Master Thesis

Early Indicator of Success

Department of Computer Science
Professorship of Software Engineering
Chemnitz University of Technology

From

Abhinav Singh Hada

Supervisor: Arooba Aqeel
Chair: Prof. Dr. Janet Siegmund
Module: M.Sc. (ICS)
Datum: 28.09.2021

Abstract

Students when begin learning to program struggle to perform better in introductory courses, even with high motivation to learn the programming skills. This encourages research over the factors that could be the reason for the failures.

Influenced with the study of Alireza Ahadi & Raymond Lister and Donna Teague [7] who devised a test on novice programmers in two universities. Their work included formal instruction in programming which was provided to the students at the starting of the semester. The tests given required the knowledge of only assignment statements in programming, this presented them with the results of two cohorts that had wide variation in the performance. Influenced by the work of Dehnadi and Bornat [4], who design the tests to capture the reasoning strategies, that how the students build different types of models to attempt to solve a programming query. Some could build and some could not, and some are even unable to build a model. But their work presented the findings that the student's model used to understand the working of program execution does not require to be correct model, instead if it has been applied consistently it will still provide better results. Alireza Ahadi & Raymond Lister and Donna Teague [7] research focused on evaluating and interpreting the results based on Neo Piagetian theory which helped in overcoming the disadvantage of earlier work of Dehnadi and Bornat [4] because their focus was more on how novices try to learn to program and, they have made familiar themselves after learning the correct model, which would help the early identification of students who could be in danger of failing the test.

Here we describe the replication of their study to validate the original results. We have performed a study including 293 students at the Technical University of Chemnitz who were learning different programming languages. After providing formal instruction through online studies, we designed a test for students during the starting week and one in the last week of their semester which contained questions of assignment statements. The test is to notice the difference in the performance of students and whether the students have learned the correct model in introductory programming courses to follow. Firstly, we have characterized the results on the basis neo-Piagetian theory to have a glance at how well the students have perceived the correct model, which could indicate their success prediction for the second test. Secondly, we have also observed the participation of students in the pandemic and how the different factors would have affected their performance in the Tests. We took their opinions by the end of the semester and added more findings to our conclusions.

Acknowledgments

I would like to begin by expressing my thanks to my supervisor **Arooba Aqeel** and **Prof. Dr. Janet Siegmund** from "Faculty of Computer Science" at Technical University of Chemnitz whose guidance, support, and encouragement have been invaluable throughout this study. I would also like to thank my colleague **Marc Weiss** for his thoughtful support and recommendations. To conclude I would also like to express my gratitude towards my parents, family, and my friends whose unconditional support throughout my academic career.

Abhinav Singh Hada

Contents

1. Introduction and Overview

1.1 Motivation

The question of what arises in our mind is about the difficulties we face in learning how to program and subsequently gets even tougher to learn to program when proceeded with the topics from introductory courses to the detailed ones and fails to learn the required skills, which could be due to the reason that students might not be taught well at school, so sometimes undergraduates arrive without being familiar with the programming courses some can do well and some cannot.

Despite substantial improvements in the teaching methods and course structures, it is still hard to predict the success rate for students who begin to learn to program.

A study conducted by Dehnadi and Bornat [4] noticed that most of the students fail in the starting introductory courses tests and the failure rate expands all over the world. Through their research they discovered the key factors of failure rate and constructed tests that could help students to achieve success in programming courses regardless of following any kind of model which students have adapted, the only thing matters are that they would be consistent in following the same model. However, they failed to notice the inconsistencies in the mental models where most of the students who followed the wrong model in the starting courses also did well in the end. A different and new mental model also helped the students in the learning process and displayed some promising results.

As the key focus should always stay on learning the correct model and ways to make students adapt it. The disadvantages in the study of Dehandi and Bornat [4] which could not be explained required the inclusion of points such as cognitive skills and which model does the student should process with and how to find he is following the correct model. The test which they devised focused on observing which type of model does the students follow, as that could provide the variety of models which is required to be taken into consideration but was insufficient to explain the inconsistencies in the model. A study conducted by Alireza Ahadi & Raymond Lister and Donna Teague [7] introduced the **Neo-Piagetian theory** which could explain the inconsistencies they observed in Dehnadi and Bornat's research. They primarily focused on whether the students have known the correct model which they require for the introductory programming course. To do so, they observed the advancement of students in the neo-Piagetian stages, which provided better insights in understanding the cognitive development amongst students.

Their study presented a new way of observation, wherein our research our primary focus in observing student performance according to Neo-Piagetian theory, instead of following the wrong model it is important to know the correct model of executing a program and having cognitive development in the same direction.

1.2 Objective

Our main motivation aims at moving on the same path as the study conducted by Alireza Ahadi & Raymond Lister and Donna Teague [7] where we take an online test before and after an acquaintance of introductory courses by the students and focus on staying with neo-Piagetian theory to shed light on the outcomes. In addition, we also devised the test which included a questionnaire about the pandemic, where students could provide their opinions about the difficulties they faced and how their studies were affected and in turn affected their performance.

Research Objective 1: We need to keep track of student performance in the test before and after the semester and observe their transitions in the neo-Piagetian stages, which is required to support our prediction that how acquainting with an understanding of the correct model of program execution and developing the cognitive skills is essential to become a good programmer.

Research Objective 2: The transition in the neo-Piagetian stages of student performance will help us to analyze and come to the conclusion of whether our prediction is correct or not.

2. Related work

Earlier work in this field of research has contributed towards the completion of our work, as researchers rely on the past work and contributions of the theories. We will be explaining the details here about the original work because we believe the previous work modules make themselves as important to describe the outlines and our present work.

Different theories lined up to support the evidence of their research for their respective reasons, which laid down the path and laid some groundwork for everyone to pursue in the direction of finding out the reason for the study. Since the teaching methods are similar earlier research was not enough to support the psychological reasoning, but we cannot exclude them. The work of previous researchers classified the student's misconceptions and mistakes. Adelson and Soloway [1] emphasized the importance of domain knowledge which is crucial to solving problems. J Bonar and E Soloway [2] put evidence to support that prior knowledge of programming in one language makes it difficult for students to program in another. Adamzadeh and others [9] focused on debugging and claimed that if students who are good debuggers are good programmers. Later Johnson-Laird [3] came up with the idea of studying mental models and understanding student's efficiency in deductive reasoning. Dehnadi and Richard Bornat [4] who familiarized themselves with the psychological reasoning provided from previous work, tried a different approach claiming that programming has no relation with age, sex, or educational attainment, nor it show any relations with conventional intelligence and program solving ability tests. They tried to find successful results before the students had any contact with the programming language. Saeed Dehandi, Richard Bornat and Simon [5] who later worked on mental models and tried to find out the regularity in the ways of how these models are used by students and if they stick with the same model would have a better success rate. Raymond Lister [6] came up with the explanation of why students cannot become good programmers by associating his reasons with the neo-Piagetian theory which focused more on abstract forms of reasoning. Alireza Ahadi & Raymond Lister (2013) later came up with another paper to support that prior knowledge produces only a small difference, but the difference is amplified with learning edge momentum. Alireza Ahadi & Raymond Lister and Donna Teague [7] supported their theory based on neo-Piagetian and claimed that developing cognitive skills is an important aspect to become a good programmer.

3. Theory behind

Learning to program is difficult, and many students fail to perform better and face different types of problems while analyzing the programs which drew the attention of researchers and encouraged them to present their perspective after analyzing the situation and find out the reason behind the increase in the failure rates.

The problem with learning programs does not stick with a particular region but it is defined as a widespread problem. There were no reasons or remedies developed to overcome this, but many researchers worked in this field and gave their respective theories. Theories are related to analyzing the mistakes, bugs, and misconceptions usually students have when they start learning to program. Students, in the beginning, are energetic with learning new courses but eventually struggle in the end in understanding the concepts and still pursue them which could be due to family pressure or maybe cannot accept failure. Those who have the innate talent do experience this problem but instead could get frustrated with the slow speed in teaching. It is not the problem with the teaching method or with a highly experienced teacher who imparts their knowledge to the students, but they still fail to recognize which method we should opt for so, students could get better results when they start the introductory courses. So many experiments have been performed in this direction, expecting better results but the problem persists and every year when the students enroll for the same course, and teachers struggle with the same problem.

The first method which is taken into consideration is to interview the students and ask about their suggestions and opinions regarding the teaching methods. Another aspect which could be discussed is about the teaching material, is it helping the novices, in the beginning, to learn to program. According to previous findings, some have achieved some success in this direction, and some have not. They also tried to interview the students' parents where they asked about having a computer at home, but generally, the common response was that they just use the computer to play games on it.

3.1 Different areas of study

3.1.1 Domain Experience

In the beginning, there was not so much related work before Dehnadi and Bornat when they presented their paper. Everyone came with different to support their theories. Adelson and Soloway [1] emphasized the role of domain experience when they could design an experiment. They presented that when a software designer working on designing software in the familiar domain, he already possesses the knowledge and skills required to design the software, unlike the person who is unfamiliar with the domain, who have no prior knowledge to work at this place. They observed and presented their findings by taking into consideration, observation of expert designers and compared novice and expert's behavior towards designing a software.

For their analysis, they proposed an experiment where they could record the process that how the experienced designer who was familiar with the domain attempted the query as compared to the beginner who was unfamiliar with the domain. They videotaped all the processes to observe the words, actions, gestures, and facial expressions too because they were asked to think aloud. The

experimenter stayed along with the designer who recorded the answer for the question "What are you thinking now?" these were some of the queries which helped them identify the required process which is important in the design process, and it was possible only by observing the experienced programmers and what is their approach.

The experiment design included the participation of both expert and novice designers. The expert had at least 8 years of experience whereas the novice had less than 2 years of experience and both were qualified in designing the communication systems. Their experiment presented the formation of mental models, their expansion, a simulation that is required to increase the speed of the design in progress. Note making, representing constraints, simulation, systematic expansion, retrieving labels are the research tools that a designer uses the aid in design software. Their conclusion that the experience with all the elements in designing requires a tool to retrieve their previous design and assemble the elements in the last phase of design, whereas the designer with no experience in the domain and is unfamiliar with it will require tools for help to infer the constrains of the design.

3.1.2 Tracing Skills

Lister and others [8] in their paper who performed an experiment over a large amount of population in seven universities in the United States and other countries presented their theory. The authors presented a test to all the students which consisted of programming problems. In their results revealed that the majority of students failed to attempt the problem correctly. Their poor performance is the result of a lack of ability to solve the problems.

To present the explanation to their findings they categorized the problem-solving method into 5 categories: 1. Abstract the problem from its description 2. Generate sub-problems. 3. Transform sub-problems into sub solutions. 4. Recompose and 5. Evaluate and iterate. These were the five steps methodology that should be adopted by the novices when they attempt to solve any programming query.

They concluded when they found out the poor performance of the students who fail at the introductory courses. Students who could not follow these steps in problem-solving did fail, also many students lack knowledge and skills which was the required approach to solve the problem and many even failed to even understand even the small piece of code. The students who may have performed well in the test but if asked to write the code with similar complexity and if fails are also likely suffering from the problem-solving skills.

3.1.3 Debugging

Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins research [9] focused on gaining insights into the mistakes especially the debugging methods which students used. The compile errors which were generated from the programs which the students wrote were observed when their programs were debugged, even the ones who had a good understanding of programming do not have proper skills to debug effectively. The focus should be emphasized more upon learning skills to how to debug a program and on teachings the skills to do that.

The programming language was taught to students, and the course consisted of fundamental language constructs, loops, methods, and strings. The programming exercises were provided to

students the data which was collected and fed to the compiler and all the error message database was collected. There were two separate tables created. One contained compiling error messages and if the program was free from all the other errors, the second table contained that.

It has been found through their study that good debuggers are good programmers. It was considered one of the most important skills which need to be taught in the courses, further a programmer must improve the ability to read other person code and should know how to debug it because it would enhance his competence as a skilled programmer.

3.1.4 Spatial Visualization

Programming is no doubt is difficult, but the problem should not be analyzed from only one dimension and other factors must be considered. Simon, S. Fincher, A. Robins, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer,M. Hamilton, R. Lister, M. Petre, K. Sutton, D. Tolhurst, and J. Tutty. [8] included the study in the other dimension where cognitive, attitudinal, and behavioral factors were taken into consideration such as Spatial visualization and reasoning. The ability to understand the programming query, design the task, and attitudes towards studying. The tests conducted suggested that the deep approach in this direction is related to success which means that every query must be done qualitatively.

The cognitive approach included a standard paper folding test, map sketching, searching a phone book, and a study questionnaire. In the means of assessing, the aptitude problems were provided to the group of students where the time was a key factor, in the given time how many questions does the students attempts correctly. The second test included a test to draw a map between two destinations and observations included which pathways do the students chose to accomplish this. The third test required participants to search a phone book where they need to find the required name and, in the end, explain the process which they used to find out that name. The fourth test included a questionnaire which is about to know the approach of students depending upon the factors (motivation, available time, personal perception).

They concluded that the deep approach is directly linked with the marks they scored. Students who engaged in solving programmed deeply with the material are more likely to succeed because they were successful and can design and sketch a map to solve the problem.

3.1.5 Test for Aptitude

Computer scientists have searched and performed many tests to determine the difficulties which the students face while learning to program. Before Dehandi and Bornat [4] nobody devised tests that could predict the success rate of students before attempting for exams. They focused on three major hurdles in learning programming which are assignment and sequence, iteration, and concurrency.

They intended to observe the mental models when students implement the approach to solve the programming queries, where students showed a wide variety of models and students were found facing confusion during exam time.

The results which they found revealed the mistakes which students face during the beginning of the course and if they try to work on these three hurdles, they could get success in introductory

courses. Even after observing the mental models which speculated the reasons for success in their research, could not explain how it works.

Even with the limitations, it provided a new ground to work upon, they proposed another research [(Mental Model, consistency, and programming attitude)] which was the extension of the previous one. Prediction of the success rate depends upon how the students encounter the problem and the ways they approach it.

They proposed the same tests as in the previous research which focused on recursion and assignment problems. They observed the variety of models and sequences which students followed while answering the questions. Questions were the sample of the java program which consisted of two or three variable-to-variable assignments. Since the students were taught in the beginning with the introductory courses the mental models were designed accordingly (Figure 3.1).

1. Read the following statements and tick the box next to the correct answer in the next column.	The new values of a and b are:	Use this column for your rough notes please

`int a = 10;`
`int b = 20;`
 	
`a = b;`
	☐ a = 10 b = 10
	☐ a = 30 b = 20
	☐ a = 0 b = 10
	☐ a = 20 b = 20
	☐ a = 0 b = 30
	☐ a = 10 b = 20
	☐ a = 20 b = 10
	☐ a = 20 b = 0
	☐ a = 10 b = 30
	☐ a = 30 b = 0
	
	Any other values for a and b:
	
	a = b =
	a = b =
	a = b =	

M1. Value moves from right to left (a←b and b←0 – eighth line in figure 1).

M2. Value copied from right to left (a←b – fourth line of figure 1, and the 'correct' answer in Java).

M3. Value moves from left to right (b←a and a←0 – third line of figure 1).

M4. Value copied from left to right (b←a – first line of figure 1, a reversed version of the 'correct' answer).

M5. Right-hand value added to left (a←a+b – second line of figure 1).

M6. Right-hand value extracted and added to left (a←a+b and b←0 – tenth line of figure 1).

M7. Left-hand value added to right (b←a+b – ninth line of figure 1).

M8. Left-hand value extracted and added to right (b←a+b and a←0 – fifth line of figure 1).

M9. Nothing happens (sixth line of figure 1).

M10. A test of equality (first and fourth lines of figure 1).

M11. Variables swap values (seventh line in figure 1).

S1. [sequence] The first assignment has its effect with initial values, then the second with the values produced by the first. (One effect is reported; the corresponding box is ticked.)

S2. [simultaneous, multiple] Each assignment takes effect using the initial values of variables. (All effects are reported; the boxes corresponding to each effect are ticked.)

S3. [simultaneous, single] Each assignment takes effect using the initial values of variables, but only the effects on the destination side are reported. (One overall effect is reported; the corresponding box is ticked.)

Figure 3.1: Anticipated mental models.

After observing the results generated from their tests, they observed that students were consistent in following any of the particular models mentioned in Figure 3.1 in executing a program. Their work introduced the phenomenon of consistency, which required further study in this direction, that is why students tend to follow only a particular model. The tests categorized two populations into subgroups one of the subgroups was successfully able to build a mental model and used it to apply consistently, these students showed promising results and predicted that their performance will be much better than the other subgroup. Many students had programming experience from before the beginning of the programming courses, but it did not show any stronger effect on results, as the important factor which needs to be studied is consistency.

3.1.6 Interactions

For many researchers, the key area of research focuses on the model which is adapted by the students and what approach they use to solve programming problems. Syntactic/semantic interactions in programmer behavior: A model and experimental results [10] presented that a cognitive framework could describe the behaviors of programmers in program composition, comprehension, debugging, and modification. Their study included both programmers and non-programmers to study the behavior that how they approach a question.

They described the cognitive model into two categories:

Cognitive Structures: These programmers possess in their memory or may possess in the future.

Cognitive Processes: Involves adding the knowledge and skills.

To understand the cognitive behavior of a programmer every programming approach must be analyzed from writing a program, searching for errors, understanding, modifying, and at last applying new programming skills and knowledge to provide a solution to the problem.

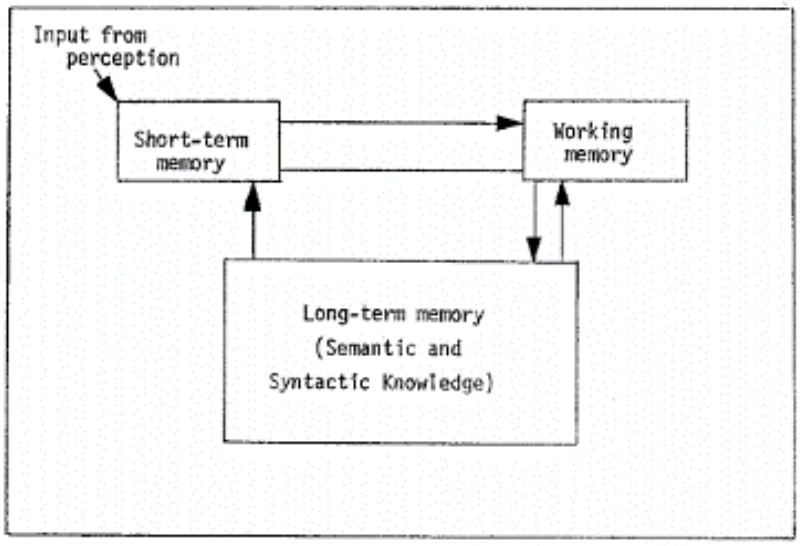


Figure 3.2: Problem-solving components.

Information is shared in a certain behavior which involves short-term memory, working memory, and long-term memory. Short-term memory with a very limited capacity, Working memory with more permanent than short-term memory but less permanent than long-term memory, and Long term memory with unlimited capacity (programmer's permanent knowledge).

The framework represented in Figure 3.2 has been evolved through the information processing approach through the psychology of learning, memory, and problem-solving. These components are important for discussion because it is important to analyze to what information does the programmers pay attention to for example reading descriptions of the problem. After getting acquainted with a new problem the first interaction occurs between short term memory and long-term memory, where new information is generated from in short term and the concepts embedded in the long term and merged with the working memory which is used to generate the result which is later stored in long term memory.

A cognitive model of a programmer's behavior separates the syntactic knowledge from the semantic knowledge. Semantic knowledge consists of programming concepts that are not related to a specific programming language, but they are important sets of information which is developed through programming experience. On the other hand, syntactic knowledge is detailed and much more precise. It involves details for example that how the iteration happens in a loop, conditional or assignment statements occur. Both semantic and syntactic are part of long-term memory where semantic knowledge is updated through meaningful learnings and problem-solving where every time new concepts are added to it whereas syntactic is not integrated whenever new information is added it interferes with the previous one.

Problem-solving and programming are two different aspects, and they should be looked upon from a cognitive process point of view. This would help to improve the syntactic structure of programming which is closely related to semantic structure and therefore making it convenient for the programming learning process.

3.1.7 Neo-Piagetian Theory

Neo-Piagetian theory of cognitive development appeared to support the areas which were presented by Jean Piaget's seminal theory of intellectual development addressed the criticisms against the theory (1) the unit of cognitive analysis is the scheme or psychological structure; (2) psychological structures undergo qualitative transformation over time; and (3) higher-order structures develop through the differentiation and coordination of lower-level structures. [12]

Concrete and Other Neo-Piagetian Forms of Reasoning in the Novice Programmer research brings [6] introduction of Neo Piagetian explanation to the test which was conducted in previous years which completely focused on predicting the programming ability in students. They presented the zone of relationship, which was never been explored, by digging more into the concepts such as conservation and reversibility.

Unlike the earlier theory, they explored the link between the Neo Piagetian theory and programming. They try to re-evaluate programming from the Neo-Piagetian perspective. This described formal operational reasoning, pre-operational reasoning, and concrete operational reasoning which are the abstract types of Piagetian reasoning, these are mostly found in literature,

which could connect with classic Neo Piagetian theory. The reasoning levels and their link with the programming were never discussed before, which motivated them to dig more into this relationship.

Neo Piagetian theory aspects explain that people progress regardless of their age and possess expert reasoning as they get expertise in a specific problem-solving domain. Also, it presents evidence to support that abstract reasoning is not gained through biological maturity but due to the increase in effective capacity.

The study concluded that when students were presented with different programming codes to solve, they exhibited three forms of reasonings. While all the lectures were taken in the terms of formal reasoning. Some of the students were in the preoperational stage where they could not answer or explain the abstraction of codes such as how the value inside the code is changing. While students who were in the concrete operational stage, could successfully see the abstraction of codes, but most students who are at these levels cannot be promised to reach the formal operational level simply by exposing them to learning materials. If a student does not possess the programming skills and manifests formal operational reasoning, that does not mean he cannot be good at programming if given the right tuition. This brings the pedagogical approach of Neo Piagetian theory that determines that the phases such as concrete operational and per operational helps in increasing the sophistication that how the students learn to answer the programming codes.

3.1.8 Consistency

In the original work Falling Behind Early and Staying Behind When Learning to Program by Alireza Ahadi & Raymond Lister and Donna Teague [7] which is inspired with the earlier work by Raymond Lister Concrete and Other Neo-Piagetian Forms of Reasoning in the Novice Programmer [6] relating to the skills of programming with Neo Piagetian study where they devised a test which consisted of questions to test the students for their tracking values in an intermediate variable, deductive and inductive reasoning, and assignments questions. Their work was also inspired by Dehandi and Bornat: Mental models, consistency, and programming aptitude [5] where they study different types of mental models which are opted by the students when they approach the programming questions. The disadvantage of their work [5] as they failed to explain the results of the students who were inconsistent in following one type of model how their results were better by the end of the tests.

The grading of the tests of their research of the students was later related to three stages of Neo Piagetian theory to contribute towards their research. Using the same scenario and goals equipped with similar tests we also devised the same test for the students and tried to analyze the scores and categories the students into the three-stage model of the neo-Piagetian theory.

They did not disapprove that programming cannot be an innate talent but instead, they focused on the principle that cognitive skills are learned and not innate. Perhaps a student who follows a correct model of how to approach a question correctly would have the cognitive advantage and gain the consistency of reasoning.

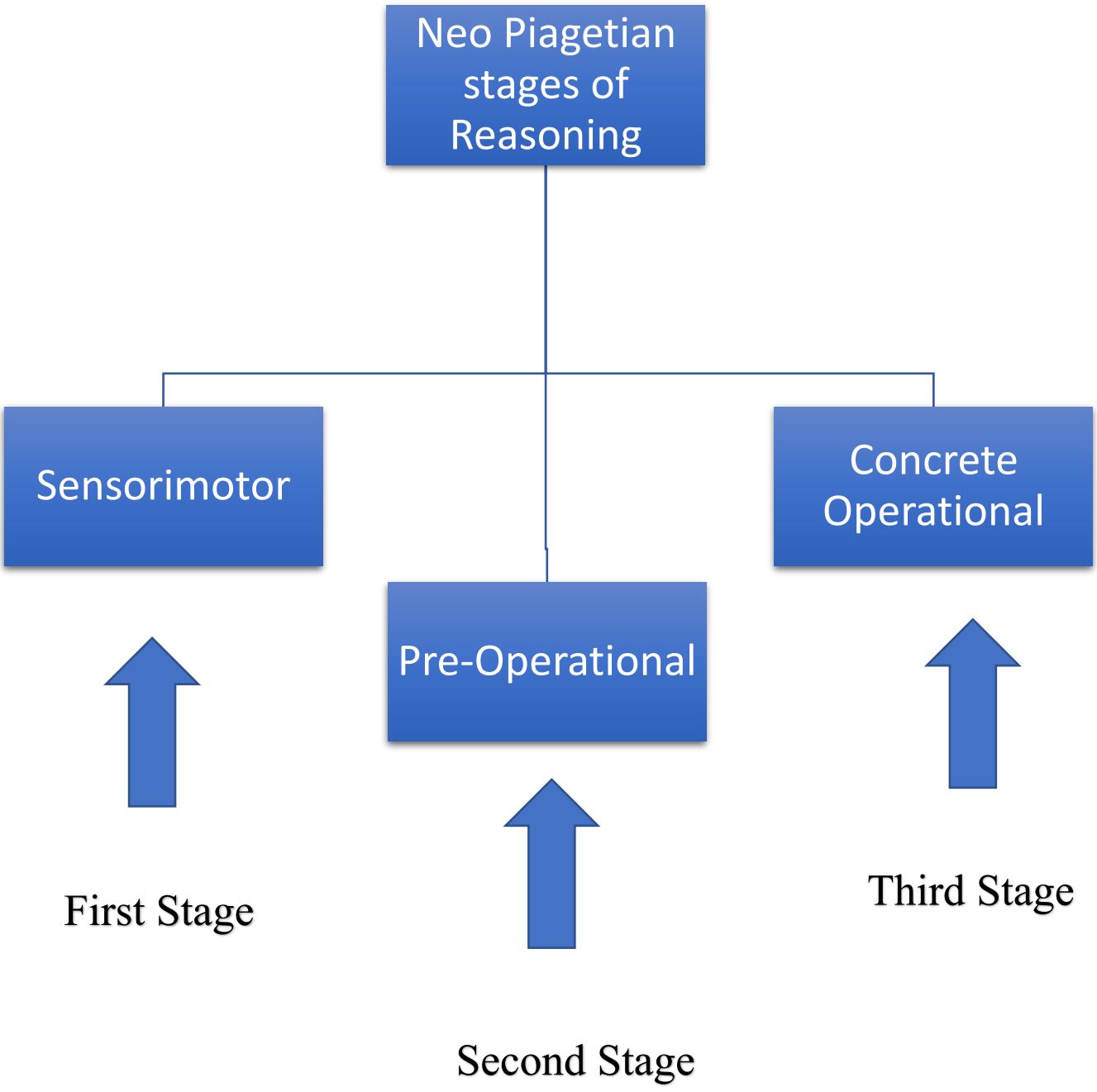


Figure 3.3: *Neo Piagetian- Three stages of Reasoning*

4. Methodology

4.1 Work Description

To achieve the desired research objective, we conducted tests to be given at the beginning and the end of the semester for the students. Regardless, of their choices in the programming languages, we developed tests that are mainly concerned with assignment statements on integer variables and sequential statements. It focused on observing the choices of answers which students put together and help us to identify the method which they adapted to answer the questionnaire.

The design of the test structure is influenced by the work of Alireza Ahadi, Raymond Lister, and Donna Teague [7] by applying the neo-Piagetian theory after receiving the results, where we place the students into the three-stage model (Figure.3.3) from early stages of learning to program:

Sensorimotor: Programmer who has adapted the incorrect model of analyzing the program execution.

Preoperational: The programmer tries to execute the code manually and perhaps the answers to his selection are based upon the assumption on the pieces of codes. Furthermore, examines the relationship between the input and output.

Concrete Operational: The programmer is familiar with the approach to adapt and which model to follow, rather than using the preoperational approach. The first stage is where students are anticipated to manifest a determined approach to writing code.

This three-stage model (Figure. 3.3) will contribute to our interpretation towards our research that whether learning the correct model and enhancing cognitive skills will allow the students to learn programming skills.

4.1.1 Experimental Tests

We designed the first and second questions with a focus on monitoring the understanding of the students about the assignment statements, how the value is moved from right to left, which later overwrites the previous values.

The questions are as follows:

Assignment Problems

Q1. In the boxes, write the values in the variables after the following code has been executed:

a = 1
b = 2
a = 3

Sol. The value of a ☐ and the value of b is. ☐

Q2. In the boxes, write the values in the variables after the following code has been executed:

r = 2
s = 4
r = s

Sol. The value of a ☐ and the value of b is. ☐

Influenced by the work of Dehnadi and Bornat who used similar questions for their test. The box will contain only one value, could not consist of more than one value. Students who face complications and have difficulty with the first two questions lie in the category of *Sensorimotor* stage.

Sequence Statements:

Q3. In the boxes, write the values in the variables after the following code has been executed:

p = 1
q = 8
q = p
p = q

Sol. The value of p is ☐ and the value of q is. ☐

Some of the students who were not able to decode the values of p and q and interpret this as swapping values query are at the *Late Sensorimotor* and *Early Preoperational Stage*.

Changing of Intermediate values:

We designed these two questions with a focus on monitoring, whether students could track the changes in the values in between variables.

Q4. In the boxes, write the values in the variables after the following code has been executed:

x = 7
y = 5
z = 3
x = y
z = x

Sol. The value in x [ ] is y is [ ] and z is. [ ]

Q5. In the boxes, write the values in the variables after the following code has been executed:

x = 7
y = 5
z = 0
z = x
x = y
y = z

Sol. The value in x is [ ] y is [ ] and z is. [ ]

The students who successfully attempted question 1,2,3 still could some difficulties in tracking the intermediate values between the variables where they could mistake in analyzing the sequence of execution in the above question. Normally, the processing of executing the code is held in the working memory (Figure. 3.2) where the students try to store the values of the variables and maybe later face difficulties in recording the values and present themselves with an error.

Students who successfully attempted these questions lie in the mid-stage of the preoperational stage of the neo-Piagetian model.

Q6. In the code above, what do you observe about the final values in x and y? Write your observation (in one sentence) in the box below.

Based on question 6 here we try to test the observation of students whether they could identify and reveal the reasoning about the code, that how their value is exchanged or swapped between variables, and these students thus work at the preoperational stage.

Q7. Three lines of code, for any set of possible initial integer values, are stored in those variables.
Assume that variables i, j, and k have been declared and initialized.

j = i
i = k
k = j

Sol. Sample answer swaps the values in variables i and k.

This question is to test if students could identify and uncover the role of this code.

Q8. Write code to swap the values stored in first and second.

Sol. Sample Answer

temp = first

first = second
second = temp

In the above questions, many students could get these assignments statements wrong order too and express their answers from back to front. This could lead to difficulty in understanding the execution of a program that would work sequentially, this would be because of simultaneous statements and every result is dependent on the last result, which in turn could create confusion that which variable would store which value. According to the neo-Piagetian model, the students who will struggle in solving these questions are at the stage of preoperational or lower and the students who were successful in answering the Q7 and Q8 are at the stage of concrete operational.

4.1.2 Online Test

We presented our test by using the survey model where we could receive data and analyze the choices of answers provided by students, and the responses can be generated in the form of CSV and other formats. The individual lines correspond to the details of the students. The students were taught through online lectures commencing from the starting week assuming that every student participated in it. For our experiment, we used the online survey tool SoSci survey where you could reliably implement your survey. In the beginning, before participating in the test, students just need to enter basic information which includes their unique id, course, etc. Through the unique id, it was easier to work on the data after the test was over.

Sample questions in the above section (Experimental Tests) are implemented similarly, where the students must enter their answers in the boxes and proceed with the next page for the further questions. On the other hand, the general introductory questions to ask about the experiences consisted of tick in a box or several boxes, which would be useful for our analysis.

A sample question from the survey is provided in Figure. 4.1:

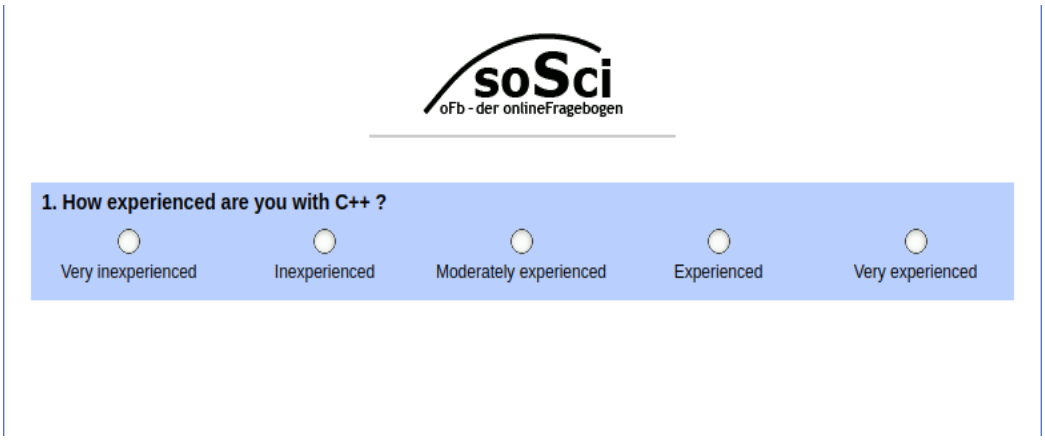


Figure 4.1: Sample Question is given in the survey.

The questionnaire (Table 4.1) which could be implemented effectively with the use of this tool and the tick responses provided by students are converted into numbers in whichever format it is downloaded from the survey tool. Similarly, the responses in other questions too were also converted to numbers, because in our script that would be beneficial to analyze the choices.

Tick Responses	Numbers
Very Inexperienced	1
Inexperienced	4
Moderately Experienced	2
Experienced	5
Very Experienced	3
Not Answered	-9

Table 4.1: In this way, we would see the tick responses in our datasheet.

4.2 Conducting Test 1

All the tutorials beginning from the introductory sessions are presented to the students through online lectures, where we believe the lectures are attended by every student who enrolled themselves for this online test. There was a requirement for the elimination of some data of the students. We eliminated the data where the students just filled the answers with some random values to complete the test quickly, there were a few other columns also which were present in our datasheet and we had to evaluate it afterward for example the time column, the browser window, operating system, etc. We had to eliminate all these columns since there was no requirement of this data in conducting our study and in the part of our analysis.

All the questions were presented to the students, in the same way, shown in the section (Experimental Test). Each question holds a score value of 1. So, the maximum score which could be achieved by the individual would be 8. Questions from 1 to 5 were easy to give markings upon but in questions 6,7 and 8 answer boxes, students must express the sequential nature of the statement and how the value has been assigned to the variables. We received different types of possible ways in which these questions could be answered, and to evaluate we had to find a solution and reward them 1 mark for each correct question. To make this accurate, we made a list of mandatory words such as swapping, interchange, etc. which must be present in the answer provided, because this could signify that the student has successfully understood the purpose of these questions and understood the correct model for executing these statements.

The second test, which was conducted by the end of the semester was given to the students using the same online tool and projected the questions which were more related to sharing their experience on pandemic and were they comfortable in taking all the online lectures and assignments and is the home environment was completely suitable for them in the studies. If not, then how does it affect their performance in the semester. This was the additional questionnaire which we devised thought would add some essential findings in our study, because in taking a test which requires the active participation of the student, so to consider all the possible other factors would be a wise choice to make.

4.2.1 Results and Grading

As in the case of grading, assigning the score for the correct question is the simple part but the important question is what that score means. There were noticeable differences in the performance of students where the **Total Scores** vary from 1 to 8, as we neglected the students who scored 0 in the tests and some students who put some random values, random alphabets, and the errors in the responses due to failure of the internet or any other issue with the connection which prevented them to enter correct results. Questions 1 to 5 require answers as individual values in the boxes of the survey questions which we received at the end of the test, whereas Questions 6,7, and 8 had to write manually their answer to represent the function and their objective.

For the analysis, we made use of **Jupyter Notebook** where the language **Python** with **panda's** package recognized was more convenient for reading our data and creating new essential columns

for better analysis. As each row represents the data of every student, where columns (S101_01 - S108_01) contain the answers to the questions (Q1 to Q8).

For the last three questions since it was difficult to analyze according to the interpreting the language with our script since the test was conducted in both English as well as the German language. So, there are a variety of ways that the answer could have been given. Reading the response through the script was time-consuming since we had to find a way to include every possible keyword to the dictionary from which we could analyze whether the answer provided by the student match with the keywords or not. But even with taking keywords into account, some students expressed their answers which could be awarded 1 point, because the meaning seems to be right but if you try to match with the keywords, we get a wrong result. So, to avoid any errors in marking we had to evaluate the choices manually for questions 6,7, and 8 and grade them accordingly. So, to analyze the datasheet we made a separate column for the last three questions. If the question is correct and the student has successfully conveyed his result we gave it 1 point on the side of it, where 1 means True and 2 means False.

PD14_01 (EIOS1/EIOS2)	EIOS1_ S101_ 01	EIOS1_ S101_ 02	EIOS1_ S102_ 01	EIOS1_ S102_ 02	EIOS1_ S103_ 01	EIOS1_ S103 _02	EIOS1_ S104_ 01	EIOS1_ S104_ 02	EIOS1_ S104_ 03	EIOS1_ S105_ 01	EIOS1_ S105_ 02	EIOS1_ S105_ 03	EIOS1_ Total Score
1mamao	3	2	4	4	1	1	5	5	5	5	7	7	7
4hSSaa	3	2	4	4	1	1	5	5	5	5	7	7	8
2eRHam	3	2	4	4	1	1	5	5	5	5	0	7	5
23MMSCM	3	2	4	4	1	1	5	5	5	5	7	7	8
mainh1	3	2	4	4	1	1	5	3	5	5	7	7	6

EIOS1_S106_01	EIOS1_S107_01	EIOS1_S108_01
Their values are swapped using a temporal variable z	values of i and k are swapped using intermediate variable j	temp = 0temp = firstfirst = thirdthird = temp
swapping of 2 numbers	swapping of numbers using a temporary variable	temp=firstfirst=secondsecond=temp
I don't get your point	j=k=i	X= firstfirst = secondsecond = X
above code, we conduct to swap value into one variable to another variable.	Above code, swapping technique is used, the act of swapping two variables refers to mutually exchanging the values of the variables	first= first+secondsecond= first-secondfirst= first-second
x gets the value of y which is: 5 and y get the value of z which: 7 because the initial value of z is 0 but when code executes the value of z is replaced by the value of x.	Sawp of the values of i and j	temp = first; // Value of second is assigned to first first = second; // Value of temp (initial value of first) is assigned to second second = temp;

Table 4.2(Page. 25 & 26): Data (Sample) which we received from the Test 1 conducted at the starting of the semester. It contains Unique Id, S106_01 - S108_01 (Ques 1 to 8) answers.

Test 1 scores	N=293 (Number of Students)	Q1%	Q2%	Q3%	Q4%	Q5%	Q6%	Q7%	Q8%
1	5	60	20	20	0	0	0	0	0
2	8	62	75	12	12	12	12	12	0
3	10	50	70	50	30	20	20	30	30
4	19	73	84	63	63	26	26	10	52
5	31	77	87	93	67	67	9	41	54
6	58	82	91	98	91	94	36	27	77
7	85	90	98	96	94	96	78	58	85
8	63	100	100	100	100	100	100	100	100

Table 4.3: Percentage of students who answered the individual questions correctly.

In the test1 out of N = 293 students, 63(22%) students were able to achieve a full score (Figure 4.2 & 4.3), which means they were able to enter every answer correctly in each part of the questions.

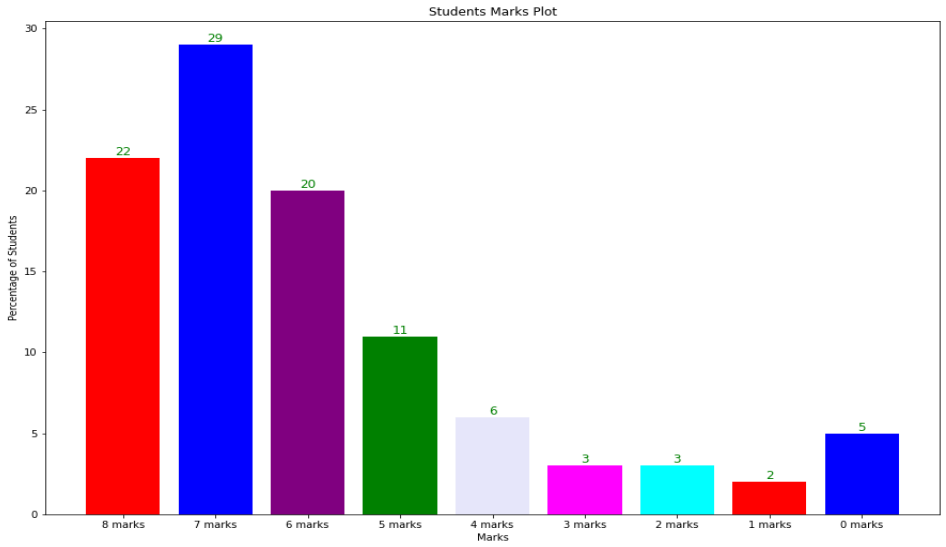


Figure 4.2: Percentage of students who scored marks in Test 1.

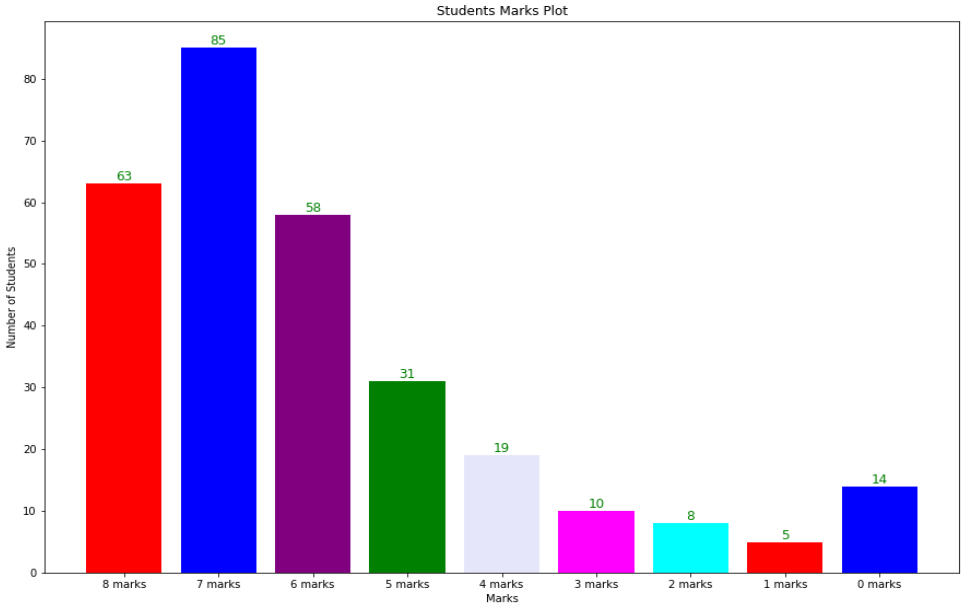


Figure 4.3: Number of Students who scored marks in Test 1.

Test 1 Scores	N (Number of Students)	Q1(a) %	Q1(b) %	Q2(a) %	Q2(b) %	Q3(a) %	Q3(b) %	Q4(a) %	Q4(b) %	Q4(c) %	Q5(a) %	Q5(b) %	Q5(c) %	Q6 %	Q7 %	Q8 %
1	5	60	80	20	60	60	60	80	0	0	60	0	60	0	0	0
2	8	62	100	75	100	37	50	62	12	12	50	12	62	12	12	0
3	10	50	100	70	100	60	60	80	40	30	70	30	60	20	30	30
4	19	78	89	84	89	68	78	94	63	63	68	36	68	26	10	52
5	31	77	100	87	100	93	96	93	70	67	90	67	87	9	41	54
6	58	86	96	91	100	100	98	98	91	94	98	96	98	36	27	77
7	85	90	98	98	100	97	97	97	95	96	97	96	98	78	58	85
8	63	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 4.4: The percentage of students who answered each question correctly in accordance with the total score.

4.2.2 Analysis

4.2.2.1 Semantics of Assignments statements (Q1 and Q2)

Table 4.3 displays the overall performance from Q1 to Q8, where questions 1 and 2 focused only on analyzing the choices made by students in assigning values in a variable. The scores would determine that how many percentages of students understood, how the value is assigned to a variable and which statement result will be displayed in the output when we have multiple lines of statements.

Of the students Table 4.4 who scored 1 mark, about 60% of students answered correctly what would be the value of variable **a** in Q1and 80% answered correctly what would be the value of variable **b** in Q1. Similarly, in Q2 where 20% of students' answers were correct in placing the value for **r,** and about 60% whose answers were right when entered the value of **s.** Significantly, as the students who scored marks increased the percentage of getting correct answers for Q1 and Q2 also rose higher. As the students who got 4 marks answered Q1(a and b) (78% and 89%) and Q2 (r and s) (84% and 89%), the ones who scored 6 marks were (86% and 96%) and about 63 students did well who got both questions correct.

Both the questions primarily focused on assigning values to a variable but Q1 had two statements, unlike Q2 which had 3 statements. The percentage of the correct answer of Q2 is lower as compared to question 1. The value of the second variable asked in the questions was mostly correct and students were able to predict the assignment, but the changes were observed when they had to assign the value to the first variable. In both cases, we tried to find out whether the students could identify that the execution of the third statement will provide them the final answer. Almost in every case, they were successful in reaching this conclusion, which means they have understood which model they should follow in analyzing the multiple variable assignment problems.

Test 1 Scores	N (Number of Students)	Q1(a) %	Q1(b) %	Q2(a) %	Q2(b) %
1	5	60	80	20	60
2	8	62	100	75	100
3	10	50	100	70	100
4	19	78	89	84	89
5	31	77	100	87	100
6	58	86	96	91	100
7	85	90	98	98	100
8	63	100	100	100	100

Table 4.5: Percentage of students who answered Q1(a, b) and Q2(a, b) correctly.

4.2.2.2 Effect of Sequence of statements (Q3)

After observing the responses of programmers in assignment questions, then we are required to see the responses in a similar assignment problem and it is followed with the sequential statements, where one must remember the previous value stored in one variable and how the same value has been transferred to another variable or could be interchanged later.

Dehandi and Bornat also used a similar question, where students could interpret this question as swapping values which they even named as 'Intentionality bug' because some students could not apply the sequence properly in solving these kinds of problems.

According to our result, the students performed well which signifies that the approach which was used by students was correct in solving Q3. Amongst all students, those who scored 2 marks could not achieve a 50% result, which means they could apply the sequential execution of the statement or maybe could not follow the structure of execution. About 37% of students answered the question, correctly which means they successfully entered the correct value of **p** and about 50% entered the value of **q** correctly and in total, out of 19 students 63% entered the correct choices in the boxes.

Test 1 Scores	N (Number of Students)	Q3(a) %	Q3(b) %
1	5	60	60
2	8	37	50
3	10	60	60
4	19	68	78
5	31	93	96
6	58	100	98
7	85	97	97
8	63	100	100

Table 4.6: Percentage of students who answered Q3 (a, b) correctly.

4.2.2.3 Tracking Intermediate Values (Q4 and Q5)

After assignment statements, the next step is to observe how well students understand in tracking intermediate values in the multiple-line statements (Q4 and Q5) because even after understanding how the assigning values to variable works, students could make errors and cannot track the values in a variable in the series of assignment statements. The possible reasons would be, that the students try to memorize the value rather than writing down on the piece paper, where the chances of making mistakes in tracking values are slightly higher, and some use the error-prone method of recording and tracking every value on the piece of paper before providing the final solution.

The students whose total score was more than 5 could certainly track values moving in between statements. These students who scored 6 or higher are in the preoperational or higher neo-Piagetian stage.

Test 1 Scores	N (Number of Students)	Q4(a) %	Q4(b) %	Q4(c) %	Q5(a) %	Q5(b) %	Q5(c) %
1	5	80	0	0	60	0	60
2	8	62	12	12	50	12	62
3	10	80	40	30	70	30	60
4	19	94	63	63	68	36	68
5	31	93	70	67	90	67	87
6	58	98	91	94	98	96	98
7	85	97	95	96	97	96	98
8	63	100	100	100	100	100	100

Table 4.7: Percentage of students who answered Q4 (a, b, c) and Q5(a, b, c) correctly.

4.2.2.4 Inductive Reasoning (Q6)

As discussed before, that Q4 and Q5 were designed to test the tracking of execution of sequential statements. But students who all attempted these questions, many of them got it correct. To analyze that did they understood how the code worked, we designed Q6 where we asked them to provide the reason and explain the behavior of statements based on their observation, that how the values swapped in between different variables.

This question also included the responses in the form of guesses, where the students just entered what they assumed could be right. Students who were not able to get Q4 and Q5 correct are least expected that they could have answered Q6 correctly. There is a huge percentage difference amongst students who did well in Q4 and Q5, but it showed mixed results when we compare the scores with Q6.

There is no steep increase observed in the percentage of students who answered correctly with respect to their total score. Students who scored 5 or less than 5 could not perform well, on average the scores were all less than 30%. There was a slightly higher difference between the scores of (Q4, Q5) and Q6. So, it is apparent that the students who scored more than 6 were able to perform much better on this question. As a result, it shows these students know the sequence of execution of statements and knew the reasoning that how the code worked and how the swapping of the values was carried in this question.

Test 1 Scores	N (Number of Students)	Q6 %
1	5	0
2	8	12
3	10	20
4	19	26
5	31	9
6	58	36
7	85	78
8	63	100

Table 4.8: Percentage of students who answered Q6 correctly.

4.2.2.5 Deductive Reasoning (Q7 and Q8)

Both Q7 and Q8 contained the three lines of codes were the variables we assumed to contain any sort of integer values that have been declared or initialized, where the students now had to explain their answer box, that how the statements would get executed, and how the values would be swapped in between variables.

There is a statistically huge difference in the performance of students when it came to attempting Q7 and Q8 only the students who scored 8 marks were able to present the better result as compared to others. The ones who scored 7 did show a good performance in Q8 where 85% successfully received full points in it.

Test 1 Scores	N (Number of Students)	Q7 %	Q8 %
1	5	0	0
2	8	12	0
3	10	30	30
4	19	10	52
5	31	41	54
6	58	27	77
7	85	58	85
8	63	100	100

Table 4.9: Percentage of students who answered Q7 and Q8 correctly.

4.2.2.6 Programming Experiences vs Scores

In this section, we could present the data, which is related to programming experience, which means how many students were successful in scoring good marks if they possess experience in languages. Since we ask the students about their experiences (Appendix A1) to share with us, related to programming experience which could also be a possible aspect that invited more discussion in this area.

Through analyzing all the scores, we made graphs to present our findings which are related to their experience.

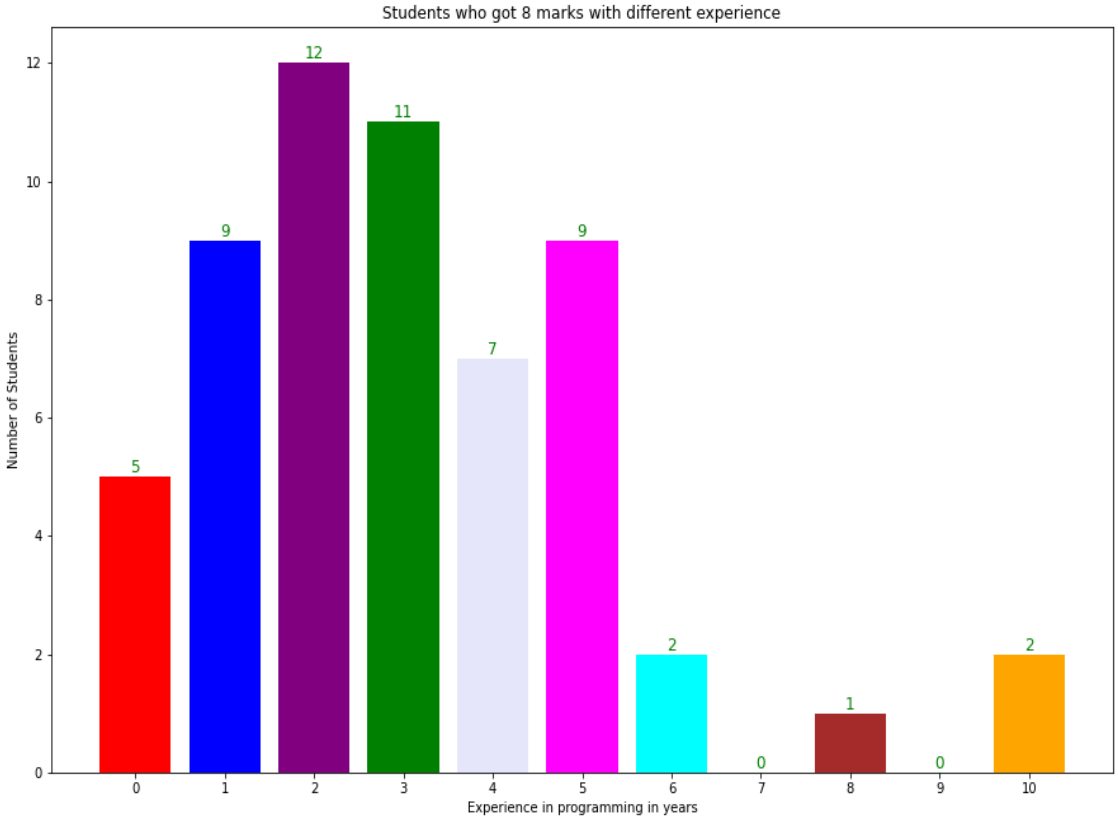


Figure 4.4: Students who scored 8 marks with different experiences in programming.

In this first Figure 4.4 it represents scores of students who scored 8 marks and we took responses of experience levels in years from 0 to 10. Assuming that it may give more insights on students' performance, and it did shed some light in this area.

In Figure 4.5 it represents the scores of students who achieved 7 marks out of 8 marks. Even with having experience more than 3 years yet the participants were unable to reach the full score.

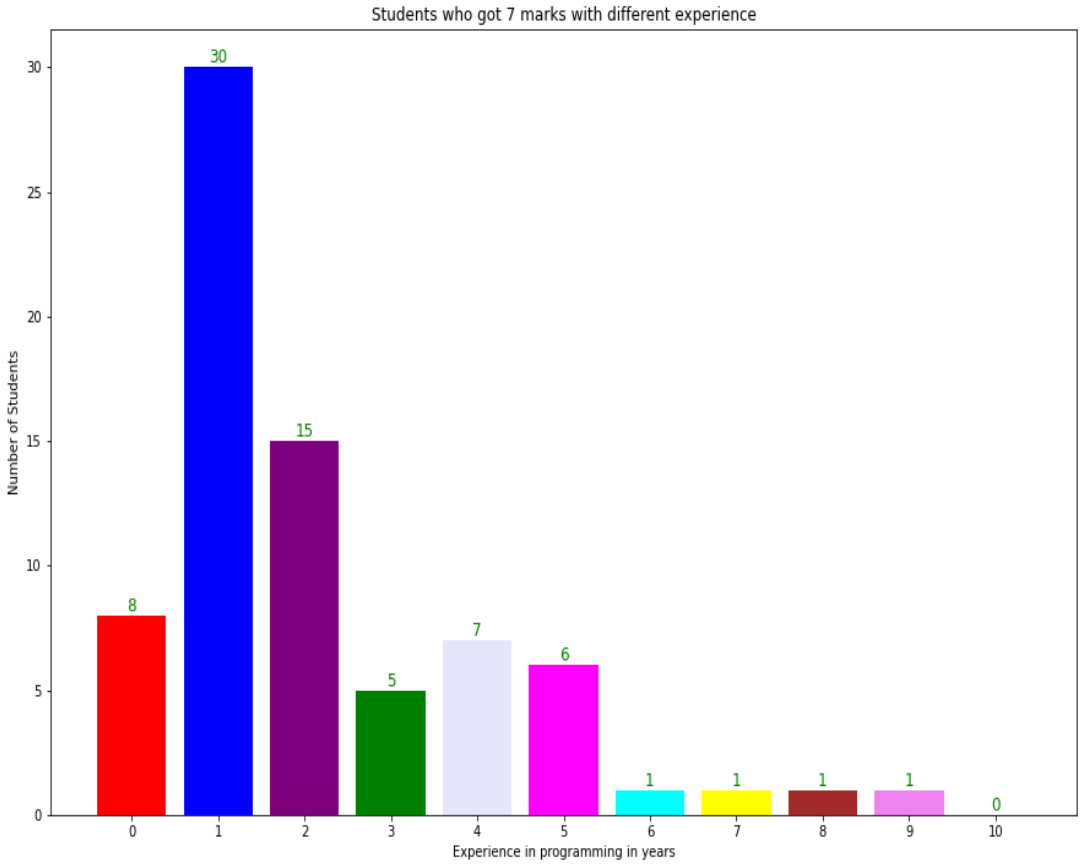


Figure 4.5: Students who scored 7 marks with different experiences in programming.

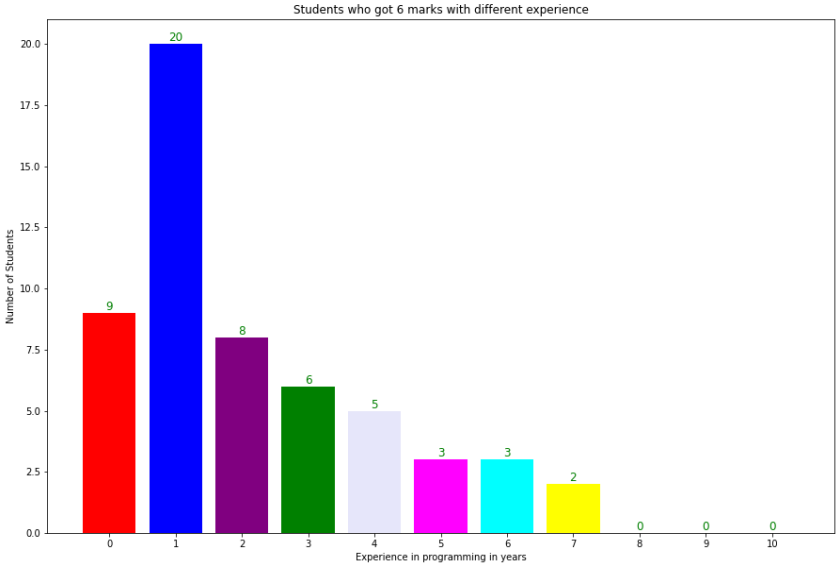


Figure 4.6: Students who scored 6 marks with different experiences in programming.

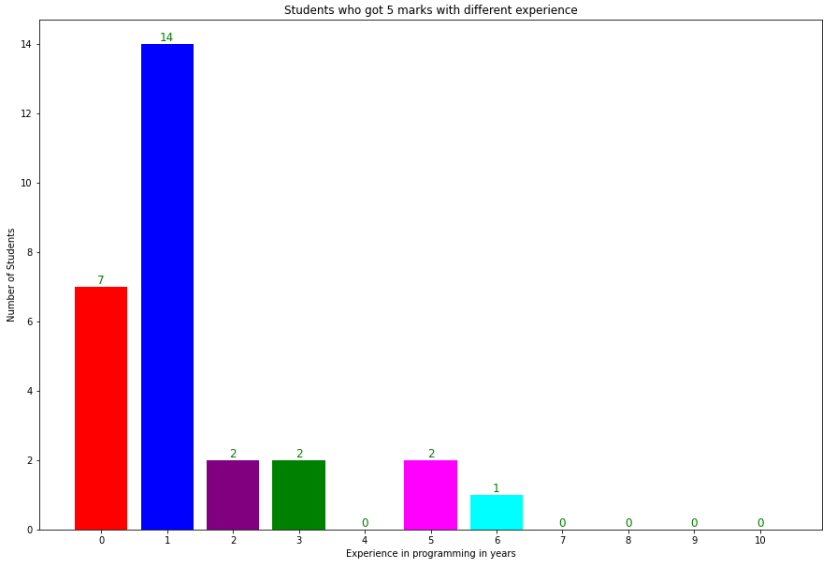


Figure 4.7: Students who scored 5 marks with different experiences in programming.

In the next category Figure 4.6 and Figure 4.7 where students who scored 6 and 5 marks out of 8 also included students whose experience varied from 0 to 7 years. It did raise questions that why the students who had experience more than 1 year are stuck at these scores.

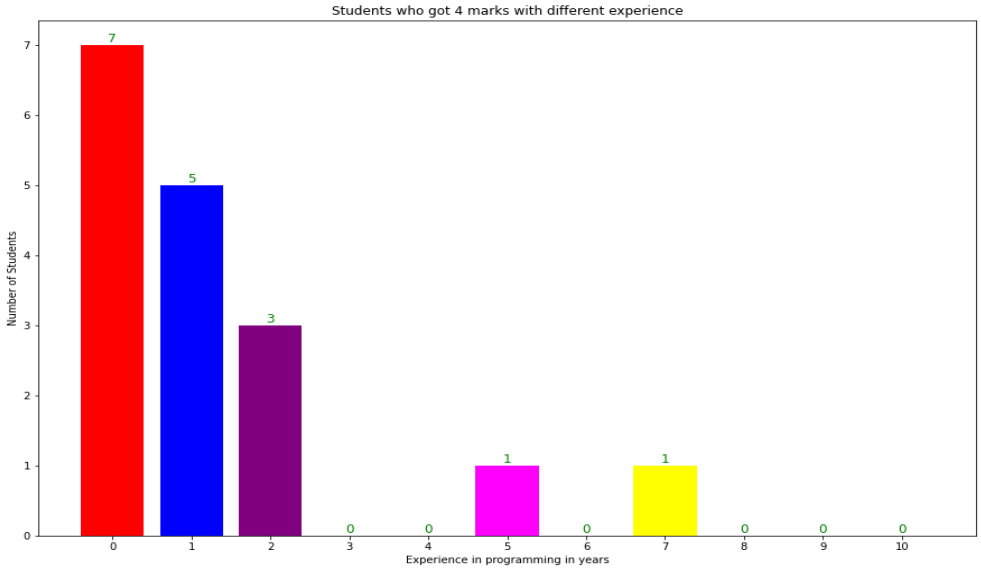


Figure 4.8: Students who scored 4 marks with different experiences in programming.

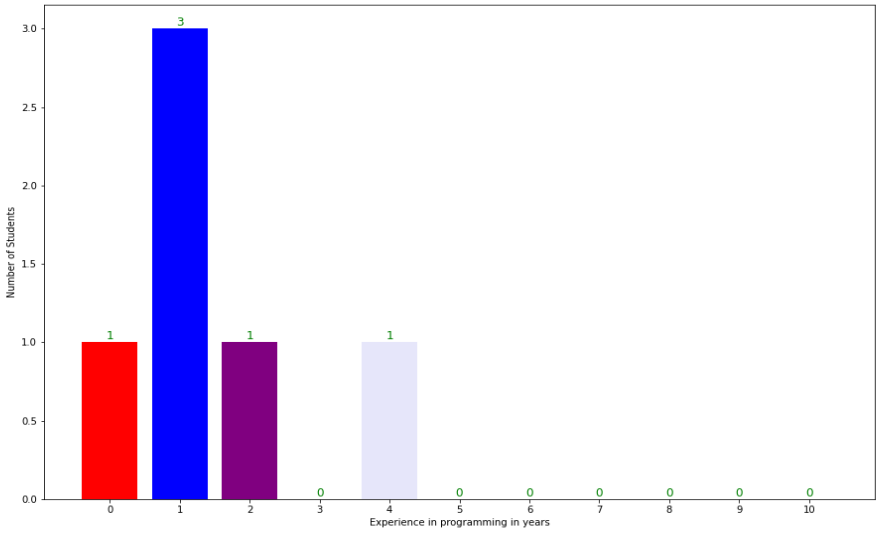


Figure 4.9: Students who scored 3 marks with different experiences in programming.

In Figure 4.8 included 1 student with experience of 7 and 5 years but scored 4 marks. But 7 students with no experience scored 4 marks. In Figure 4.9 1 student with 4 years of experience scored 3 marks.

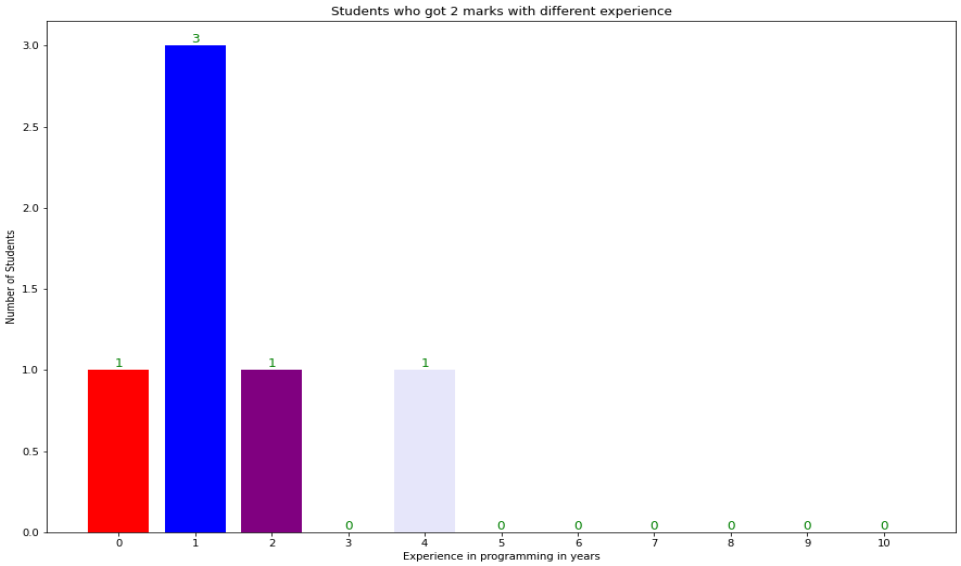


Figure 4.10: Students who scored 2 marks with different experiences in programming.

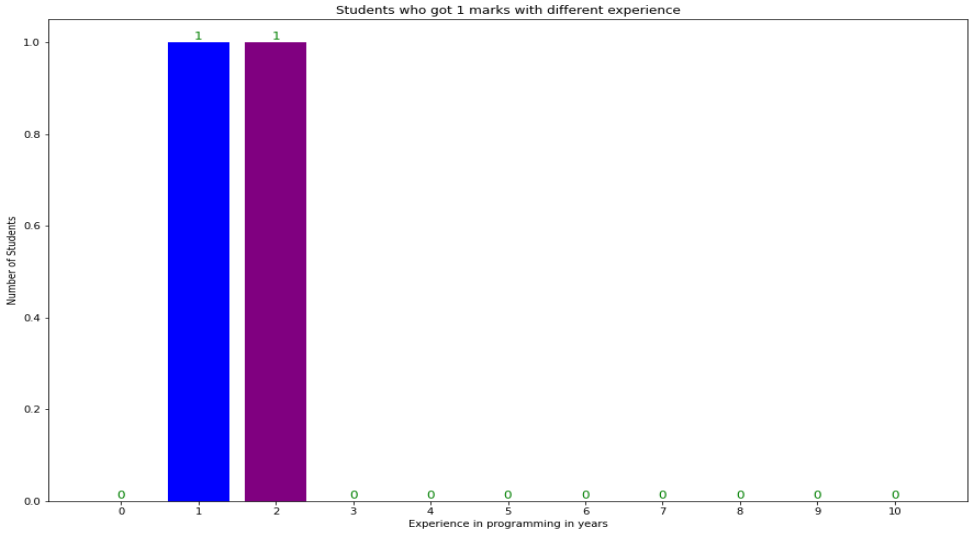


Figure 4.11: Students who scored 1 mark with different experiences in programming.

Figure 4.11 included 1 student with 4 years of experience with a score of 2, whereas in Figure 4.12 had only 2 students with experience of 1 and 2 years who scored 1 mark.

4.2.2.7 Neo Piagetian Summary for test 1

On analyzing the scores of test 1 we can now make our statement by presenting our findings based on Neo Piagetian theory and characterizing the scores of students in which stage does are the students currently.

1. According to table 4.3, where the students who scored 2 scored more than 50% tend to have more grasp and understanding over the assignment questions. They were able to fill the correct values in the box in the online survey test 1. They showed poor performance in the remaining questions, which showed they had a poor grasp in attempting the questions with the sequential assignment. Thus, they lie or work in the late Sensorimotor/Early preoperational stages.

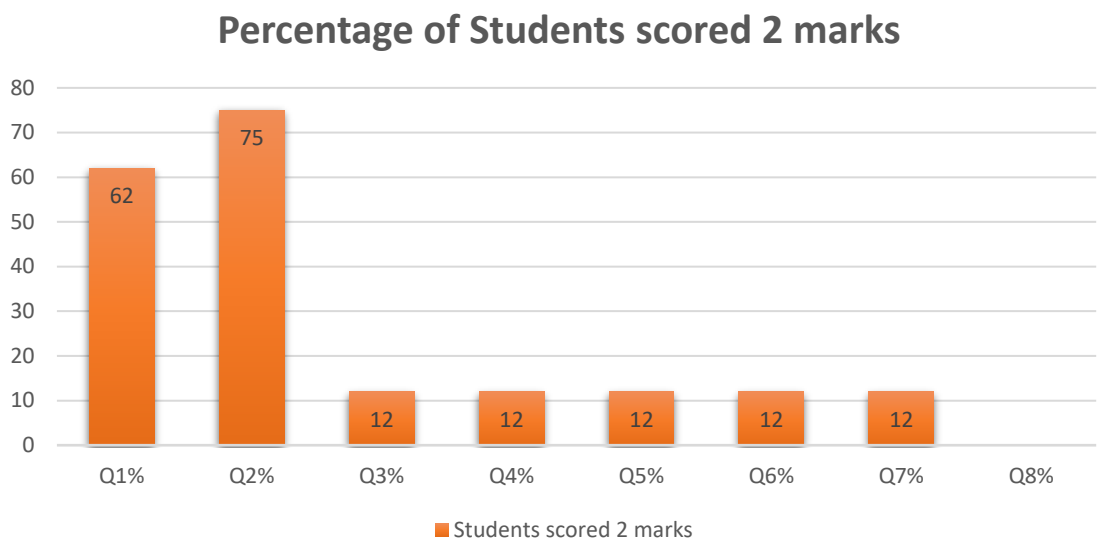


Figure 4.12: Percentage of students who scored 2 marks and answered which questions correctly.

2. The students who scored 4 showed better performance in Q3 which was to analyze whether they understood how the assignment in sequential statement happen. About 63% correctly answered this question, but they showed a poor grasp in tracking intermediate values in later questions in Q4 and Q5. They struggled with inductive reasoning and remaining questions. We characterize these students in mid-range Preoperational.

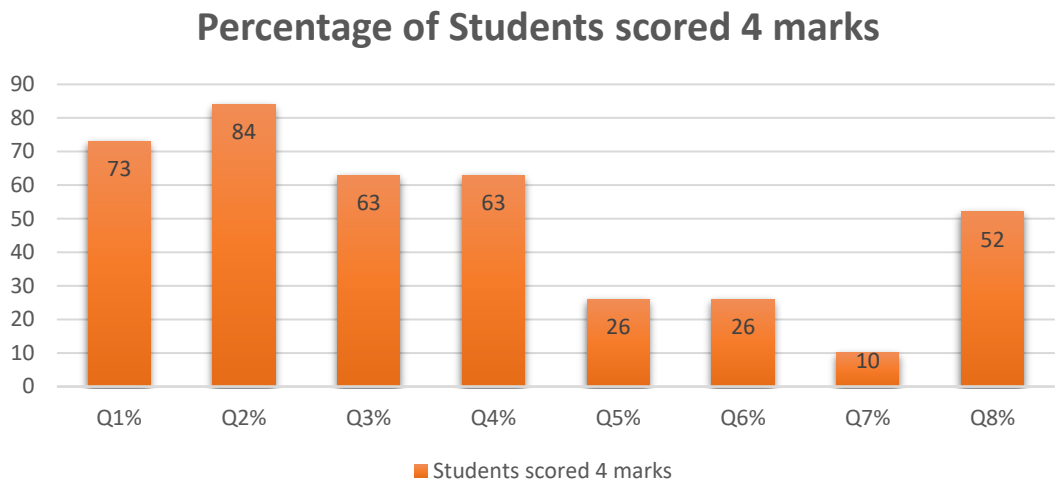


Figure 4.13: Percentage of students who scored 4 marks and answered which questions correctly.

3. Students with a score of 6 performed much better and were successful in attempting and understanding the assignment, sequential, and in tracking the intermediate values. The majority of them successfully got them correct but could not perform better in inductive reasoning questions. Only 37% of students were able to attempt successfully inductive reasoning. So, we could characterize these students in mid-range Preoperational/Late Preoperational.

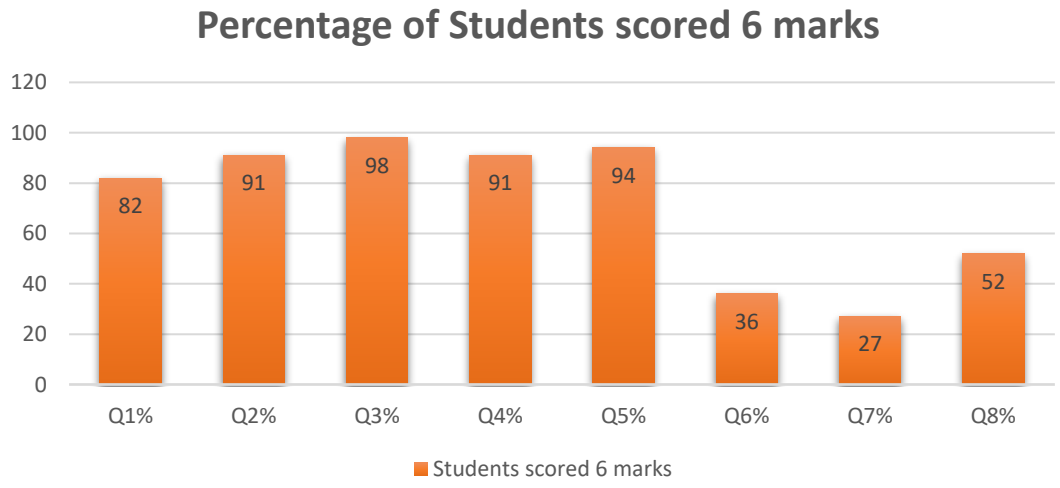


Figure 4.14: Percentage of students who scored 6 marks and answered which questions correctly.

4. Students who scored 8 were the students who were successful in correctly answering each question, and these were the students who were able to perform well in inductive and deductive reasoning. Their performance was consistent in every question. So, we characterize the students in Concrete Operational, who showed the right approach in solving all the questions.

Classification of test scores according to Neo Piagetian Theory for the TEST 1 Scores

In [19]:

```
for i,row in read_file_final.iterrows():
    if row['EIOS1_Tot_per'] ==100.0:
        print ("{}     {}       {}        Concrete Operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_per'])))

    elif row['EIOS1_Tot_per']>=75.0 and row['EIOS1_Tot_per'] <= 85.0:
        print ("{}     {}       {}          early concrete operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_

    elif row['EIOS1_Tot_per']> 85.0 and row['EIOS1_Tot_per'] <= 99.0:
        print ("{}     {}       {}          Mid Concrete operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_pe

    elif row['EIOS1_Tot_per']<75.0 and row['EIOS1_Tot_per']>50.0:
        print ("{}     {}      {}           Late Pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_per'])

    elif row['EIOS1_Tot_per']==50.0:
        print ("{}     {}      {}           mid range Pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_p

    elif row['EIOS1_Tot_per']>25.0 and row['EIOS1_Tot_per']<50.0:
        print ("{}     {}      {}           pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_per'])))

    elif row['EIOS1_Tot_per']==25.0:
        print ("{}     {}      {}           Sensorimotor/early Pre-Operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EI

    elif row['EIOS1_Tot_per']<25.0 and row['EIOS1_Tot_per']!=11.0:
        print ("{}     {}      {}           Sensorimotor".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS1_Tot_per'])))
```

```
0     nan       87.5          Mid Concrete operational
1     1mamao      87.5          Mid Concrete operational
2     4hSSaa      100.0        Concrete Operational
3     2eRHam     62.5           Late Pre operational
4     23MMSCM      100.0        Concrete Operational
5     mainh1      75.0          early concrete operational
6     2eRYam      87.5          Mid Concrete operational
7     3uMIoh      100.0        Concrete Operational
8     9uGCaa      75.0          early concrete operational
9     5mGTub      100.0        Concrete Operational
10     4wusuu      75.0           early concrete operational
11     565601      75.0           early concrete operational
12     3eSSaa      75.0           early concrete operational
13     2iANho      100.0        Concrete Operational
14     0aRKou      75.0           early concrete operational
15     2omhon      62.5           Late Pre operational
16     6uvsai      50.0           mid range Pre operational
17     2sreeja       75.0          early concrete operational
18     2ajayk      37.5           pre operational
```

Figure 4.15: Classification of test scores according to the Piagetian Theory for test1 scores.

4.3 Conducting test 2

Since both, the test must be conducted in 6 months in beginning and at the end of the semester to record the performance of students that at which Neo Piagetian stage do, they lie after they get familiar with a correct model to follow which they would have learned through lectures and test 1.

Test 2 contained two parts, which we had to design according to the present situation because of the pandemic, where the students could take the survey and the test by the end of the semester. The pandemic survey was necessary, as we had to find out that online courses conducted in the university were helpful and there are many more factors that are associated with it that could affect the performance of students.

As mentioned, test 2 was conducted in two parts, in one part we received the test scores from the professors (Table 4.10 and Table 4.11) where the students gave the test online and recorded the scores in the excel file. The other part consisted of questions that were entirely about the pandemic. The university was closed for an entire year and the students have to attend the online courses to complete their semester. The courses were designed accordingly in the best way which could be favorable for the students. As the survey was conducted in a very fixed duration of time, where the students were asked about their opinions and suggestions so, they could plan the structure of the study program which could fulfill the needs of the students and there could be more sessions of queries which could be arranged at the end of the lectures.

Code	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Sum
1eASaa	3	4	7	7	9	10	-	40
7nHGha	3	4	7	7	8	10	-	39
6aJGhh	3	4	7	7	8	10	-	39
1sPSaa	3	4	7	6	9	10	-	39
1aDBhh	3	4	7	5	8	10	-	37
7nKTra	3	4	7	6	8	8	-	36
3aUGho	3	4	5	7	6	10	-	35
9iRHha	3	3	6	7	7	9	-	35

Table 4.10: Data (Sample) received from the professor for test 2.

PD14_01	Test 1	Final Exam
8mjtia	2	
	7	
1mamao	7	19.3
4hSSaa	8	19
2eRHam	5	
23MMSCM	8	20.5
mainh1	6	29
2eRYam	7	20.5
3uMloh	8	19
9uGCaa	6	
5mGTub	8	
4wusuu	6	7.5
565601	6	26.5
3eSSaa	7	
2iANho	8	
0aRKou	6	7.5
2omhon	5	29
6uvsai	4	15
2sreeja	6	

Table 4.11: Data (Sample) received in Test 2.

At the end of the survey, there were responses which we felt could be the reason for the students who could not perform that well as in the first test. The changes in the Neo Piagetian stage in which the student was earlier some stayed at the same stage, some made a progress in their case and some even moved from a higher level to a lower one.

Questions (Appendix A.2) that were included in the test 2 survey related to the pandemic. It contained starting questions that were general to ask about student grade and their unique code. The rest of the questions were more study-specific, which were focused on gaining information regarding their opinions about the learning platform which they use and the last few questions regarding the pandemic.

The study and the pandemic questions were designed again using the SoSci Survey tool same as used for test 1. Before attempting students are prompted to enter their unique code and general details about their course. In every question, they had to select their choice and move to the next page. The possible options were present according to the queries which a student might select for example Yes, no, or Not answered (Table 4.12).

The results are generated in the form of Docx, CSV format as the tool can provide results in the number of formats whichever could be suitable according to the user. The columns contained such as ST08_01, ST09, ST11_01, etc. consist of values that students selected in the test. The other test results provided by professors, wherein one test the score students scored was out of 50 and another was out of 32 marks. Since in every test we had different total scores so to compare the score we converted each score into percentage and then analyzed it. Based on Neo Piagetian it was more effective by converting it into percentage and then categorizing the students in which stage they lie. It gave us better insights to see whether students stayed at the same stage or did they learned the correct model or approach to solving the questions in the test and moved to a higher stage. Also, the ones who were on higher stage performed in the same way as we predicted or did, they fall into lower stage and what was the reason which affected that.

PD14_01	ST01	ST03	ST02	ST04	ST05_01	ST06	ST07	ST08_01	ST09	ST10
3oRGhi	2	1		1	regelmäßig	1	4		2	1
7eTTnl	2	2				2			2	1
1ejewa	2	2				1	2		2	2
5AHSLO	2	2				2			1	2
	1	2	2			1	3		1	1
8nTGra	2	2				2			2	2
	2	2				2			1	2
3 a H L h i	1	2	2			1	3		2	2
4vKLeu	2	2				2			3	3
	1	1	1	1		1	1		1	1
6yrloo	2	2				2			2	2
1eMGuu	2	2				2			2	1
8aAHmu	1	1	1	1	2 Mal	2			3	3
0eRGal	1	2	2			2			3	1

Table 4.12: The pandemic survey data (Sample) in test 2

Classification of test scores according to Neo Piagetian Theory for the TEST 2 Scores

```
In [20]: for i,row in read_file_final.iterrows():
             if row['EIOS2_Summe_per'] ==100.0:
                 print ("{}     {}       {}        Concrete Operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe_per']))

             elif row['EIOS2_Summe_per']>=75.0 and row['EIOS2_Summe_per'] <= 85.0:
                 print ("{}     {}       {}          early concrete operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summ

             elif row['EIOS2_Summe_per']> 85.0 and row['EIOS2_Summe_per'] <= 99.0:
                 print ("{}     {}       {}          Mid Concrete operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe_

             elif row['EIOS2_Summe_per']<75.0 and row['EIOS2_Summe_per']>50.0:
                 print ("{}     {}      {}           Late Pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe_per'

             elif row['EIOS2_Summe_per']==50.0:
                 print ("{}     {}      {}           mid range Pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe

             elif row['EIOS2_Summe_per']>25.0 and row['EIOS2_Summe_per']<50.0:
                 print ("{}     {}      {}           pre operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe_per'])))

             elif row['EIOS2_Summe_per']==25.0:
                 print ("{}     {}      {}           Sensorimotor/early Pre-Operational".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EI

             elif row['EIOS2_Summe_per']<25.0 and row['EIOS2_Summe_per']!=11.0:
                 print ("{}     {}      {}           Sensorimotor".format((i),(row['PD14_01(EIOS1/EIOS2)']),(row['EIOS2_Summe_per'])))
```

```
117     1eASaa       76.92307692307693          early concrete operational
118     9apjhn      53.84615384615385          Late Pre operational
119     2oTRaa      3.8461538461538463          Sensorimotor
120     4tMMeu      32.69230769230769          pre operational
121     5hSLoa      26.923076923076923          pre operational
124     1nHRce      26.923076923076923          pre operational
125     3yDKhe      61.53846153846154          Late Pre operational
126     9iSMeo      19.230769230769234          Sensorimotor
127     0oPEah      7.6923076923076925          Sensorimotor
129     5nCDei      51.92307692307693          Late Pre operational
131     3aUGho      67.3076923076923          Late Pre operational
132     8iAGue      7.6923076923076925          Sensorimotor
133     0aFTea      21.153846153846153          Sensorimotor
134     2tRBaN      57.692307692307686          Late Pre operational
135     0trmhi      50.0          mid range Pre operational
137     2iEGül      48.07692307692308          pre operational
138     9oMGun      65.38461538461539          Late Pre operational
139     2IABUI      61.53846153846154          Late Pre operational
140     3rCHei      19.230769230769234          Sensorimotor
```

Figure 4.16: Classification of test 2 scores according to Neo Piagetian theory.

145	3aKBhi	100.0	concrete operational	70.0	Late Pre operational
148	2rRGeh	100.0	concrete operational	48.0	pre operational
149	9nRZPu	100.0	concrete operational	14.000000000000002	Sensorimotor
151	2nGAhe	100.0	concrete operational	62.0	Late Pre operational
154	3tHGno	100.0	concrete operational	62.0	Late Pre operational
155	4eUGoe	100.0	concrete operational	48.0	pre operational
158	7nMEta	100.0	concrete operational	60.0	Late Pre operational
160	2iABui	100.0	concrete operational	64.0	Late Pre operational
162	8aBMue	100.0	concrete operational	50.0	mid range Pre operational
163	8lHHsu	100.0	concrete operational	68.0	Late Pre operational
176	5eAMha	100.0	concrete operational	14.000000000000002	Sensorimotor
178	7aAnhe	100.0	concrete operational	62.0	Late Pre operational
183	6aOGho	100.0	concrete operational	68.0	Late Pre operational
191	6eCYiu	100.0	concrete operational	60.0	Late Pre operational
196	7nHGha	100.0	concrete operational	78.0	early concrete operational
204	6aJGhh	100.0	concrete operational	78.0	early concrete operational
228	7aJLha	100.0	concrete operational	6.0	Sensorimotor
237	4iMVha	100.0	concrete operational	16.0	Sensorimotor
238	6vTGlo	100.0	concrete operational	30.0	pre operational
252	9iJDii	100.0	concrete operational	50.0	mid range Pre operational
257	5nFKho	100.0	concrete operational	66.0	Late Pre operational
269	5iMGei	100.0	concrete operational	54.0	Late Pre operational
281	5aMGht	100.0	concrete operational	26.0	pre operational
290	1lPUrb	100.0	concrete operational	52.0	Late Pre operational

2	4hSSaa	100.0	concrete operational	59.375	Late Pre operationa
4	23MMSCM	100.0	concrete operational	64.0625	Late Pre operationa
7	3uMIoh	100.0	concrete operational	59.375	Late Pre operationa
19	5IAPUU	100.0	concrete operational	55.625	Late Pre operationa
26	1oSEPB	100.0	concrete operational	64.0625	Late Pre operationa
40	8arnaa	100.0	concrete operational	90.625	Mid Concrete operational
44	9AUJAA	100.0	concrete operational	35.3125	pre operational
53	9tbmal	100.0	concrete operational	56.25	Late Pre operationa
79	9oamya	100.0	concrete operational	28.125	pre operational
80	5amaao	100.0	concrete operational	28.125	pre operational

Figure 4.17: Comparison between the test 1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piageti an Stages. (i)

121	5hSLoa	87.5	Mid Concrete operational	28.000000000000004	pre operational
126	9iSMeo	87.5	Mid Concrete operational	20.0	Sensorimotor
127	0oPEah	87.5	Mid Concrete operational	8.0	Sensorimotor
132	8iAGue	87.5	Mid Concrete operational	8.0	Sensorimotor
133	0aFTea	87.5	Mid Concrete operational	22.0	Sensorimotor
143	4nRGhr	87.5	Mid Concrete operational	6.0	Sensorimotor
150	8aRGaa	87.5	Mid Concrete operational	62.0	Late Pre operationa
153	1aDBhh	87.5	Mid Concrete operational	74.0	Late Pre operationa
156	5lMFsh	87.5	Mid Concrete operational	62.0	Late Pre operationa
165	9aBWho	87.5	Mid Concrete operational	62.0	Late Pre operationa
172	9awkyu	87.5	Mid Concrete operational	50.0	mid range Pre operational
186	5nDFhi	87.5	Mid Concrete operational	60.0	Late Pre operationa
188	2AMMLE	87.5	Mid Concrete operational	14.000000000000002	Sensorimotor
189	4iTGho	87.5	Mid Concrete operational	36.0	pre operational
190	4iREhe	87.5	Mid Concrete operational	64.0	Late Pre operationa
193	3iUAhi	87.5	Mid Concrete operational	57.99999999999999	Late Pre operationa
194	3iaxaa	87.5	Mid Concrete operational	28.000000000000004	pre operational
195	8aFGir	87.5	Mid Concrete operational	8.0	Sensorimotor
197	1eMLsa	87.5	Mid Concrete operational	56.00000000000001	Late Pre operationa
214	2eTGer	87.5	Mid Concrete operational	50.0	mid range Pre operational
217	4yMAta	87.5	Mid Concrete operational	12.0	Sensorimotor
226	1nHGre	87.5	Mid Concrete operational	54.0	Late Pre operationa
229	1eWGha	87.5	Mid Concrete operational	64.0	Late Pre operationa
233	8lwgoa	87.5	Mid Concrete operational	60.0	Late Pre operationa

1	1mamao	87.5	Mid Concrete operational	60.3125	Late Pre operationa
6	2eRYam	87.5	Mid Concrete operational	64.0625	Late Pre operationa
29	1ATSUI	87.5	Mid Concrete operational	69.6875	Late Pre operationa
32	6amfau	87.5	Mid Concrete operational	52.5	Late Pre operationa
41	3rb4ai	87.5	Mid Concrete operational	47.8125	pre operational
42	3isght	87.5	Mid Concrete operational	66.5625	Late Pre operationa
55	8anvyr	87.5	Mid Concrete operational	28.125	pre operational
56	9ipiyh	87.5	Mid Concrete operational	28.125	pre operational
57	5sMKer	87.5	Mid Concrete operational	73.125	Late Pre operationa
60	3aRKye	87.5	Mid Concrete operational	93.75	Mid Concrete operational
72	1eVKot	87.5	Mid Concrete operational	79.375	early concrete operational
73	1ntgai	87.5	Mid Concrete operational	82.8125	early concrete operational
74	9nRGhn	87.5	Mid Concrete operational	59.375	Late Pre operationa
81	9rMThd	87.5	Mid Concrete operational	87.1875	Mid Concrete operational

Figure 4.18: Comparison between the test 1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piageti an Stages. (ii)

129	5nCDei	62.5	Late Pre operational	54.0	Late Pre operational
192	5iOPri	62.5	Late Pre operational	6.0	Sensorimotor
218	8aRHui	62.5	Late Pre operational	48.0	pre operational
225	9aawro	62.5	Late Pre operational	56.00000000000001	Late Pre operational
227	9oJGwa	62.5	Late Pre operational	52.0	Late Pre operational
246	6hhhoi	62.5	Late Pre operational	60.0	Late Pre operational
254	4aRGür	62.5	Late Pre operational	62.0	Late Pre operational
265	0rAEoa	62.5	Late Pre operational	54.0	Late Pre operational

15	2omhon	62.5	Late Pre operational	90.625	Mid Concrete operational
38	0artre	62.5	Late Pre operational	74.375	Late Pre operational
65	9rMRer	62.5	Late Pre operational	65.625	Late Pre operational

Figure 4.19: Comparison between the test 1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piageti an Stages. (iii)

119	2oTRaa	50.0	mid range Pre operational	4.0	Sensorimotor
125	3yDKhe	50.0	mid range Pre operational	64.0	Late Pre operationa
157	5oGMio	50.0	mid range Pre operational	10.0	Sensorimotor
243	9rTFia	50.0	mid range Pre operational	32.0	pre operational
247	2ajaeo	50.0	mid range Pre operational	48.0	pre operational
248	8VJKOI	50.0	mid range Pre operational	56.00000000000001	Late Pre operationa

16	6uvsai	50.0	mid range Pre operational	46.875	pre operational
51	1aacho	50.0	mid range Pre operational	50.0	mid range Pre operational
66	9eKOua	50.0	mid range Pre operational	28.125	pre operational
69	1AABEA	50.0	mid range Pre operational	65.625	Late Pre operationa

Figure 4.20: Comparison between the test 1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piageti an Stages. (iv)

118	9apjhn	0.0	Sensorimotor	56.00000000000001	Late Pre operationa
182	4lFRlu	0.0	Sensorimotor	6.0	Sensorimotor
216	8oakus	0.0	Sensorimotor	16.0	Sensorimotor
291	4ikGha	0.0	Sensorimotor	32.0	pre operational

118	9apjhn	0.0	Sensorimotor	56.00000000000001	Late Pre operationa
182	4lFRlu	0.0	Sensorimotor	6.0	Sensorimotor
216	8oakus	0.0	Sensorimotor	16.0	Sensorimotor
291	4ikGha	0.0	Sensorimotor	32.0	pre operational

Figure 4.21: Comparison between the test 1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piageti an Stages. (v)

4.3.1 Results (End of Semester)

Our findings presented the comparison between the performance of students in both test 1 and test 2 but after test 2 the results we presented here only of the students who participated in both the tests. We have attached the results in tabular form of all participating students. Performance results could vary, and it is not always predicted that the students who performed well in the first test could perform in the same way. But if we dig into this area, we can also throw some light on the factors which could be responsible for their bad performance, the survey results we generated from the responses would help to analyze the data more accurately in this direction. We had to make separate analyses at the end of the semester.

4.3.2 Neo-Piagetian Summary after test 2

1. About 40 students were present in both the tests and were successful to be categorizing themselves into concrete operational. Predictions about these students who were thought to know the correct model for program execution and would perform better by the end of the semester. All these students could not stay in the same stage according to their performance which revealed that some of them moved to the Early Concrete Operational stage, different Preoperational stage. Most of the students did go backward but did not huge transition of movement in Neo-Piagetian stages as test2 also required skills and concepts to approach the questions.
2. 47 students who were categorized into Mid Concrete Operational means were able to attempt 87% of the questions correctly. (Figure 4.22) About 23% of students could not perform better and did not score enough which is enough to take them to the sensorimotor stage. But 77% of students were successful in giving better performance. They could not stay in the same stage but did not give a bad performance which could lead to a huge shift to lower stages.

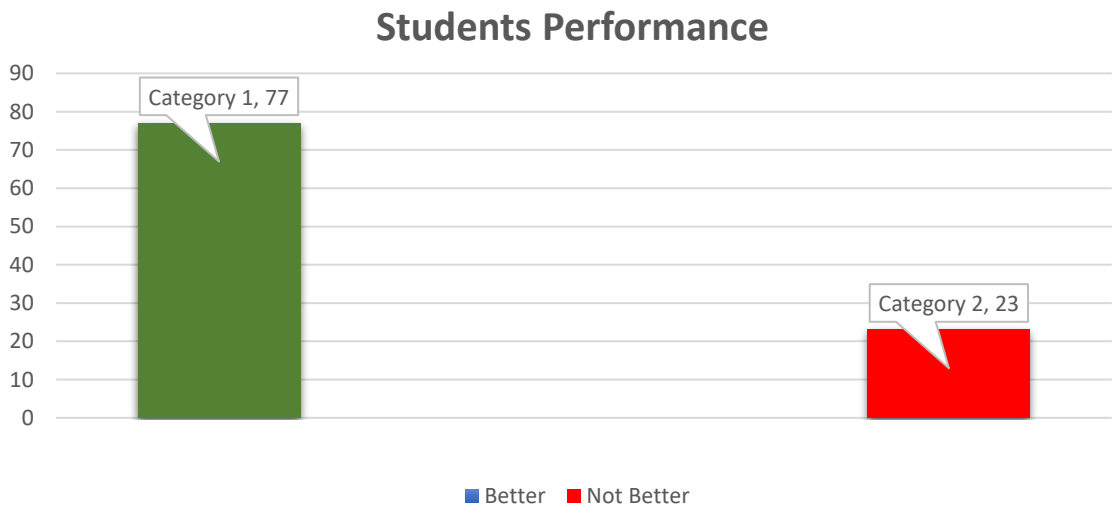


Figure 4.22: Student performances who were in higher stage (Mid Concrete Operational).

3. About 11 students who were in the category of late preoperational stage performed better in their performance which could be stated from their results as most of them stayed in the same stage late preoperational even after test 2.
4. (Figure 4.23) 10 students who were in the mid-range preoperational also performed well in test 2 same as their performance in test 1. Most of them stayed in the same stage which signifies, that they did grasp over the understanding of the model of execution of a program. Out of 10 students, 3 students showed the jump from the mid-range preoperational to late preoperational.

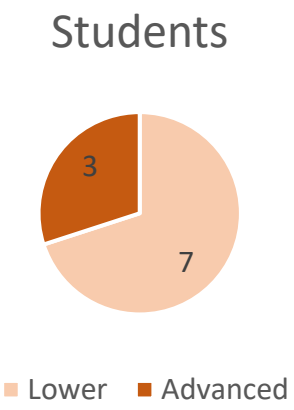


Figure 4.23: Student's performance who were in Mid Preoperational Stage.

5. 6 Students were in the sensorimotor stages in test 1. By the end of the test (Figure 4.24) 2 out of 6, 2 students showed better performance one student moved from sensorimotor stage to preoperational and another one was made to late preoperational.

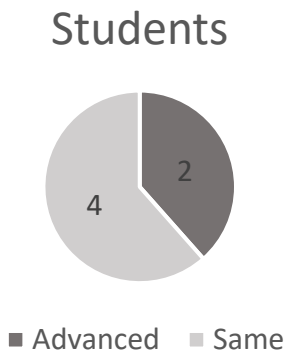


Figure 4.24: Student's performance who were in Sensorimotor Stage.

6. There were some additional results to which brought some interesting observations to our research that there were 4 students who were unable to perform in test 1 and scored 0 which we could categorize as sensorimotor or maybe not. But out of 4 students (Figure 4.25), 2 students scored the marks which were enough to put them in the stage of pre-operational and another one into the stage of late preoperational. It clearly states that by the end of the

semester when given formal instruction of the statements and programming problems they get to know the idea of attempting the question and which model they should follow.

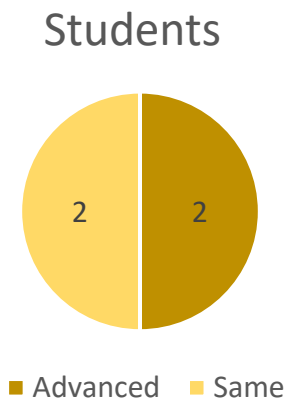


Figure 4.25: Student's performance in who scored 0 scores.

The performance of students in both the tests gave us insights on various points which we discussed before and here is one question which arises that even most of the students performed well by the end of the semester and still many could not show the transition from lower Neo Piagetian stages to higher ones or vice versa. This brought our attention towards considering the opinions of students which they provided us in the survey about the difficulties which they could have faced during the course.

Ques. Overall how does pandemic affect your study?

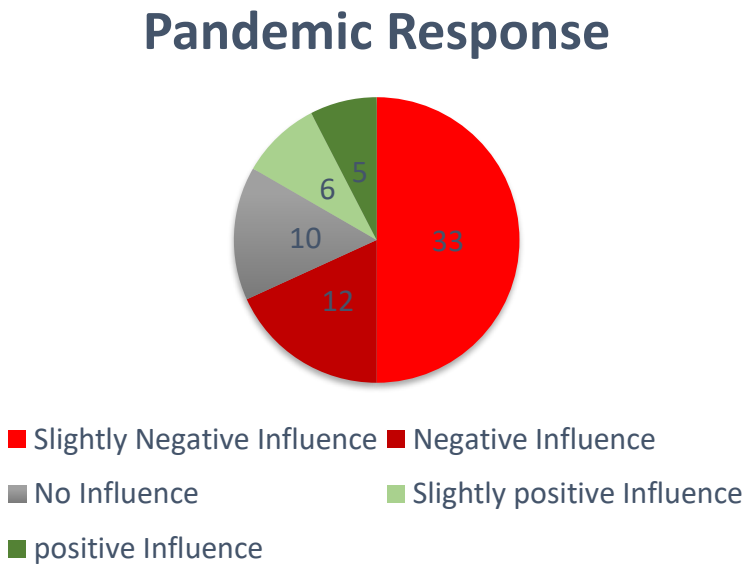


Figure 4.26: Pandemic Responses provided by students.

We asked the students did the pandemic affected their study sessions. We received 62 responses from the students. Out of 62 responses (Figure 4.26), 33 students chose that it brought a slightly negative influence and 12 students felt that it had a negative influence and as a result affected their result and the remaining opinions included no influence or positive influence. But more than 50% of students felt that their study was affected due to a pandemic which was reflected in their scores in the tests. As most of them agreed that attending university and the environment for studying is important in the learning process.

5. Conclusion and Future Work

In our research, we tried to replicate the original study [7] we have conducted 2 tests one in the beginning and one by the end of the semester with students at the Chemnitz University of Technology, where students showed active participation in the test. As it is important to mention, that even during the pandemic the proper course structure was planned for learning skills and knowledge in the area of programming. The students were also provided with assignments and every query has been solved through group discussion or even by taking appointments with the instructors.

This research was conducted to predict the success rate among the students who begin to learn to program but many students did have previous experience or were familiar with programming languages. But we designed the test keeping in mind they all are at the beginning of the semester and attending online lectures to learn to program. Programming can be an innate talent and it cannot be denied, but our results did not focus on this part instead we tried to Figure out, if students get better at reasoning and understand the correct model which is required to attempt the programming questions, they will have better chances to pass the tests or introductory programming courses. Neo Piagetian focuses on developing cognitive skills which are primarily learned and cannot be considered as innate. As it can be clearly said that working on reasoning skills can help the student to getter better at learning programming languages.

5.1 Threats to Validity

1. Test 1 results showed better performance amongst students, and we tried to relate the results with Neo Piagetian theory. As many students were successful to achieve good marks and were fit in the category of the Concrete Operational Stage, but in test 2 students in this category were not able to show the same performance and secure their place in Concrete Operational Stage.
2. We found that many students possessed experience in programming but still were not able to reach higher Neo Piagetian stages. Even though if we assume these students were better in reasoning skills and had experience in programming then why they could not show the concrete reasoning skills by the end of the semester.

5.2 Future Work

1. These are some areas that need to be researched upon that the students who were able to secure their place in Concrete Operational Stage and scored full marks in test 1 could not give the same performance as in test 2.
2. Students who possessed experience in programming were unable to show the reasoning skills as expected which did reflect from their scores in both the tests.

Thus, more empirical work could be done in these two areas, that what other better methods could be adopted to make reasoning skills better so, it could help students to perform better at introductory courses.

5.3 Conclusion

In terms of Neo Piagetian theory, we predicted that the students who did not perform better and who exhibited a lower Neo Piagetian stage, in the beginning, were predicted to show higher stage by the end of the semester and will possess higher formal reasoning and better understanding towards solving programming questions.

Most of the students who could not perform better in test 1 and were in the lower stage of Neo Piagetian were able to manifest better results by the end of the semester. Even with the students who scored 0 as we took this case too, where 50% of students did reach the Late preoperational stage. 53 students were able to secure place in Late preoperational stage after test 2, as compared to test 1 where 11 students were in the same stage. There were also fewer students in the early stages such as Sensorimotor or Early Preoperational Stage and moved themselves to higher stages such as Preoperational and Mid-Range Pre-Operational. This signifies and supports our prediction that improving cognitive skills and reasoning and following the correct model of execution, is the dimension that should be worked upon. To push yourself to higher Neo Piagetian stages students must gain consistency in following the correct model and reasoning skills.

List of Figures

4.19: Comparison between the test1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piagetian stages. (iii)

4.20: Comparison between the test1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piagetian stages. (iv)

4.21: Comparison between the test1 scores and test 2 scores based on Neo Piagetian theory that how their performance shows the changes between the Neo Piagetian stages. (v)

4.22: Student performances who were in higher stage (Mid Concrete Operational).

4.23: Student's performance who were in Mid Preoperational Stage.

4.24: Student's performance who were in Sensorimotor Stage.

4.25: Student's performance in who scored 0 score.

4.26: Pandemic Responses provided by students.

List of Tables

References

[1] Adelson, B., & Soloway, E. (1985). The Role of Domain Experience in Software Design. *IEEE Transactions on Software Engineering*, *SE-11*(11), 1351–1360. https://doi.org/10.1109/tse.1985.231883

[2] Bonar, J., & Soloway, E. (1985). Preprogramming Knowledge: A Major Source of Misconceptions in Novice Programmers. *Human-Computer Interaction*, *1*(2), 133–161. https://doi.org/10.1207/s15327051hci0102_3

[3] Bell, V. A., & Johnson-Laird, P. N. (1998). A Model Theory of Modal Reasoning. *Cognitive Science*, *22*(1), 25–51. https://doi.org/10.1207/s15516709cog2201_2

[4] Dehnadi, S., and Bornat, R. (2006) The camel has two humps (working title) https://www.cs.kent.ac.uk/dept_info/seminars/2005_06/paper1.pdf

[5] Bornat, R., Dehnadi, S., and Simon (2008) Mental models, consistency and programming aptitude. Tenth Conference on Australasian Computing Education (ACE '08). pp. 53-61. http://crpit.com/confpapers/CRPITV78Bornat.pdf

[6] Lister, R. (2011) Concrete and Other Neo-Piagetian Forms of Reasoning in the Novice Programmer. Thirteenth Australasian Computing Education Conference (ACE '11), pp. 9-18. http://crpit.com/confpapers/CRPITV114Lister.pdf

[7] Ahadi, A., Lister, R. and Teague, D., 2014. Falling Behind Early and Staying Behind When Learning To Program. [online] ResearchGate. Available at: https://www.researchgate.net/publication/268151571_Falling_Behind_Early_and_Staying_Behind_When_Learning_to_Program

[8] Simon, S. Fincher, A. Robins, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer, M. Hamilton, R. Lister, M. Petre, K. Sutton, D. Tolhurst, and J. Tutty. Predictors of success in a first programming course. In Proc. Eighth Australasian Computing Education Conference (ACE2006), pages 189–196. ACS, 2006

[9] Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins. An analysis of patterns of debugging among novice computer science students. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education pages 84–88. ACM Press, 2005.

[10] Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. *International Journal of Computer & Information Sciences*, *8*(3), 219–238. https://doi.org/10.1007/bf00977789.

[11] https://gitlab.hrz.tu-chemnitz.de/abhh--tu-chemnitz.de/early-indicators-of-success.

[12] Mascolo, M. F. (2015). Neo-Piagetian Theories of Cognitive Development. *International Encyclopedia of the Social & Behavioral Sciences*, 501–510. https://doi.org/10.1016/b978-0-08-097086-8.23097-3

Appendix

Appendix A

A.1 Survey Test 1 using the online survey tool SoSci, which included questions explained in section 4.1.1 where students required to select the options and complete the test.

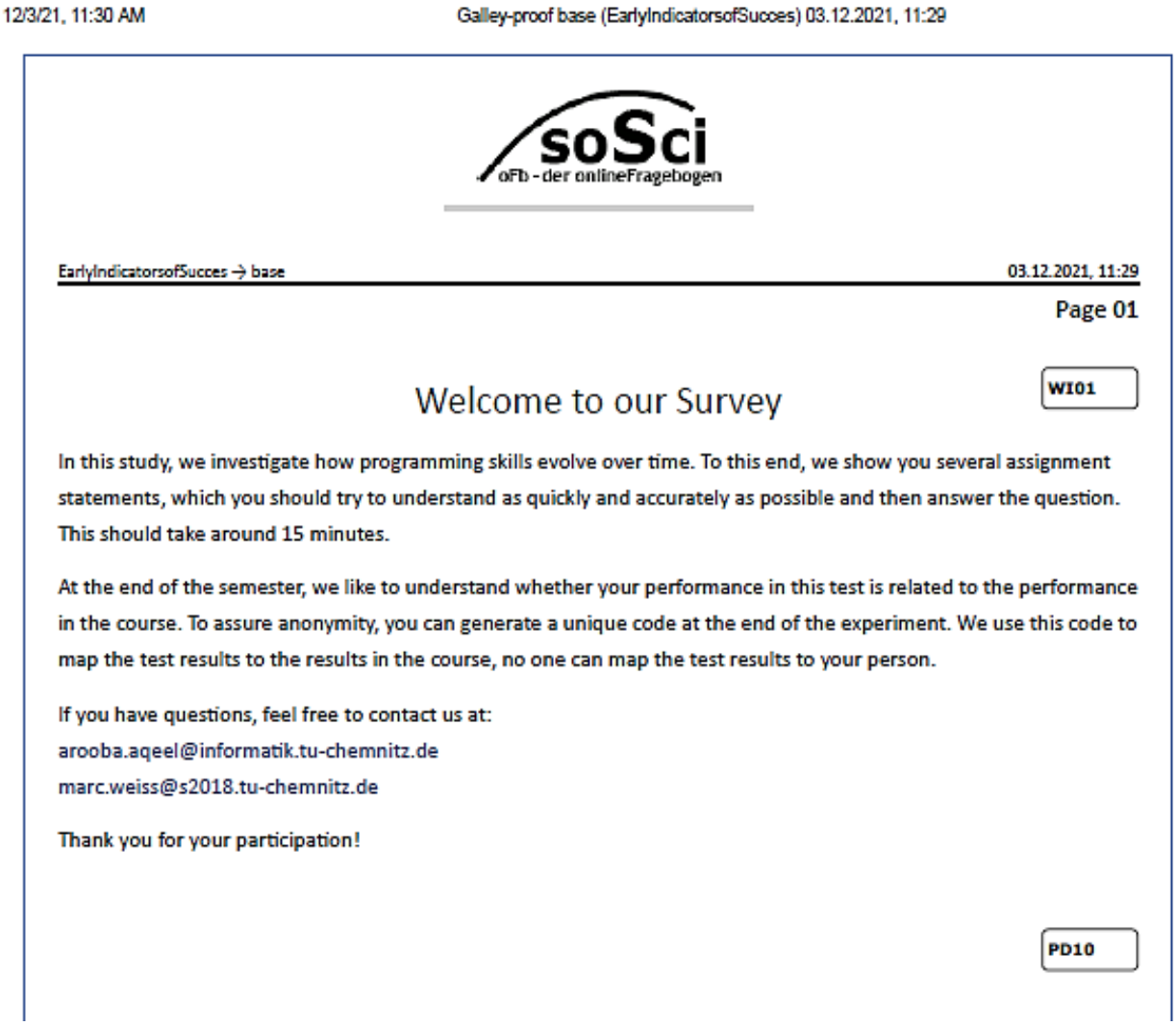


Figure A.1.1: test 1 for students Page 1

GQ12
test

1. Are you currently studying in:

Masters

Bachelors

Other:

GQ14

2. What is the name of your degree Program ?

Please enter the name of your degree program e.g. BS Informatik, MS Automative Engineering etc.

GQ02

3. Which course are you currently studying ?

Please enter the name of course e.g. Software Engineering Programming Basics , Softwaretechnologie, Softwarepraktikum etc.

GQ13

4. In which semester are you currently in ?

E.g. It is my 1 semester

It is my semester

Figure A.1.2: test 1 for students Page 2

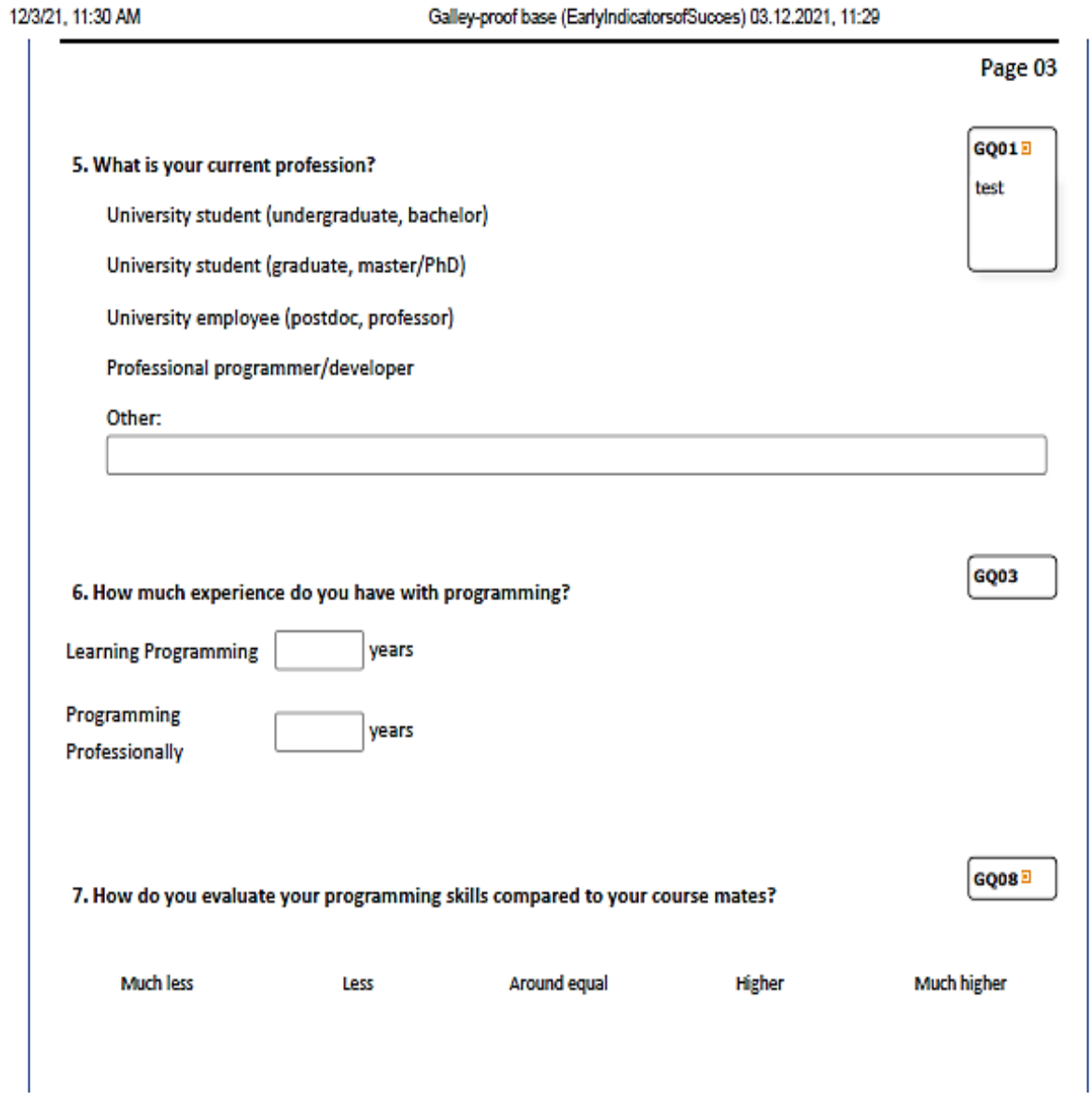


Figure A.1.3: test 1 for students Page 3

Page 04

GQ10

8. How experienced are you with C ?

Very inexperienced | Inexperienced | Moderately experienced | Experienced | Very experienced

GQ09

9. How experienced are you with C++ ?

Very inexperienced | Inexperienced | Moderately experienced | Experienced | Very experienced

GQ11

10. How experienced are you with Python ?

Very inexperienced | Inexperienced | Moderately experienced | Experienced | Very experienced

GQ05

11. How experienced are you with Java ?

Very inexperienced | Inexperienced | Moderately experienced | Experienced | Very experienced

GQ06

12. How many additional languages do you know (moderate experience or better)?

Page 05

S101

In the boxes, write the values in the variables after the following code has been executed:

a = 1

b = 2

a = 3

The value in a is [ ] and the value in b is [ ]

Figure A.1.4: test 1 for students Page 4

Page 06

S102

In the boxes, write the values in the variables after the following code has been executed:

r = 2

s = 4

r = s

The value in r is [ ] and the value in s is [ ]

Page 07

S103

In the boxes, write the values in the variables after the following code has been executed:

p = 1

q = 8

q = p

p = q

The value in p is [ ] and the value in q in [ ]

Page 08

S104

In the boxes, write the values in the variables after the following code has been executed:

x = 7

y = 5

z = 3

x = y

z = x

y = z

The value in x is [ ] y is [ ] and z is [ ]

Figure A.1.5: test 1 for students Page 5

Page 09

S105

In the boxes, write the values in the variables after the following code has been executed:

x = 7

y = 5

z = 0

z = x

x = y

y = z

The value in x is [ ] y is [ ] and z is [ ]

S106

In the code above, what do you observe about the final values in x and y?

Write your observation (in one sentence) in the box below.

Page 10

S107

In one sentence that you should write in the box below, describe the purpose of the following three lines of code, for any set of possible initial integer values stored in those variables.

Assume that variables i, j and k have been declared and initialized.

j = i

i = k

k = j

Page 11

```
Assume the variables "first" and "second" have been initialized.
```

S108

Write code to swap the values stored in first and second.

Figure A.1.6: test 1 for students Page 6

Page 12

PD07

13. What is your biological gender?

Male

Female

Prefer not say

PD08

14. How old are you ?

I am ____ years old.

Page 13

PD14

15. To enable us to link your data anonymously across multiple studies, we ask you to generate a unique [illegible] code below:

Please use the following instructions to compose the code

The **second** digit of your birthday.	The **second** letter of your mothers name	The **first** letter of your fathers name.	The **first** letter of your first schools name	The **second** letter of your city of birth	The **second** letter of your first name
e.g.: 04.12.1998	e.g.: Lisa	e.g.: Karl	e.g.: Goethe Schule	e.g.: Chemnitz	e.g.: Marla
4	i	K	G	h	a

PD13

16. If you would like to receive your code per e-mail, you can enter your e-mail address here. We will use your e-mail-address for no other purpose.

Email address

Unique Code

Figure A.1.7: test 1 for students Page 7

Page 14

WI03

Thank you for completing this survey!

If you have questions, feel free to contact us at:

arooba.aqeel@informatik.tu-chemnitz.de
marc.weiss@s2018.tu-chemnitz.de

Thank you for your participation!

Last Page

Thank you for completing this questionnaire!

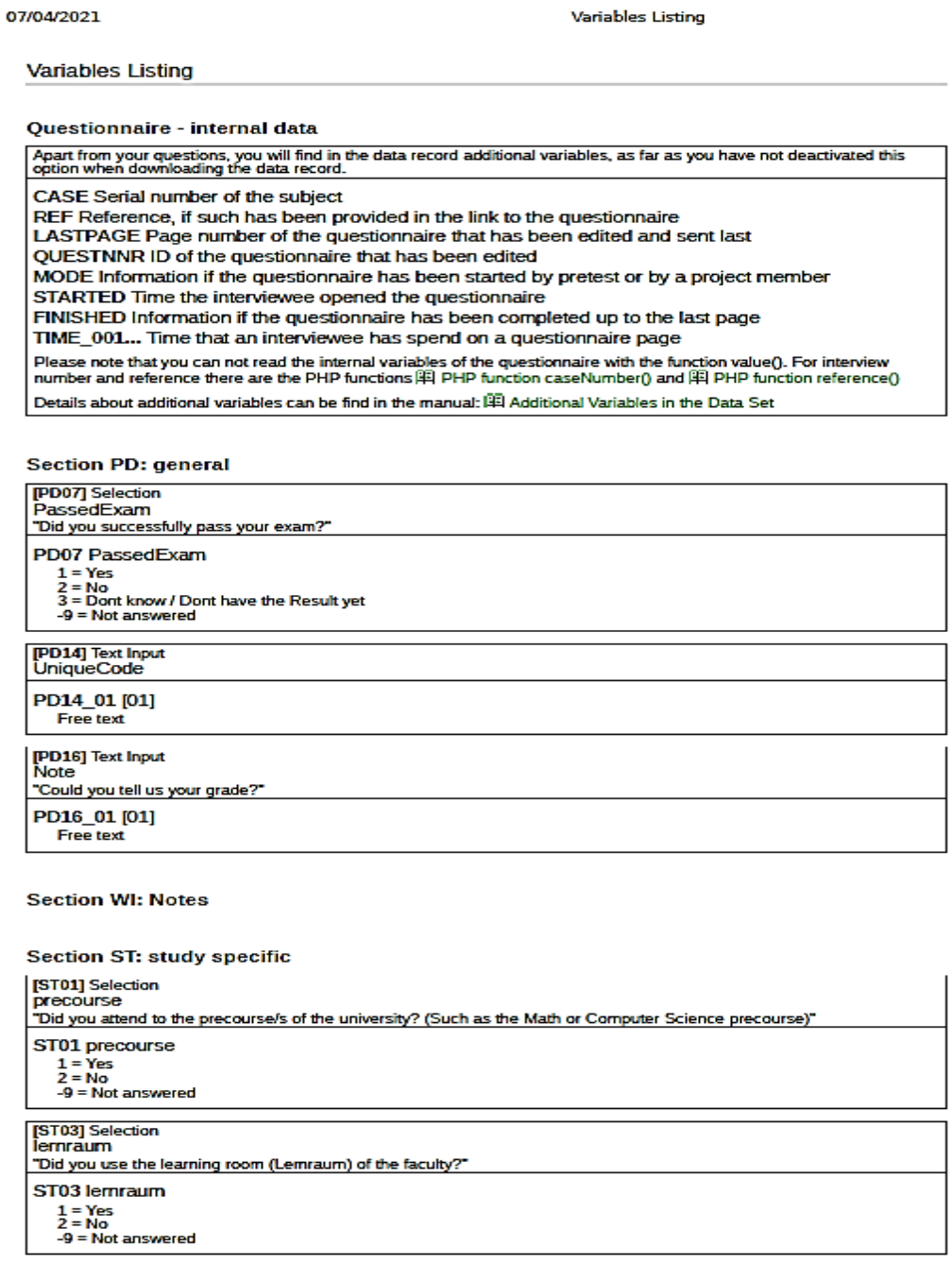
We would like to thank you very much for helping us.

If you have questions, feel free to contact us at:
arooba.aqeel@informatik.tu-chemnitz.de
marc.weiss@s2018.tu-chemnitz.de
Thank you for your participation!

Arooba Aqeel, Marc Weiß , Chemnitz University of Technology – 2020

Figure A.1.8: test 1 for students Page 8

A.2 Survey test 2 which included the questions about the pandemic, that how it affected their study whether it laid a positive or negative effect.



07/04/2021 Variables Listing

Variables Listing

Questionnaire - internal data

Apart from your questions, you will find in the data record additional variables, as far as you have not deactivated this option when downloading the data record.

CASE Serial number of the subject
REF Reference, if such has been provided in the link to the questionnaire
LASTPAGE Page number of the questionnaire that has been edited and sent last
QUESTNNR ID of the questionnaire that has been edited
MODE Information if the questionnaire has been started by pretest or by a project member
STARTED Time the interviewee opened the questionnaire
FINISHED Information if the questionnaire has been completed up to the last page
TIME_001... Time that an interviewee has spend on a questionnaire page

Please note that you can not read the internal variables of the questionnaire with the function value(). For interview number and reference there are the PHP functions PHP function caseNumber() and PHP function reference()

Details about additional variables can be find in the manual: Additional Variables in the Data Set

Section PD: general

[PD07] Selection
PassedExam
"Did you successfully pass your exam?"

PD07 PassedExam
1 = Yes
2 = No
3 = Dont know / Dont have the Result yet
-9 = Not answered

[PD14] Text Input
UniqueCode

PD14_01 [01]
Free text

[PD16] Text Input
Note
"Could you tell us your grade?"

PD16_01 [01]
Free text

Section WI: Notes

Section ST: study specific

[ST01] Selection
precourse
"Did you attend to the precourse/s of the university? (Such as the Math or Computer Science precourse)"

ST01 precourse
1 = Yes
2 = No
-9 = Not answered

[ST03] Selection
lernraum
"Did you use the learning room (Lernraum) of the faculty?"

ST03 lernraum
1 = Yes
2 = No
-9 = Not answered

Figure A.2.1: Survey questions for students in test 2

[ST02] Selection
precourse helpful
"Did you find the Precourse helpful in passing the exam?"

ST02 precourse helpful
1 = Yes
2 = No
-9 = Not answered

[ST04] Selection
lernraum helpful
"Did you find the learning room ("Lernraum") helpful in passing the exam?"

ST04 lernraum helpful
1 = Yes
2 = No
-9 = Not answered

[ST05] Text Input
lernraum how often
"How often did you use the learning room ("Lernraum")?"

ST05_01 [01]
Free text

[ST06] Selection
group
"Were you part of a learning group?"

ST06 group
1 = Yes
2 = No
-9 = Not answered

[ST07] Selection
groupSize
"How many people (you included) were in this group?"

ST07 groupSize
1 = 2 Persons
2 = 3 Persons
3 = 4 Persons
4 = 5 Persons
5 = 5 or more Persons
-9 = Not answered

[ST08] Text Input
groupMeeting
"How did you meet and communicate as a group during the semester?"

ST08_01 [01]
Free text

[ST09] Selection
tutorial
"Did you use the tutorial or other *offers* of the course?"

ST09 tutorial
1 = Yes
2 = No
3 = There where no such offers
-9 = Not answered

[ST10] Selection
selfLearningMaterial
"Did you use selflearning material provied by the course?"

ST10 selfLearningMaterial
1 = Yes
2 = No
3 = There where no such offers
-9 = Not answered

Figure A.2.2: Survey questions for students in test 2

[ST11] Text Input
CourseTime
"Overall, how many hours per week did you invest for the course?"

ST11_01 [01]
Free text

[ST12] Text Input
ExamTime
"Overall, how many hours did you invest in preperation for the exam?"

ST12_01 [01]
Free text

[ST13] Selection
futherGropu
"Do you plan on further doing learning groups?"

ST13 futherGropu
1 = Yes
2 = No
-9 = Not answered

[ST14] Text Input
futherGroupWhy
"Why or why not?"

ST14_01 [01]
Free text

[ST15] Text Input
precourseFreeText
"Why or why not?"

ST15_01 [01]
Free text

[ST16] Text Input
precoursehelpfulFreeText
"Why or why not?"

ST16_01 [01]
Free text

[ST17] Text Input
lernraumFreeText
"Why or why not?"

ST17_01 [01]
Free text

[ST19] Text Input
GroupText
"Why or why not?"

ST19_01 [01]
Free text

[ST20] Text Input
TutorialText
"Why or why not?"

ST20_01 [01]
Free text

[ST21] Text Input
selflearningText
"Why or why not?"

ST21_01 [01]
Free text

Figure A.2.3: Survey questions for students in test 2

[ST18] Text Input
lernraumhelpfulFreeText
"Why or why not?"

ST18_01 [01]
Free text

Section CR: Corona

[CR01] Selection
room
"Do you have a room in which you can learn alone?"

CR01 room
1 = Yes
2 = No
-9 = Not answered

[CR02] Selection
parents
"Do you currently live with your parents?"

CR02 parents
1 = Yes
2 = No
-9 = Not answered

[CR03] Selection
town
"Do you live in the same town or near where your university is?"

CR03 town
1 = Yes
2 = No
-9 = Not answered

[CR04] Text Input
KM
"How many Kilometers away do you live from your university town?"

CR04_01 [01]
Free text

[CR05] Scale (fully labeled)
influence
"Overall, how has the pandemic influenced your study?"

CR05_01 [No Description] 01
1 = Negative influence
2 = Slightly negative influnce
3 = No influence
4 = Slightly positive influence
5 = Positive influence
-9 = Not answered

[CR06] Text Input
influence why
"Why?"

CR06_01 [01]
Free text

Figure A.2.4: Survey questions for students in test 2

Conclusion

I confirm that I have independently prepared my thesis and all the required information which I took from research papers and all the other help I have mentioned in my thesis.

Chemnitz, 5.12.2021