temp object to list that you created

                }
            br.close();
        }
        catch (Exception e){
            System.out.println(e.toString());
        }
        Scanner in = new Scanner(System.in);
        int selectedYear=in.nextInt();
        // print all the movies in the list that belong to selected year

    }
}

```

Question 6:

```
import java.io.*;
import java.net.*;

public class MyServer {
    public static void main(String[] args){
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept();
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String str;
            while(ss.isClosed()!=true){
                str=(String)dis.readUTF();
                System.out.println("message= "+str);
            }
        }catch(Exception e){System.out.println(e);}
    }
}
```

The class MyServer in the code provided creates a server that is reading from an input stream. The class MyClient (running on the same computer) is supposed to send messages as long as there is an input from System.in; however, it is not working properly. Make the necessary changes so that client works properly. Rewrite the class working properly to the back of this page.

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

public class MyClient {
    public static void main(String[] args) {
        try{
            Socket s=new Socket("192.168.1.1",6669);
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            Scanner sc = new Scanner(System.in);
            String input;

            while (sc.hasNext()){
                input = sc.next();
                dout.writeByte(input);
                dout.wait();
            }

            dout.close();
            s.close();

        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Question 7:

Write a Java method that finds and prints out all the possible combinations (or “permutations”) of the characters in a string. So, if the method is given the string “dog” as input, then it will print out the strings “god”, “gdo”, “odg”, “ogd”, “dgo”, and “dog” – since these are all of the possible permutations of the string “dog”. Even if a string has characters repeated, each character should be treated as a distinct character – so if the string “xxx” is input then your method will just print out “xxx” 6 times. Make sure that you use recursion.

You may use a second helper method. Base your implementation on the following method signature:

```
public void permute (String input);
```

Appendix B: TAM Questionnaire

Full name: _____ Age: _____

1- Select your gender.

 Male Female

2- Have you taken any programming course prior to this course? If yes, how many courses have you taken?

 Yes _____ No

3- Are you repeating this course? If yes, how many times did you take this course before?

 Yes _____ No

4- Are you working in a job related with software engineering or programming?

 Yes No

5- Have you taken a paper-based programming test before?

 Yes No

6- Have you taken a computerized programming test before?

 Yes No

7- Have you taken a computerized test in any other course or exam before?

 Yes No

Technology Acceptance Model Questionnaire

Please place an "X" in the appropriate box to rate the following items using a scale of 1-7:

1=Strongly Disagree, 4= Neutral, 7=Strongly Agree

	Strongly Disagree	Neutral					Strongly Agree
	1	2	3	4	5	6	7
The computer-based test enables me to answer questions more quickly.							
Computer-based test has improved my performance in the exam							
Computer-based test makes it easier to solve the questions.							
Computerized-tests improve my productivity							
Computer-based test gives me greater control over the exam.							
Computer-based test enhances my effectiveness on the exam.							
	1	2	3	4	5	6	7
My interaction with computer-based test has been clear and understandable.							
Overall, computer-based test is easy to use.							
Learning to use computer-based test system was easy for me.							
I rarely become confused when I use computer-based test							
I rarely make errors when using the computer-based test.							
I am rarely frustrated when using computer-based test.							
	1	2	3	4	5	6	7
I dislike the idea of computer-based test in programming exams.							
I have a generally favorable attitude toward using computer-based test in programming exams.							
I believe it is a good idea to use computer-based test in the next programming exams.							

Appendix C: Overview of data analysis

Research questions Or hypotheses	Instrument used	Data	Source	Data analysis
What are factors that affect students' attitudes towards CBT?	Interviews	Text	Students who take computerized programming test	Coding
The mean score of the students who take the CBT will be statistically higher than that of students who take the PBT.	Exam	Exam score graded from 0 to 100.	Exam results.	Two sample case t-test
Students who take the CBT will complete the test in statistically significant less time.	Exam and software	Time in minutes and seconds	Student's exam performance	Two sample case t-test
PEU positively correlates to PU of CBT.	TAM questionnaire	Categorical data scaled from 1 to 7	Students	Spearman's rho
PEU positively correlates to A of CBT.	TAM questionnaire	Categorical data scaled from 1 to 7	Students	Spearman's rho
PU positively correlates to A of CBT.	TAM questionnaire	Categorical data scaled from 1 to 7	Students	Spearman's rho
S positively correlates to PEU of CBT.	TAM questionnaire	Categorical data scaled from 1 to 7	Students	Spearman's rho
T negatively correlates to PEU of CBT.	TAM questionnaire	Categorical data scaled from 1 to 7	Students	Spearman's rho

Appendix D: Coding Scheme

Not using review button	Design of the software	Software
Locations of buttons		
Usefulness of images		
Bad design of the layout		
Less quality pictures		
Ease of adding	Use of IDE	
Ease of focusing		
Spending less time		
Familiarity with Eclipse		
Ease of highlighting		
Ease of writing	Suggestions	Implementation of the test
Missing features of IDE		
Changing the layout		
Changing the locations of button	Testing procedure	
Lack of information		
Lack of blank paper		
Similarities with PBT		
Differences with PBT		

Appendix E: Screenshots of the software

UNIVERSITEIT TWENTE.

Deniz
GÜRSOY

Software Security
Mock Exam

Time remaining
39:52

Review

Finalize

Previous 1 of 7 Next

```
public class Rpcraven{
    public static void main(String argv[]){
        Pmcraven pm1 = new Pmcraven("One");
        pm1.run();
        Pmcraven pm2 = new Pmcraven("Two");
        pm2.run();
    }
}

class Pmcraven extends Thread{
    private String sTname="";
    Pmcraven(String s){
        sTname = s;
    }
    public void run(){
        for(int i =0; i < 2 ; i++){
            try{
                sleep(1000);
            }catch(InterruptedException e){}

            yield();
            System.out.println(sTname);
        }
    }
}
```

Question :

Which statement is true for the code shown in the picture?

- Compile time error, class Rpcraven does not import java.lang.Thread
- One One Two Two
- One Two One Two
- Compilation but no output at runtime

Previous 6 of 7 Next

```
import java.io.*;
import java.net.*;

public class MyServer {
    public static void main(String[] args){
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept();
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String str;
            while(ss.isClosed()!=true){
                str=(String)dis.readUTF();
                System.out.println("message= "+str);
            }
        }catch(Exception e){System.out.println(e);}
    }
}
```

Question :

The class "MyServer" in the code provided creates a server which is reading from an input stream. The class "MyClient" (running on the same computer) is supposed to send messages as long as there is an input from System.in; however, it is not working properly. Make the necessary changes so that client works properly.

Reset

Find

```
1 import java.io.*;
2 import java.net.*;
3 import java.util.Scanner;
4
5 public class MyClient {
6     public static void main(String[] args) {
7         try{
8             Socket s=new Socket("192.168.1.1",6669);
9
10            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
11            Scanner sc = new Scanner(System.in);
12            String input;
13            while (sc.hasNext()){
14                input = sc.next();
15                dout.writeByte(input);
16                dout.wait();
17            }
18
19            dout.close();
20            s.close();
21
22        }catch(Exception e){
23            System.out.println(e);
24        }
25    }
26 }
```

```

public class Movie {

    private String name;
    private int rank;
    private int year;
    private double ranking;

    public void setName(String name){
        this.name=name;
    }
    public String getName(){
        return name;
    }
    public void setRank(int rank){
        this.rank=rank;
    }
    public int getRank(){
        return rank;
    }
    public void setYear(int year){
        this.year=year;
    }
    public int getYear(){
        return year;
    }
    public void setRanking(double ranking){
        this.ranking=ranking;
    }
    public double getRanking(){
        return ranking;
    }
}

```

Question :

The Top 250 films in IMDB (Internet Movie Database) are listed in the file input.txt in the format shown below. In this file, each line contains information about a movie separated with "\".

For example:

Line 1 is : 1\The Shawshank Redemption\1994\9.2

```

Rank of the movie: 1
Name of the movie: The Shawshank Redemption
Released year: 1994
Ranking of the movie: 9.2

```

Complete the code provided so that all the movies are stored in a list and movies that are released in a selected year are printed to the console. Use the class provided to store the movie information.

Reset

Find

```

1 import java.io.BufferedReader;
2 import java.io.FileInputStream;
3 import java.io.InputStream;
4 import java.io.InputStreamReader;
5 import java.util.ArrayList;
6 import java.util.List;
7 import java.util.Scanner;
8
9 public class Program {
10     public static void main(String[] args) {
11         //Create a list that can store Movie object
12
13         try{
14             InputStream ips=new FileInputStream("input.txt");
15             InputStreamReader ipsr=new InputStreamReader(ips);
16             BufferedReader br=new BufferedReader(ipsr);
17             String line;
18             Movie temp;
19             // write the code that read the file line by line
20             while (
21                 ){
22                 String[] splited = line.split("\\\\");
23                 temp= new Movie();
24                 // assign the rank,name,year,ranking that is read to temp object
25                 // and add temp object to list that you created
26
27
28
29
30
31             }
32             br.close();
33         }
34         catch (Exception e){
35             System.out.println(e.toString());
36         }
37         Scanner in = new Scanner(System.in);
38         int selectedYear=in.nextInt();
39
40         // print all the movies in the list that belong to selected year
41
42     }
}

```