T.C.
İSTANBUL UNIVERSITY
INSTITUTE OF GRADUATE STUDIES IN SCIENCE AND ENGINEERING

M.Sc. THESIS

DETERMINING THE PROGRAMMING ABILITY OF STUDENTS PRIOR TO PROGRAMMING

Amira ABUSHRIDA

Department of Computer Engineering

Computer Engineering Programme

M. Sc. Student transferred from Fatih University which has been closed

SUPERVISOR
Asst. Prof. Dr. Birol TİLKİ

June, 2016

İSTANBUL

DETERMINING THE PROGRAMMING ABILITY OF STUDENTS PRIOR TO Programming

by

Amira Abushrida

A thesis submitted to

the Graduate School of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

In

Computer Engineering

June 2016
Istanbul, Turkey

APPROVAL PAGE

This is to certify that I have read this thesis written by Amira ABUSHRIDA and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Computer Engineering.

_____

Asst. Prof. Birol TİLKİ
Thesis Supervisor

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

_____

Asst. Prof. Kadir TUFAN
Head of Department

Examining Committee Members

Asst. Prof. Kadir TUFAN _____

Asst. Prof. Oğuzhan ÖZTAŞ _____

Asst. Prof. Birol TİLKİ _____

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate School of Sciences and Engineering.

_____

Prof. Nurullah ARSLAN
Director

June 2016

DETERMINING THE PROGRAMMING ABILITY OF STUDENTS PRIOR TO Programming

Amira ABUSHRIDA

M.S. Thesis – Computer Engineering
June 2016

Thesis Supervisor: Asst. Prof. Birol TİLKİ

ABSTRACT

Nowadays, programming capability of some of the students who prefer computer departments are too weak. These students are suffering in programming courses in their departments. After graduation, they cannot work in a job related to programming. Student candidates can learn their programming ability by taking computer aptitude test before selecting the computer departments. Then, they may decide the computer departments according to test results.

In this study, current computer aptitude tests are investigated. Questions of mathematic, logical thinking, problem-solving and so on within the scope of the test are updated. We created a questions database and developed a website that uses the created question database and tests the programming ability. To measure the efficiency of the developed system; programming capability test is implemented on 80 students from computer engineering and computer programming departments, before taking their programming courses. Once students receive their programming courses, programming course notes and programming ability test notes are compared by using descriptive statistics, correlation, regression and independent t-test techniques.

Keywords: programming aptitude test, programming ability, programming education.

PROGRAMLAMA ÖNCESİ, ÖĞRENCİLERİN PROGRAMLAMA BECERİSİNİN BELİRLENMESİ

Amira ABUSHRIDA

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Haziran 2016

Tez Danışmanı: Yrd.Doç.Dr. Birol TİLKİ

ÖZ

Günümüzde, bilgisayar bölümlerini tercih eden bazı öğrencilerin programlama kabiliyeti çok zayıftır. Bu öğrenciler, bölümdeki programlama derslerde sıkıntı çekmekte ve mezun olduktan sonra programlama ile ilgili bir işte çalışamamaktadır. Öğrenci adayları, bilgisayar bölümlerini seçmeden önce programlama kabiliyeti testi yaparak durumlarını öğrenebilir; test sonuçlarına göre bilgisayar bölümlerini seçebilir.

Bu çalışmada, mevcut programlama kabiliyeti testleri incelendi. Testin kapsamına giren matematik, mantıksal düşünme, problem çözme vb. alanlardaki sorular güncellendi ve bir soru veri tabanı oluşturuldu. Bu veri tabanını kullanan ve programlama kabiliyetini test eden bir web sitesi geliştirildi. Geliştirilen sistemin verimliliğini ölçmek için; programlama kabiliyeti testi, bilgisayar mühendisliği ve bilgisayar programcılığı bölümünden 80 öğrenci üzerinde programlama dersini almadan önce uygulandı. Öğrenciler programlama dersini aldıktan sonra programlama ders notları ve programlama kabiliyeti testi notları; tanımlayıcı istatistik, korelasyon, regresyon ve bağımsız T-test teknikleri kullanılarak karşılaştırıldı.

Anahtar Kelimeler: programlama yetenek testi, programlama yeteneği, programlama eğitimi.

Dedicated to my family,

وَقُل رَّبِّ زِدْنِي عِلْمًا

(*and say, "My Lord, increase me in knowledge."*)

ACKNOWLEDGEMENT

Firstly, I would like to thank my supervisor, Asst. Prof. Birol TİLKİ for his support and motivation throughout the project. His encouragement has enabled me to develop a good understanding of the subject. His invaluable guidance and assistance has made the completion of this project possible.

I would like also to express my gratitude and appreciation to the other faculty members, Assoc.Prof. Cemil KUZEY and Assistant. Mustafa DURAN, for their valuable suggestions and comments.

I would also like to thank the Ministry of Higher Education in Libya for providing me with a postgraduate scholarship.

I offer my blessing and regards to my parents, sisters and brothers for their moral support during the difficult times.

My special thanks and respect to my soul partner, my husband, the real supporter during the hard times.

Finally I would like to thank all my friends and mates who have not been mentioned, but who have supported me to achieve my goal.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

AutoLEP	Automated Learning and Evaluating Programming
ACT	American College Test
ASP.NET	Active Server Pages .NET
CIF	Collaborative Instruction Framework
CSS	Cascading Style Sheets
CV	Curriculum Vitae
GAB	General Abilities Battery
GATB	General Aptitude Test Battery
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTS	Hybrid Testing System
IBM	International Business Machines
IDE	Integrated Development Environment
IELTS	International English Language Testing System
IQ	Intelligent Quotient
IT	Information Technology
LSI	Learning Style Inventory
MCQ	Multiple Choice Question
p	Probability
PAT	Programming Aptitude Test
SAT	Scholastic Aptitude or Assessment Test
TAB	Technical Abilities Battery
TOEFL	Test of English as a Foreign Language

TSI	Test of Sequential Instructions
TOEIC	Test of English for International Communication
SPSS	Statistical Packages for Social Science
VB.NET	Visual Basic.NET
α	Level of Significant

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Programming is a difficult skill which needs specialized knowledge, and generally, it involves some activities, few of which are, i) designing the code, ii) testing the code, iii) fixing the problem, iv) generating algorithms and v) maintaining the source code. Students improve the aforementioned skills during their studies in computer science degree (Pennington, N and Grabowski, B, 1990).

Students must have some capabilities which help them to learn programming easily, like, creativity, Mathematical skills, logical thinking and problem solving skills. A successful students after learning programming must be able to understand, write, design, analysis, and implement programming codes, additionally, they should be able to develop code lines further in order to add new features and solve problems. Succeeding an introductory programming course is a proof of learning key skills a programmer requires. These capabilities are an essential foundation which makes a student professional in programming is more easily in the future.

When the student's entry to introductory programming courses, it is known that the programming is difficult and complex to learn so a lot of them fail in these courses. The student eager to find and follow a set of instructions to success, and teachers help them. Although many of them find it hard to learn regardless of how much they try. Whereas, other students have a natural programming skills. Several of multinational researchers

detect that this problem is widely spread and known. Nonetheless, there is no agreement regarding the reasons. Some factors can be considered to be the cause of this phenomenon, some of which are:

- **Learner's skills:** some students have natural skills, meaning that they have the ability to understand programming easier than others.
- **Teaching methods:** there are different reasons for which teaching method can create an advantage for some students over others, for example, unsuitable teaching ways and materials, conflicts in teaching and learning.
- **Complexity of programming:** the theme might be excessively hard and complex. The basic programming may not have been prepared for students' requirements (Alan Blackwell, et al., 2002).

Having mentioned this, there have been speculations that by applying aptitude tests, the relationship between students' general abilities and their programming skills can be established. Creating this matrix helps in determining the programming abilities for every individual at early stage, hence, students can avoid failing the programming course.

The aptitude test is an old assessment that has been used for centuries to evaluate people's general abilities and specific skills. The first exam was an intelligence aptitude exam, although, it is significantly different from the current exams. Francis Galton who was start the history of aptitude exams. In the nineteenth century Francis Galton created the first exam. However, these exams were basically physical in comparison with current's tests which are intellectual tests. Due to this, Galton's exams was quickly disappeared. (Aptitude Tests; A Historical Perspective, 2004).

In 1960s there was a deficit in the supply of computer programmers. At the same time the computer industry began to evolve and spread widely. Therefore, the need for the significance software appeared. In this time the programming ability was not learned but was an innate, thus, it was difficult to find an expert programmer. It was believed that good programming skills were dependent on qualified individuals. Companies used aptitude tests as means of finding people with good programming skills. These tests focus on logical and

mathematical puzzles to looking for those who like intellectual challenges. In 1962, 80% of companies used aptitude tests when employing programmers. They used IBM Programmer Aptitude Test (PAT) which was considered the most important one (Nathan, E, 2016).

As a result, these tests developed by the time to be more accurate and effective. Now these tests include intellectual, physical and verbal, and ability (Aptitude Tests; A Historical Perspective, 2004).

There has been insufficient number of studies in the past few years for academic success in computer programming. Also it is always assumed that any bright student is able to do programming, however, this has been proven to be inaccurate assumptions based on classrooms reviews. A successful student in many other study materials might fail in learn to program (Byrne, Pat, and Gerry Lyons, 2001). Large number of people who failed in studying computer science has been surveyed, and main reason behind this failure was attributed to the lack of programming skills. Therefore, it has been decided to work on this dissertation to evaluate individual programming abilities for the students before they enter to computer science.

This research is focused on determining the programming capabilities for students before they enter to computer science by using aptitude test. By doing so, students will know their programming abilities before they decided to study it in order to avoid any failure in the future.

The main objectives of this research study are as follows:

- Measuring individual programming abilities for Computer Science students by applying aptitude test to measure their programming skills.
- Designing the aptitude test website on the internet to help students determining their programming skills.
- Analyzing data from this research by using SPSS statistics program in order to determine the relationship between the results from aptitude test notes and programming course notes.
- Highlighting key recommendations for future work in this research area.

- Finally, joining the results of this study to literature that used to lead future research into this area of evaluating programming capabilities.

1.2 RESEARCH QUESTIONS

The next three research questions are the foundation of the thesis:

- **Research question 1:** Is there a relationship between aptitude test and programming test?
- **Research question 2:** What is the relationship between engineering department and programming department in aptitude test?
- **Research question 3:** What is the relationship between engineering department and programming department in programming test?

1.3 ORGANIZATION OF THESIS

This thesis is divided into five main chapters:

CHAPTER 1: The introductory chapter introduces an overview of learning programming skills. Also explains the main problem behind this work which are problems faced by students in the introductory programming courses. This chapter also illustrates how to solve this problem by the aptitude tests. And highlights the objectives which the results of this work will be compared with in section 1.1. Then, it explains the research questions of this work in section 1.2. Lastly, the layout of the dissertation is given in section 1.3.

CHAPTER 2: Presents the literature review on the study of aptitude tests. This chapter includes a discussion on the abilities in section 2.1. After that, learning and evaluating programming and some skills required for programming in section 2.2. In addition to, a discussion on the types of testing and examination questions is introduced in

section 2.3. Lastly, the aptitude tests history, research conducted on them, different types of aptitude test, and their advantages and disadvantages in section 2.4.

CHAPTER 3: discusses the steps of aptitude test preparation and performed it on the students in section 3.1. Then, the types of computer programming aptitude test and hybrid test is presented in section 3.2. Also, analysis of data collected is discussed in sections 3.3. As well as, the research questions and hypothesis is introduced in section 3.4. Moreover, the online aptitude test is shown in section 3.5. Lastly, design the aptitude test website is discussed in section 3.6.

CHAPTER 4: illustrate the main outcomes from this work and explains the data analysis. In detail, starts by small introduction. Section 4.1 displays how the data collected. Then explains the Statistical Methods Used in data Analysis in section 4.2. While Section 4.3 introduces Research questions with hypothesis and answer these questions by statistical tests.

CHAPTER 5: presents the main outcomes made in Chapter 4. Also, this chapter highlights a few recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter displays a detailed discussion on literature available on the study of learning and evaluating programming skills. Also, this chapter includes discussion on aptitude tests. This chapter is structured as the following; it starts by a small introduction. Then, small discussion on the abilities in section 2.1, this section includes types of abilities in section 2.1.1 and the ability of grasping mathematics in section 2.1.2. After that, the programming and some skills required for programming in section 2.2, this section involves learning programming skills, evaluating programming skills and what programmer does? In sections 2.2.1, 2.2.2 and 2.2.3 respectively. While, section 2.3 contains the types of testing and examination questions, it starts by types of tests in section 2.3.1, and then types of examination questions in section 2.3.2. Lastly, section 2.4 discusses the aptitude tests which include the history of aptitude tests, the studies done on aptitude tests, types of aptitude tests and aptitude tests advantages and disadvantages in sections 2.4.1, 2.4.2, 2.4.3, and 2.4.4 respectively.

2.1 THE ABILITY

In general, the ability “is to be able to do something whether physical or mental”. Also it can be defined as “an obtained or natural ability or talent that enables to conduct a specific work or activity successfully” (What Is an Ability? Definition and Meaning, 2016).

Although there has been confusion between capacity and cleverness, sometimes a smart Person can't do anything practically. Therefore, it is important for people to know their best traits at early stage in order to select the right profession path (Aptitude Tests; A Historical Perspective, 2004).

Organizational performance is known by a ‘study of human behavior in the job place’. It is requires to understanding every individual actions. The humans’ behaviors are affected by their capacities. The next diagram displays the different factors affecting on the individuals’ actions (see figure 2.1) (Individual and Physical Abilities, 2016).

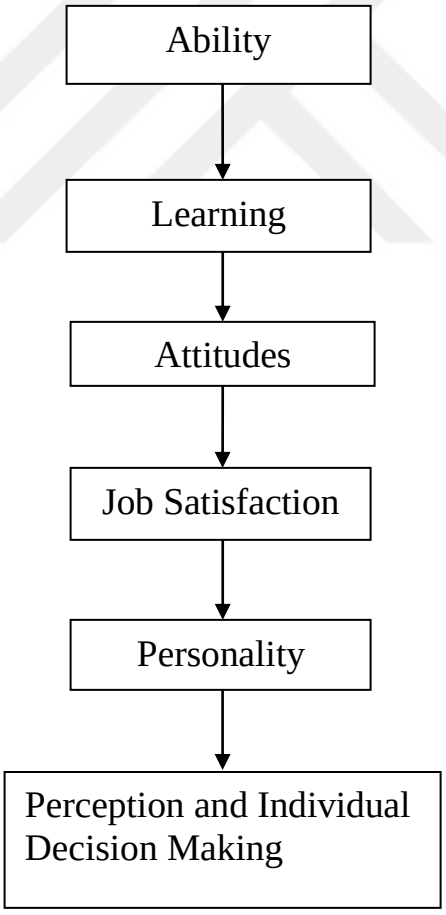


Figure 2.1 Behavior of Individuals.

2.1.1 Types of Ability

Ability is a possibility of the person to implement different activities. The person's capacities consist of various factors, below are the most important ones (Individual and Physical Abilities, 2016):

1) **Intellectual Abilities:**

It is necessary in order to implement intellectual tasks. Intelligent quotient (IQ) exams used to measure those tasks, and there are many tests that are designed to measure and evaluate intellectual abilities. It is supposed that there are some various factors which effects on such capacities, for example, mathematics, verbal comprehension, deductive reasoning, spatial visualization, memory.

2) **Physical Abilities:**

Some tasks require particular physical capacities to success or attributes that require management to determine the physical abilities for the employees.

3) **Job Fit ability:**

The performance of an employee is believed to be enhanced for the ones with higher abilities.

The following diagram displays different types of abilities (see figure 2.2) (Individual and Physical Abilities, 2016).

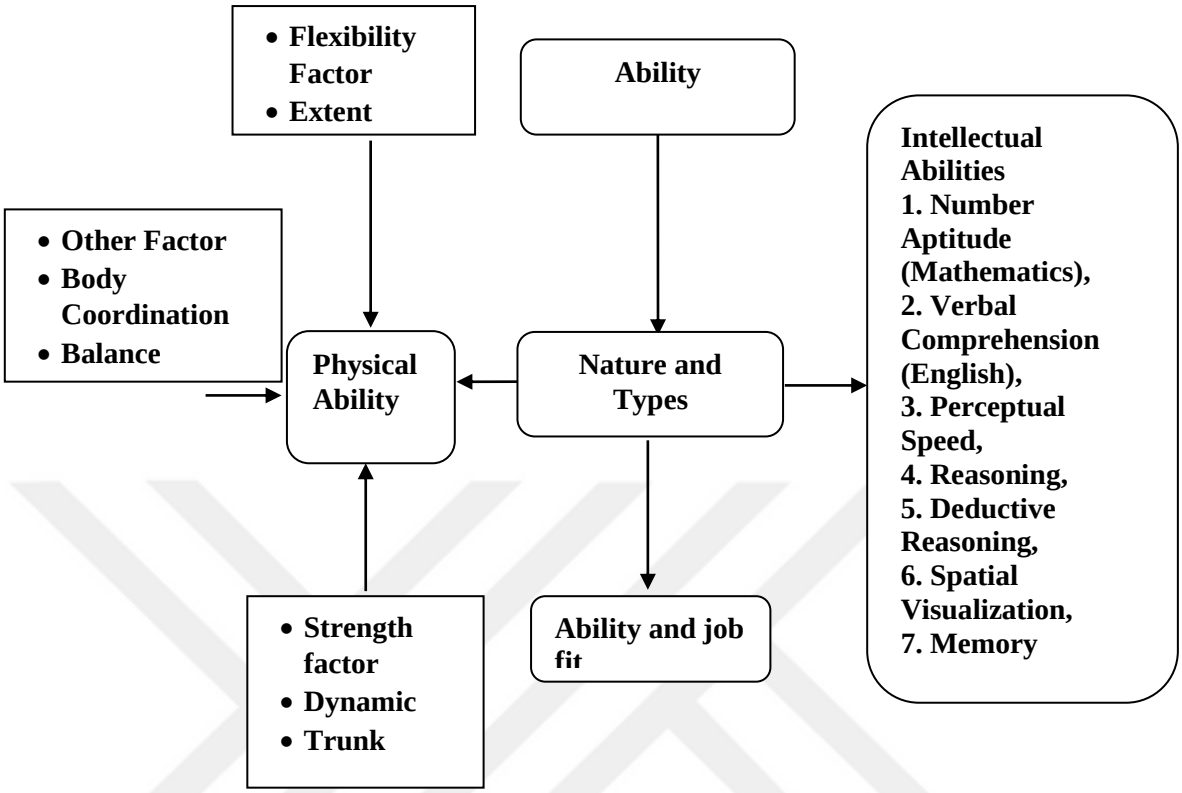


Figure 2.2 Various Types of Abilities.

2.1.2 Mathematical Capability

Early studies focused on determining the tools that can help to select suitable students to study computer science, by making correlation between students' performance and success in computer courses. Those tools, such as aptitude test, were undertaken by students with a large portion of those students having a math background. The relationships between mathematical capability and programming skills was accepted by a lot of researchers (Byrne, Pat, and Gerry Lyons, 2001).

Wilson, B.C. et al (2001) found that understanding math had strong correlations with the programming results (Wilson, B.C. and Shrock, S, 2001). Also, a study performed by McCoy, L.P et al (1988), it was said that the mathematical abilities help in students' success in programming courses (McCoy, L.P. and Burton, J.K, 1988). In another work which was done by McCoy, L.P. (1990), it was said that the students learn the math's

variables easily if they have experiences from programming (McCoy, L. P, 1990) and (Saeed Dehnadi, 2009). Moreover, Byrne, P. et al (2001) illustrated the correlation between programming aptitude and mathematics background (Byrne, Pat, and Gerry Lyons, 2001)

2.2 PROGRAMMING

Admittedly, programming is hard and complex subject, therefore, the programmers gain skills and experience. In addition to problem solving ability and mathematical background, there are some skills any programmer should possess, for instance; a programmer must be able to write a program, save it, test it, find the incorrect codes, then correct them and modify the program to be better. The main objective of programming is to write a program correctly. The process of writing codes requires multi-disciplinary knowledge, like; “logical, algorithm and application domain” (Computer Programmer, 2016).

Programming requires a person with a set of skills that forms a hierarchy (Kathryn D. Sloane and Marcia C. Linn, 1988). The programmer should be able to use any of them at any time. The students learning a hierarchy from lower level up to upper level (Bereiter, C and E. Ng, 1991).

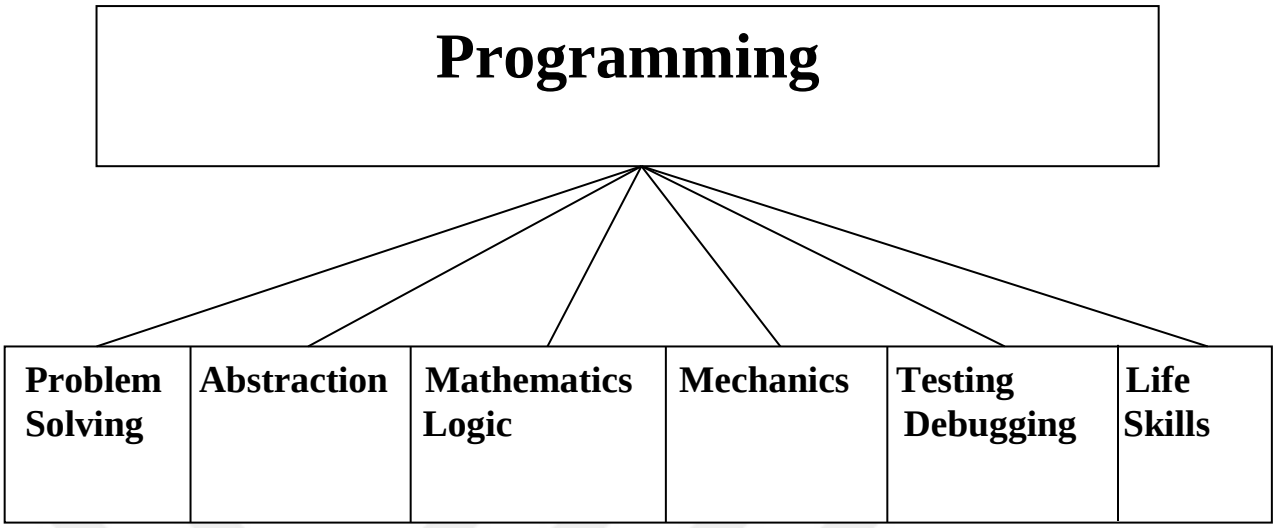


Figure 2.3 Skills required for programming.

Figure 2.3 presents some abilities needed for programming; the most important skill which is considered a basic programming ability that every programmer should have which is a problem solving. A programmer must have the ability to solve a problem innovate ways. Abstract thinking is an important element for learning programming. These expressions must be made in mathematical and logical ways. The mechanics information helps to making a program (modifying the program, find the results). The test skill helps to find the wrongs in the program and correct it.

Bennedsen, Jens et al 2006 conducted a study confirming that the abstraction capacity has a positive effect on programming capacity. Nonetheless, results presented by this study did not give the expected results, on the contrary, the results show no correlation between the above two abilities (Bennedsen, Jens and Michael E. Caspersen, 2006).

In addition, there are other skills that do not have a clear implication on students' performance on learning programming; these skills are named life skills. Programmers require life skills especially at the beginning of the programming course as it is considered to be a difficult subject. Therefore, many students struggle in the time period as a result of transition stage for students on their first year. These subjects should be the basic of learning programming, if they do not understand it clearly they will find problem in

programming, the students' needs to develop their life skills during their study (Fincher, S, 1999). Indeed, all the programming skills are important and the programing students need it to be proficient in programming.

There are many approaches for learning and evaluating programming ability. Each of these approaches has benefits and drawbacks. The next two sections will describe the ways of learning and evaluating programming skills.

2.2.1 Learning Programming Skills

Students in computer science are expected to master programming, which is one of many skills the students are required to learn. Most of students find programing difficult to learn, because students do not have the aptitude for programming, hence, they need to acquire some programming skills, such as problem solving and mathematical ability, to help them to learn programming. In fact, there are many factors at work helps students learn programming without difficulty. Examples of those factors include; the level of experience that the teachers and their students have, learning environment, difficulties and complexities faced by students.

In case students find learning programming difficult, one reason might be the ineffective ways of teaching programming by teachers. Therefore, many of current researches in the computing education focus on new and simplified methods to teach programming, and in that effort, many ways have been invented. For example, teaching programming to students through games and engagement sessions which has been proven to be an effective method.

Moreover, many studies have focused on researching ways of teaching programming to students, for instance, Mayer, R. E. (1981). This work investigated basic skills that students require in order to be able to learn programming more readily (Mayer, R. E, 1981).

Whereas, Pea, R. (1986) and Spohrer, J. & Soloway, E. (1986) looked at the mistakes that learners make (Pea, R, 1986) and (Spohrer, J., & Soloway, E, 1986), and Soloway, E.

et al (1982) identified the failures in acquiring programming skills (Soloway, E, et al., 1982), therefore, these studies are only applicable for teaching programming.

The LSI which is (Learning Style Inventory) is a tool applied for predict methods of improving the educational environment (Kolb, D, 1985). The LSI is a closed-loop cycle consists four methods to learning which are; concrete experience, reflective observation, abstract conceptualization and active experimentation shown in the Kolb's cycle (see Figure 2.4). Varying learning styles can be important during overall programming process. The LSI is effects on the programming process. Each person is examined through some questions, the Kolb's cycle can be shown by one of the quadrants, which are accommodator, diverge, assimilator and converge.

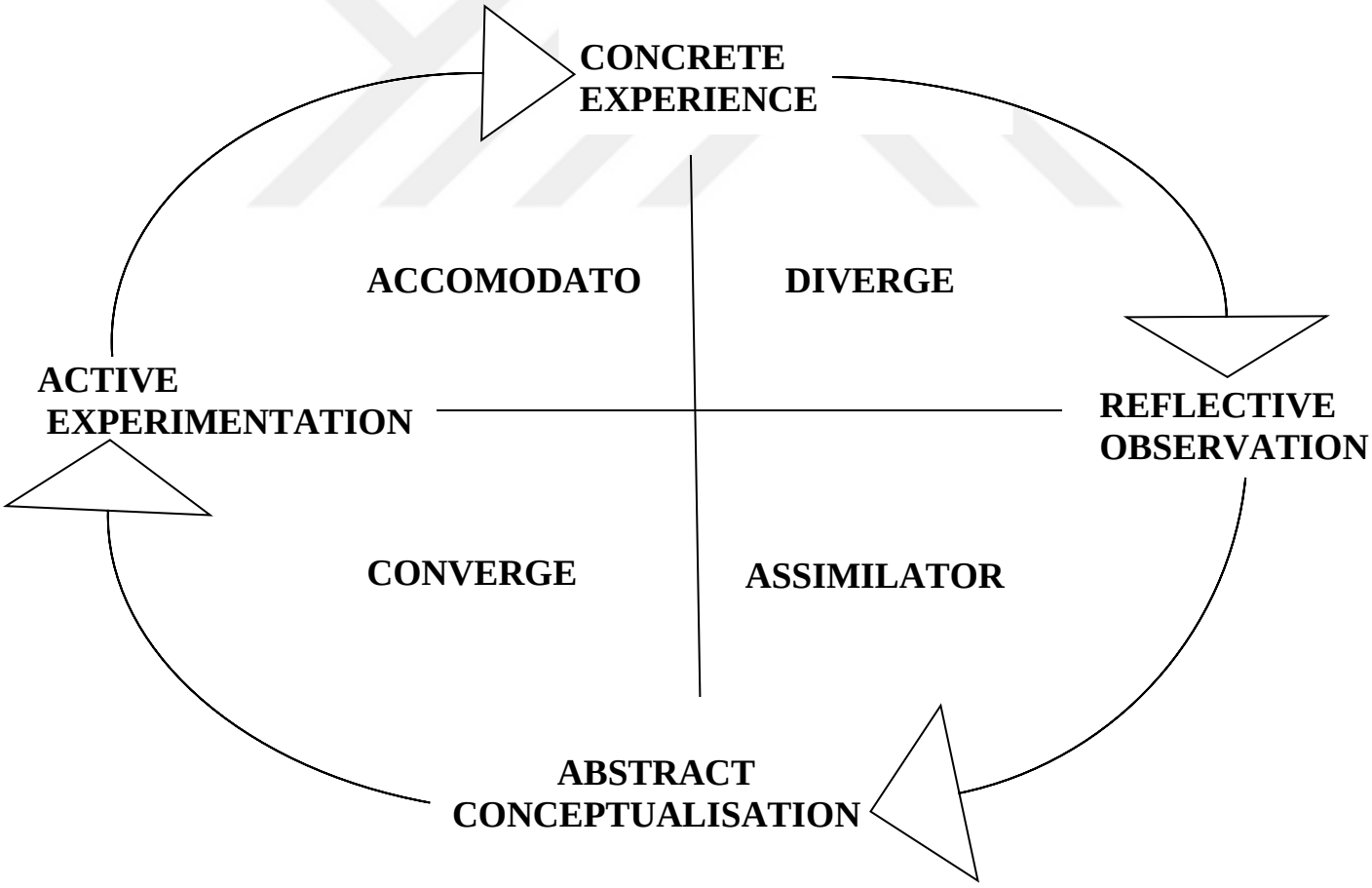


Figure 2.4 Learning styles (after kolb).

There are two cognitive factors that makes learn to programming complex and they consist of the following (Tony, J, 2002):

i. Learning style:

This allows students to acquire programming skills. Some students like to learn programming in specialized ways; some of them prefer to learn individually, whereas others like to learn in dynamic environment by discussing and engaging with other students. The teacher must be provide the suitable style to the students.

ii. Motivation:

Students always need to be motivated during their learning journey. Hence, they must be adopting the right motivation to find programming easy to learn.

Motivation is a substantial component for an effective education. Serrano, C. et al. (2014) presented an assessment of motivation by collaborative learning for 139 students, and four different methods were used: traditional lesson, cooperative education, cooperative education guided by CIF (an educational frame for cooperative education), and cooperative education supported by MoCAS and guided by CIF (a cooperative education tool) (Serrano, C. et al., 2014). Accordingly, these methods were used to get the results as the following, the results indicates to the motivation has significant correlation with MoCAS and CIF. As a result, that helps to improving the education operations, also they find, students learned collaboratively were less encouraged and motivated than students learned alone.

In (Tony, J, 2002) the author give some advices for learning programming, it is argued by the author that teaching programming should be in a second year of university. Programming should be taught by those who possess teaching abilities and have the technical capabilities to program, not by those who can only program but do not have the soft skills to teach. Programming courses should be flexible and suitable to help students in different ways, computer departments must recognize that there are some programming subjects are too hard to learn and providing support to the students by all means to make their learning journey as pleasant and enjoyable as possible.

2.2.2 Evaluating Programming Skills

There are many multi-national studies, which were searching for factors connected to programming success in computer science. These factors include gender, mathematical abilities, abstraction ability, problem solving ability, cognitive abilities, experiences prior courses, learning style, comfort level, difficulty of exam, mental model, and they starting with aptitude exams to Pick up appropriate persons in order to place them in the software work. As a conclusion, programming difficulty was found to be a global issue.

McCracken et al. (2001) tested the programming competency students and looked at the assessment to measure students' programming ability in the end of first year (McCracken et al., 2001). In order to find methods for evaluating students and develop these types of assessments to explore if the students can program, they apply this test for 216 students from four universities. Instructors found the students' skills is not enough for learning how to program. Therefore, they focused on mastery the basic programming skills. The results indicated that there are many students can't write a simple program at the end of their introductory courses. They explain that the programming skills and the knowledge for students are too weak.

In addition Lister, R. et al (2004) made another multi-national study after the work done by McCracken et al. (2001). The study investigated students' ability to complete a code and expect the results from that code, a lot of students could not achieve that task. An argument made in order to justify the failure of learning how to programming, which is the lack of the problem solving skill (Lister, R. et al., 2004). On the contrary, Simon, et al (2006) said that we can't determine the programming efficiency, thus, it is not possible find connection between performance on tasks and programming aptitude (Simon. et al., 2006).

All teachers that teach programming always think about how to determine an individual programming ability, and it is done in different ways. Chamillard, A. T and Jay J. Joiner (2001) evaluated students' aptitude programming in an introductory computer science course by using lab practice (Chamillard, A. T and Jay K. Joiner, 2001). This work concluded that lab practice success to provide an effective method for evaluating each student's programming skills, and the approach is general enough to be used in different

types of courses. Whereas, Alahakoon, P. M. et al (2012) used a Hybrid Testing System (HTS) that was prepared and conducted in the University of Peradeniya, the (HTS) which required from the students to answer a paper pencil test which is multiple choice questions. (Alahakoon, P. M. K et al., 2012). Also hybrid test provide means of examining the abilities in a complex details. Moreover, the benefits on the examiners are the comfort ways in designing the tests, putting questions for different levels, reduce the possibility of copying the test, and did not take a time in marking. As the results, they said that the (HTS) is very successful compared to the (ICT). In a study conducted by Flecher, Stephen H. (1984) to determine the cognitive abilities needed for success in computer programming. Forty-one students from introductory computer class in Los Angeles University (Fletcher, Stephen H, 1984). Firstly, the general cognitive abilities had been determined, then the results were compared with the results of a test of computer programming skills. The results presented that the computer programming has high correlation with a cognitive abilities. Also it is pointed out that the outcomes of this study were identical to the results from a study done by Shaha (1983).

Furthermore, some factors should be taken into account when choosing the best students for programming courses. In computer courses, approximately 72% of students are males, it seems that there are some factors preventing females from enrolling in computer science classes. There are no established reasons why females do not prefer to study in computer science, however, there are speculations related to the nature of society. However, an investigation indicated that female students performed well which can encourage them to stay and study computer science. Despite this study, Byrne, Pat et al (2001) explored factors that help students to succeed in programming (Byrne, Pat, and Gerry Lyons, 2001). The authors tested the correlation between students' outcomes in a first year of computer science and the aptitude skills of gender, previous programming knowledge and educational style. The study did not show any dominant features in programming as the results showed that the programming ability has positive correlation with a mathematics background. Also the authors found few students with previous programming experiences who performed better that those without any experience. Programming requires some experience as well as mathematical background. Furthermore, they found that females achieved higher scores as males which contradicting the results

found in previous studies. These factors affect teaching and learning, as the study done by Besie, C. et al (2003) which looked at some data, such as age and sex, and how they affect success in programming courses, and they proved that the data had no connection with success in programming courses (Besie, C. et al., 2003).

There are several studies examined the success in programming by some factors such as; (Wilson, B.C. and Shrock, S, 2001) and (Ramalingam, V and Wiedenbeck, S, 1998), both studies used Computer Programming Self-Efficacy Scale test. They applied the test on a set of students, they examined some factors that helps to predict success in exam before learning programming. They found that there are some important factors including comfort level, mathematical skills, the luck, and capability. Therefore, they found that some factors such as comfort level and mathematical skills had strong correlations with the exam, whereas luck had not correlation. Whereas, (Rountree, N.et al., 2004) in two studies, they argue that the students have high possibility to success those who are predicting to get an 'A' score in the exam. They considered that student's expectations may effect on their results.

There are studies that test the mental model, for instance, Saeed Dehnadi (2006). This study suggests that it can expect the success in the computer science course. The idea of this study was that different people bring several of knowledge in any new educational operation, and proved that how can addresses the problem in a various methods based on their mental before the students have had any contact with programming. The results from this study demonstrated that in order to examine success prediction the mental models is a very important to build tools to determine the computer programming skills. Whereas, form more than 25 years ago researchers had an active research for expecting the students' success of in computer science courses. Dehnadi and Bornat (2006) found test for programming skills to separate programming sheep from non-programming goats, which picked up those students who have an opportunity to success in programming and refuses those who have no opportunity. Depending on this theory, Caspersen, Michael E. et al. (2007) repeated their test in order to confirm this theory and generalize the results, they tested a connection between mental thinking of the students and the results of the students in an introductory computer science courses. But this test fails to predict students' success

in programming course. Also they criticize Dehnadi and Bornat's exams and put other tools to help them to design good test.

(Wang, Tiantian. et al., 2011) tried to develop and use an automated educational and evaluating system, to help the beginner programmers to learn programming skills easily. Furthermore, this can be better for the students' educational, capability of self-learning, the teaching methods can be improved and minimize the work of the teaching staff. AutoLEP (Automated Learning and Evaluating Programming) has been applied at Harbin Institute of Technology in 2004. Also, it was applied in (C) programming language courses. As a result, the AutoLEP had positive outcomes from students and teachers because of including it into the introductory programming course.

Automated Learning and Evaluating Programming is presents in (Figure 2.5). AutoLEP consists of the next three main parts.

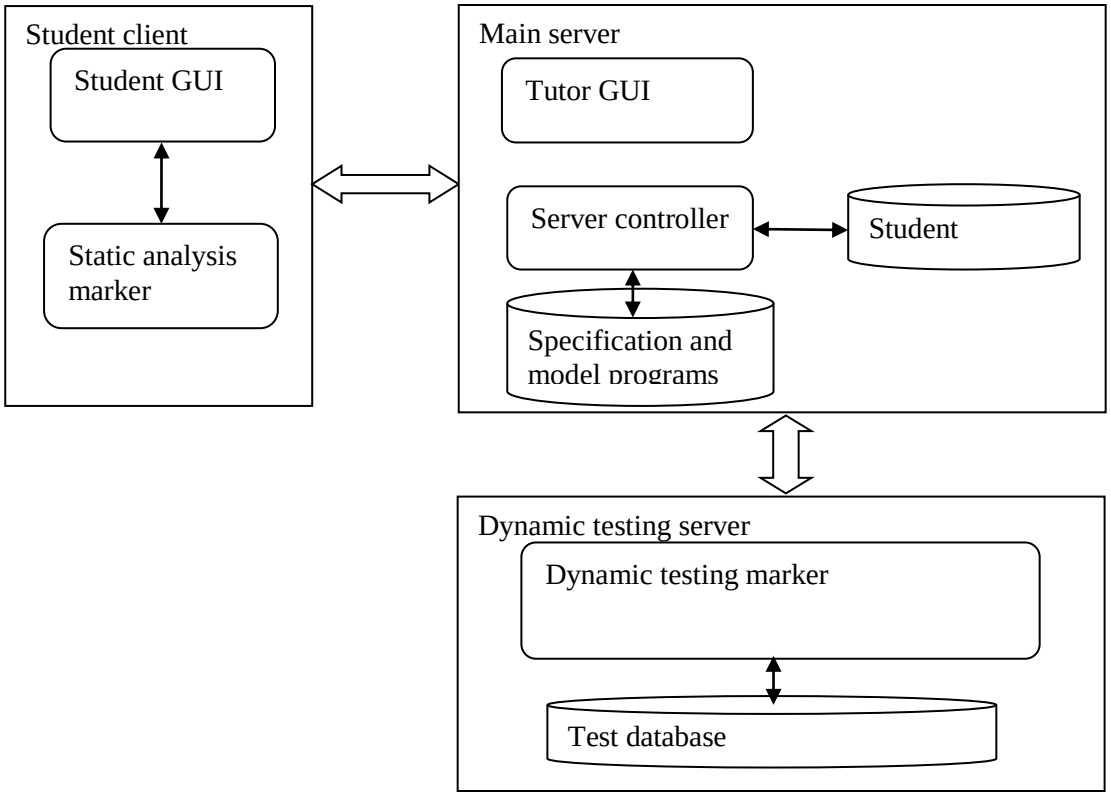


Figure 2.5 The Architecture of AutoLEP.

I. Student clients part:

It is consist a (GUI) for the students and the analysis marker. The GUI provides to the students to submit their programs to the system to be marked, and provides feedback to the students.

II. Dynamic testing server part:

The testing server consist the dynamic examine marker and an exam database. The dynamic marker used to mark the students tests.

III. Main server part:

This part is the main part which consist the tutor GUI, a server controller, a student database, and a specification and model program database. The tutor GUI provide some features for the teachers to prepare new tasks, present different results, and check the students' needs. The server controller is the control center. Moreover, it is connected with the student client's part and the testing server part.

2.2.3 What Does the Programmer Do

There is an ongoing discussion on whether writing a program is a talent or a programming discipline. The programmer does not require to have any certification in order to be called "programmers". Although, Describe themselves as a "programming engineer" without academic license, it is illegal in a lot of countries. (Computer Programmer, 2016). Therefore, a computer programmer needs a certification in computer science to learn different programming languages in the university. So they must be able to do different tasks, such as start by designing flow charts then writing the code after that testing the code to find any errors if there are any errors they should be correct them to be free of errors through a process known as debugging and run correctly. Also saving the program's health is the responsibility of the programmers. Programmers spend most of their time at writing codes, writing diagrams and thinking about the structure. Therefore, some works in writing codes such as Mobil phone applications takes from programmers few days. In contrast, other works like computer operating systems might take few months

to over a year. Whereas, some programmers developed themselves by moved to become computer systems analysts or software developers. Therefore, this moved requires problem solving skills, analytical skills and concentration for writing long codes. The Bureau of Statistics predicts to increase this career field an 8% in 2012-2022 (What Does a Computer Programmer Do? Careers and Degrees, 2016).

A computer programmer can work anywhere in the world for writing codes anywhere in office or able to work from home (What Does a Computer Programmer Do? Careers and Degrees, 2016). Despite, most programmers work in offices and they spend a lot of their time in front of a computers, however may be needed to work overtime to solve technical problems or meet deadlines (Computer Programmer, 2016).

2.3 TYPES OF TESTING AND EXAMINATION QUESTIONS

2.3.1 Types of Tests

Tests are methods to verify your knowledge and evaluate your learning and those methods are used by all educational institutions. According to some studies, tests have many advantages; one of them is make you learn and remember more than what you have without it.

There are various types of tests, each of them has different target and style (Types of Tests, 2016):

1) Diagnostic Tests:

These tests are used to diagnose the amount of students' knowledge and what they know as well as helping a teacher to know what students need in a class.

2) Placement Tests:

These tests are used to place students in the suitable class. Like what happening in language schools these tests used to verify of student's language level then placed the students in the right class to suit what they need.

3) Proficiency Tests:

These tests check learner levels in relation to general standards. Such as the TOEFL and IELTS exams, which are compulsory for foreign speakers in order to admission to English universities. In addition, the TOEIC checks students' knowledge, as a condition for employment.

4) Internal Tests:

Internal tests are those given by the institution like a final exam where the learner is taking the course.

5) External Tests:

External tests are those given by an outside body. For instance, the TOEFL, TOEIC, IELTS, SAT, ACT and GMAT etc. These exams for admission to university and employment.

These are the basic tests, as well as there are many others types of tests.

2.3.2 Types of Examination Questions

Exams are a popular assessments in schools and universities, examinations are structured in different ways, each type needs a different strategy in putting the question and answer it, there are many types of examination questions including (Different Types of Tests, 2016), (Exam Questions, 2016):

1) Multiple choice Question:

This type consist of a question with various answers (choices), which is one correct answer and the others are incorrect answers. So, students choose the correct answer by putting circle on the related number. Students can answer on this type of questions quickly, because of the answer is existing on the exams paper not from student's mind.

2) True/False Question:

This type consist only one statement. It has only two potential of answers (true or false). That means there is 50% chance of guessing the right answer.

3) Matching Question:

Students should matching the statements with one of the options in the question. They are quick and simple to create and scoring it by the teachers. Whereas, the students need to along time to answer them.

4) Short answer Question:

This type consist of a quick brief that requires from the students to write answer from one or two words to a short sentences. They are applied to examine the key words. Short answer questions difficult and take time for the students to guess the answer.

5) Essay Question:

Essay questions requires written responses, it is different in length from some of paragraphs to many pages. Students can explain their understanding and show creativity, but it takes a lot of time to answer it.

6) Oral exams Question:

This type the students to answer directly to the teacher’s questions. These tests are common in language tests. Also, it can be applied to evaluate students. There are some cons

to this type: the time in this exam takes ten to fifteen minutes per student and the problem of record-saving.

7) Computational Question:

This type the students should be implement calculations duo to solve the problem. This type of question used to evaluate student's memory to apply some mathematical methods to solve questions.

8) Open Book Exams:

This type is used when a student needs to see the charts or other materials. But, students should be studied for an open book test as they did for a closed book test. Because of they have no time to search in details at the exam time.

2.4 THE APTITUDE TEST

The definition of the word 'aptitude' today is "a natural ability to do something" (Aptitude Tests; A Historical Perspective**,** 2004). Also, there is another definition for the aptitude word is "an obtained or natural skill (usually determined by aptitude exams), for educational in a particular area, and is reflected in current performance which is predicted to be better over time" (What Is Aptitude? Definition and Meaning, 2016). Whereas, Aptitude test is defined as "an employee chosen an exam like, IQ exam in which the candidate's innate skills and possibility for success (and not his or her information) is ascertained" (What Is Aptitude Test? Definition and Meaning, 2016). As well as, it is defined as "an aptitude exam is a test applied to identify a suitable persons to achieve in a specific work" (Aptitude Test Definition, 2016).

Some people acquires the knowledge or skills easily, others acquires the knowledge or skills difficulty and takes more time, because of all people have different aptitudes or skills acquisition. There has been confusion between aptitude tests and achievement tests, there is differences between them. In general aptitude exams are applied to expect the

achievement in a course of study or career, that mean applied to plan for future education and job (job guidance). Whereas, success in exams are prepared to determine how much a person learned in courses (Aptitude Assessment, 2013).

2.4.1 The History of Aptitude Test

The aptitude exam used from many years ago to evaluate specific skills. Aptitude tests started with Sir Francis Galton in the early of nineteenth century. Whereas, in early 20th century, the Stanford-Binet Intelligence measure was published. This scale was used to help the army testing mental capability and it was divided into two; Alpha test was used for those who were speak English well, and the Beta test was used for those who did not speak English well. These were the basic tests used in the USA. In 1920's these tests were changed in (1926) to be the first SAT test. In 1905 Alfred Binet who is a French psychologist, started to design an exam of intelligence, named by the Stanford-Binet Intelligence exam. By World War I, aptitude Army Mental exams were performed to assign U.S works in the war (Dan, F, 2009).

During the First World War, Robert Yerkes had tested U.S. the army; one of Yerkes' assistants was Carl Brigham, who taught at Princeton. After the war, Brigham began using the test by making it more difficult to be used in a college admission. It was first administered in 1926, and in 1933, James Bryant Conant established to begin a new scholarship for students who had aptitude. Henry Chauncey found the tests used to evaluate candidates for these scholarships. Chauncey met Carl Brigham, Brigham advised him to use the SAT, because he thought it measured pure intelligence regardless of the quality of the high school education knowledge. In 1938, Chauncey began applying the SAT as a one test, which was used only for employees. In 1942, duo to the Second World War II, the SAT became the exam for all employees. In 1948, the SAT test became the university admissions exam for students (PBS, 1995).

In 1930's, in the USA many tests had been designed to determine the aptitude of people for employment, at that time these tests were more general, and after that tests became more specific. In 1947, these exams called as the GATB (General Aptitude Test Battery). After that, in 1983, these exams were changed, which was used to the 1990's for

employment. The aptitude exams were divided into three groups; the GAB (General Abilities Battery) for recruitment staff up to junior administration level, TAB (technical Abilities Battery) for those recruited to technical administration and DAT for school leavers and guide them for right career. These tests were used in the USA and Europe (Aptitude Tests; A Historical Perspective**,** 2004) and (Aptitude Assessment, 2013).

The ACT and the SAT become the most important in the college-admissions, also for some schools. The SAT (Scholastic Aptitude Test or Scholastic Assessment Test) started in 1926 which was designed before ACT, and in 1930 its familiar form was assumed. The test was accepted by a lot of universities in the end of the World War II. It remained unchanged until 2005. Whereas, in 1959 Everett Franklin Lindquist designed the ACT (American College Testing). The exam included math questions and English language skills, the ACT evaluate students on their learning information. Despite, some countries prefer ACT others prefer SAT. The SAT is a logical testing, while the ACT is a test of knowledge. (Dan, F, 2009).

2.4.2 The Studies of Aptitude Test

Where he began development of computer technology in 1960s, the need of programming courses has increased. Despite, a large proportion of the applicants fail in these courses for some reasons, therefore the researchers found factors which leads to success in these courses.

A study conducted by (Reinstedt, R. N. et al., 1964), they were worried to make any programming aptitude test which help the management to determine the professional programmers. Therefore, they tested the relationship between the job performance for employees and some measures including cognitive capabilities and CV. Thus, they used a battery tests PAT (Programmer Aptitude Test used by IBM) and TSI on a set of programmers. Despite they found a strong correlation between programming aptitude test and ranking of performance, the correlation between Test of Sequential Instructions and ranking of performance was not significant (Saeed Dehnadi, 2009).

Some studies applied aptitude exam to expect the best employees to be selected to learn programming. In (Cross, E, 1970)'s study, he said that the aptitude tests had not been effective in expecting in the programming courses, which also the results have not been satisfactory. Also, a study by (Mayer, D.B, 1964) criticized the aptitude exams and said it is a traditional tests to choose the employees.

He confirmed that the interviews for software employees instead of a paper pencil exam to Identifying the programming ability, to detect personal attributes are preferred. As a result, these interviews gave him 60-70% effectiveness in selection (Saeed Dehnadi, 2009). Further, (Mayer, D.B and Stalnaker, A.W, 1968) mentioned that the PAT (programming aptitude tests) measures some programming ability and TSI (Test of Sequential Instructions) measures other abilities, nevertheless, there is no test that can measure the whole programming ability of individuals. In 1966, approximately 350 Canadian and American companies applied PAT-IBM to sort employees, whereas the results showed that the exam has low correlation with the job performance of programmers.

Determine variables that expect programming skills that assist instructors to choose suitable students. In (Huoman, J, 1986) study, he invented a computer aptitude exam. Then, evaluated it in the introductory programming course by (Tukiainen, M and Mönkkönen, E, 2002) they presented this study to assess how expectation of the new programming aptitude testing was. Computer aptitude testing has good results with the education form many years ago. The best test has been designed was the IBM Programmer Aptitude Test (PAT) which is applied to expect the computer skills. The exam include of three questions. The first question was provide the series of numbers to choose the following number, the second question was to choose identical in numbers, and the third question about mathematical puzzles. They found between the Huoman exam and the final exam a strong correlation firstly. Then, it give no success in the programming test when individual programming was examined (Saeed Dehnadi, 2009).

(Wolfe, J.M, 1971) developed the aptitude tests to overcome such problems, his exams used a serious of complex numbers to select the candidates, in order to test if they can follow instructions. (Winrow, B, 1999) continued to develop the battery test, which

included many tasks to measure logical ability, numerical ability, ability to follow instructions, e.g. (I-PAT and PAAT).

In some countries the admission from universities requires aptitude testing, the SAT tests are used In the U.S. in spite of some universities contemplating to cancel it (Organ, D.W, 2001). Because of these exams were not perfect, it is said that the exams represent about 20% of the differences in the university grade (Ransdell, S, 2001).

2.4.3 Types of Aptitude Test

An aptitude test evaluates the talent/ability/potential to perform tasks, with no prior knowledge or training. Aptitude exams are used to evaluate logical thinking and problem solving. Nowadays, there are at least 5000 aptitude tests, some of them tested one type of ability for instance numeric ability, verbal ability etc. Whereas others tested different types of ability (Aptitude Tests – Introduction, 2016) and (What Is an Aptitude Test, 2016).

Essentially, there are four types of aptitude tests (Can Aptitude Tests Really Predict Employee Success?, 2016):

- **Skills Tests:** Usually they are easy tasks, for example, typing quickly on the keyboard or data entry and it can be easily measured.
- **Knowledge Tests:** These tests are used to measure what candidates know, to ensure that a candidate has the high level of science information.
- **Ability Tests:** ability tests are designed to determine intellectual skills.
- **Personality Tests:** These tests are used to measure the candidate's ability on certain personality traits. Which are: emotional stability, conscientiousness, and extroversion.

There are several types of aptitude test questions (see figure 2.6) and they are classified as the following (Aptitude Tests – Introduction, 2016) and (What Is an Aptitude Test, 2016):

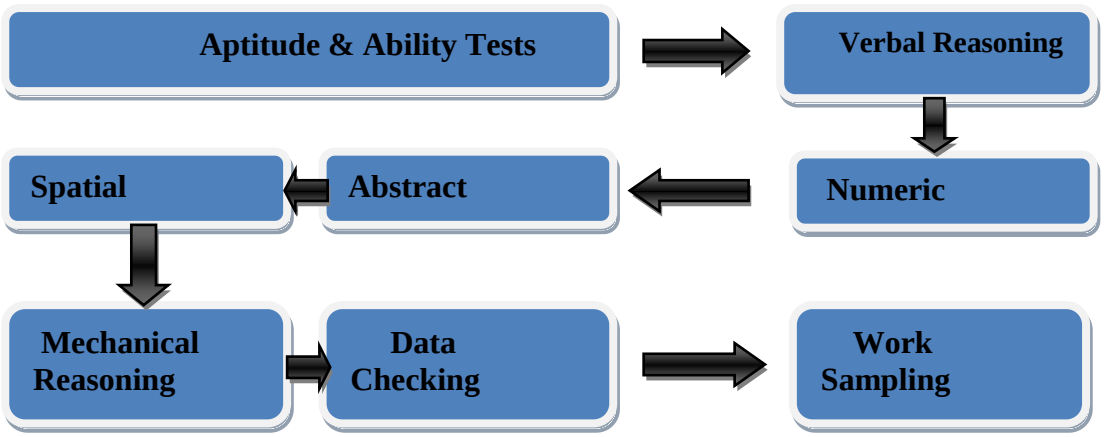


Figure 2.6 Types of aptitude test questions.

1. **Verbal Tests:**

This test used to determine the applicant skills including spelling, grammar, analyzing a text, ability to understand and follow instructions. Verbal testing is typically found in the form of text analysis and linguistically based questions.

2. **Numerical Reasoning Tests:**

Numerical reasoning tests designed to determine the applicant ability including basic arithmetic, number sequences and simple mathematics. This type of tests is the most common one. In this type usually the question shows with charts and graphs that need to be explained.

3. **Abstract Reasoning Tests:**

Abstract reasoning tests measures the applicant ability to determine the solution by identifying the logic of a pattern. This type includes questions which ask you to draw logical conclusions based on information through shapes, patterns, words and pictures, this type can found in all aptitude exams.

4. Spatial Tests:

This tests Measures individuals' ability to manipulate shapes in two or three dimensions by using images and diagrams depicting mirror reflections. These questions usually not found in general aptitude tests. Spatial tests are usually for those who work in space such as astronauts.

5. Mechanical Reasoning Tests:

Mechanical reasoning tests are designed to evaluate knowledge of physical and mechanical and measure how intelligent and fast the applicant can solve problems of mechanical faults. The questions in this test presents by picture. These tests used to select set of jobs such as police forces, fire services, technical and engineering occupations.

6. Data Checking Tests / Error Checking Tests:

Data checking tests used to measure how quickly and accurately the applicant can discover the errors in data, to select them for data input jobs. Because of, it is indicates to the attention to detail and error detecting skills.

7. Work Sample Tests:

Work sample tests consist of a sample of the work that you will be expected do. This type of test can be wide ranging and used for different employment.

Furthermore, there are other types of tests for example; Inductive exams, Deductive exams, Concentration exams, Diagrammatic exams, Sensory exams Etc.

2.4.4 Aptitude Test Advantages and Disadvantages

- **Aptitude Test Advantages:**

There are several advantages for aptitude tests which are (Can Aptitude Tests Really Predict Employee Success?, 2016) and (Advantages and Disadvantages of Conducting Tests for Selection of Employees, 2014):

1) There are several studies prove that the aptitude exams are accurate in identify the possibility for success.
2) Aptitude tests provide a uniform basis that means all candidates examining by same test for comparing the performance of applicants.
3) Aptitude tests provide proper assessment, therefore, used to determine the suitability of candidates for various jobs whether it is for employment or in promotion.
4) Aptitude tests help in selecting the better individuals and placing them in suitable jobs, as a result they will improve their efficiency and job satisfaction.
5) Interviews alone are not enough to determine candidate's capabilities. In such cases, aptitude tests are accurate to be implemented in evaluating a person's strengths and weaknesses.
6) Aptitude tests are an opportunity to examine the candidates' skills and to determine their skills and what they are good at.
7) Proper selection of candidates will reduce labor turnover for the employee. Because, when a person gets a suitable job he /she would not leave it. As well as, for the students when the universities select the suitable students they will succeed and do not leave their discipline.

- **Aptitude Test Disadvantages:**

There are several disadvantages for aptitude tests which are (Can Aptitude Tests Really Predict Employee Success?, 2016) and (Advantages and Disadvantages of Conducting Tests for Selection of Employees, 2014):

1) Aptitude tests might be costly to conduct for both developing them and administrating.
2) These tests are time-consuming to finish the selection process, acquire the time is very important for the company.
3) Job descriptions and requirements change and the aptitude tests must change with a company's needs.

4) Aptitude tests may make the job applicant feel fear and stressed. Some persons may not submit to the tests for this reason. The companies may be deprived of the services of such personnel who are afraid to appear for the tests.
5) These tests are unreliable because of maybe get the incorrect results in some cases, the skill of a candidate may not be judged correctly with the tests.

CHAPTER 3

METHODOLOGY

Chapter 3 explains the steps taken to prepare the aptitude test used in this work. Then, ways to perform it on students. Therefore, this chapter is structured as the following; it is starts by small introduction. Then, the research design with aptitude test preparation steps in section 3.1, this section includes the test, the scoring and the variable in sub-sections 3.1.1, 3.1.2 and 3.1.3 respectively. While, the section 3.2 contains of types of computer programming aptitude test including types of hybrid test in section 3.2.1. After that, a discussion on the analysis of data collected by several statistical methods is made in section 3.3, followed by research questions and hypothesis in section 3.4. Next, presents the online aptitude tests in section 3.5, and finally, design the aptitude test website is discussed in section 3.6.

3.1 RESEARCH DESIGN

Programming is one of many skills that computer science students are expected to master. A substantial amount of research has been conducted in order to identify variables that are predictors of students' success aiming for a university degree. The variables represent scientific factors (e.g. math score). However, these variables only account for a fraction of the variation of students' performance (McCracken et al, 2001).

The objective of this study is to determine the programming ability for the student because of the following reason; when students in the first year of the university and they desire to enter to a computer science, whereas they don't know whether they have programming ability and can succeed in programming or not, so this study was conducted to determine the students' individual programming ability before learning programming to avoid the student's failure and loss a year in the university.

In regards of the aforementioned purpose, this study was conducted for the first year students, Department of Computer Science, Fatih University Istanbul. To examine the students' programming abilities prior to their learning programming by determining students other capabilities associated with programming skill (Example: Mathematics, logical thinking and problem solving…Etc.). In this study the aptitude test was used, and it was conducted on the students in 2015-2016. The aptitude test was applied after 8 weeks from the beginning of the semester for two departments in the computer science. In the end of the semester the exams was performed from the university. These exams including the programming test which we was used in this study. The present study generated a set of students' scores, and the comparison was made between the scores, to investigate the relationship between them.

3.1.1 The Test

The general aptitude test was administered on 80 students enrolled in the first year computer science from two departments, which are 40 students from engineering department and the rest from programming department and all students are from Fatih University 2015/2016. The aptitude test was a paper pencil test designed to measure students' programming abilities.

There are several types of computer programming aptitude test which will be discussed in the next section, the hybrid test was chosen to be used in the current study, the hybrid test consists of five testing abilities, and also, it will be discussed in the next section. The aptitude test was prepared from fifteen questions and it was divided into the following: four numerical reasoning questions, four logical reasoning questions, three problem solving questions, two pattern recognition questions and two questions requiring the ability to

follow complex procedure. At the beginning the questions were prepared as Multiple Choice Questions (MCQ), after that it was thought that some students might not think about the questions and they would just select any answer from the choices randomly. Therefore, to ensure the credibility of the students' answers, we decided to cancel the choices from the questions (See appendix A and appendix B).

After the aptitude test preparing stage, the test was conducted in fall semester of 2014-2015 on 80 students and the exam lasted for thirty minutes. The students from two departments which was divided into three groups, the number of students in the first group was forty, which was from programming department testing was applied by Turkish language, and the second group was forty students from engineering department which is divided into two groups the first one testing was conducted by English language on twenty-nine students, and the second group testing was conducted by Turkish language on eleven students.

The current study provides a set of students' scores, the data was analyzed by using SPSS program. Several analyses were conducted on the collected data, to check the relationship between the aptitude test and the programming test. Finally, the aptitude test was designed as web-based site on the internet.

3.1.2 The Scoring

The general aptitude test has fifteen questions, but unfortunately two questions from the test which are from pattern recognition were unclear in the printing. Although most of students answered them correctly, others didn't understand them, then, it was decided to cancel them. As a result, the aptitude test has thirteen questions. If students answer the question right, they get one mark, otherwise, they get zero. Therefore, the thirteen questions of the aptitude test have a total of thirteen scores.

The equation was used to make the score in percent as the following Eq. (3.1):

$$\text{The score in percentage} = \frac{\text{The Score} \times 100}{13} \tag{3.1}$$

3.1.3 The Variables

In the current study there was two variables, the aptitude test grades are the independent variable, whereas the programming test grades are the dependent variable.

3.2 TYPES OF COMPUTER PROGRAMMING APTITUDE TESTS

Basically there are three types of computer programming aptitude test (Why Programmer Aptitude Test Reject Programmers for Jobs?, 2004):

1) A Standard Battery of Test:

This type of aptitude test can measure the mental abilities of people like logical reasoning, non-verbal reasoning and numerical reasoning. This type is suitable for technical computing jobs, because of this type measured the mathematical and logical abilities.

2) Hybrid Test:

This type of aptitude test composed from different testing ability such as non-verbal reasoning (pattern recognition), problem solving, logical reasoning, numerical reasoning, ability to follow complex procedure. Some of these tests do not require to programming skills. However, they are included because it is helps to determine the programming skills.

3) Programming Simulation Test:

This type of aptitude test is designed for the experienced programmers to determine if they can do complex programming tasks such as pseudo codes, search for tables, Boolean true and false, looping, arrays, functions etc.

3.2.1 Types of Hybrid Tests

Hybrid tests are part of computer programming aptitude tests, which consists five types tests:

1) Non-Verbal Reasoning (Pattern Recognition Test):

This type to tests a mental abilities, such as ability to understand and analyze the shapes and solve the problems through identifying relationships, similarities and differences between shapes. This test similar to tests used during employment in the companies or school admittance exams.

This type of test is very important in aptitude test because it is the ability to see order in a chaotic environment and solve it. Pattern can be sequences of shapes, images, symbols and words. In the pattern recognition questions there are be sequences of shapes with a missing one, the candidates answer it by choose the right one. It allows candidates to solve the problem without language skills, so it solve the problem of people who have problem with languages (Parts of an IQ Test, 2016).

2) Problem Solving Test:

This test is to determine analytical ability, logical reasoning and persistence ability, also used to understand and analyze information and solve problems step by step. The problem solving test enables candidates to analyze and solve complex problems. This type consists of most types of numerical aptitude questions, this test used during employment in the companies or school admittance exams.

3) Logical Reasoning Test:

This type tests the ability of thinking logical and analytical, which consists of letters sequences. Also, this test similar to tests used during employment in the companies or school admittance exams.

4) Numerical Reasoning Test:

This type determines the numerical ability and how can applicants solve numerical problems in short time this test categorized as speed test, which consists of the main mathematical calculation (subtraction, addition, division and multiplication), the sequences of numbers (fractions, powers, percentages, etc.). This exam used during employment in

the companies or school admittance exams. It was advised not to use the calculator, but they can use pen and paper for these tests (Numerical Ability Tests, 2016).

5) Ability to Follow Complex Procedure Test:

This type tests the ability of thinking logically and analytically, which consists of complex puzzles and numerical procedures which need to thinking to extract meaning from it. This test used during employment in the companies or school admittance exams.

3.3 DATA COLLECTION AND ANALYSIS

The current study provides a set of students' scores (160) in aptitude test and programming test, which are entered in the computer by Microsoft Excel spreadsheet. After that the data was analyzed by using SPSS program, the data were copied and pasted from Microsoft Excel to SPSS program. Then, the comparisons was performed between the aptitude test grades and programming test grade, several analysis were conducted on the collected data which are Descriptive statistic, Correlating, Regression and T-Test by SPSS, to test the relationship between the aptitude test and the programming test. Also, it was used some histograms and scatter plot, which related with correlation and regression analysis.

The data analysis in the current study were performed by different statistical methods:

- In this study, the data analyses were conducted by using the SPSS program and the data were coded by the following techniques:

  - The scores of the students in aptitude test and programming test were transferred to the computer by SPSS program.
  - The engineering and programming departments are coded by 1 and 2 respectively, in SPSS program.

- Descriptive statistics were used to provide numbers of students, minimum, maximum, means and standard deviations for all students' scores for aptitude and programming tests.
- Correlations were applied to determine the strength of the linear relationship between aptitude and programming tests.
- Regressions were used to determine the linear relationship between the grade from aptitude test and the grade from programming test.
- Independent sample T-test, applied to compared the means and if there are differences between each other.
- The probability of doing error was ($\alpha = 0.05$).

3.4 RESEARCH QUESTIONS AND HYPOTHESISES

This thesis was built on the next three research questions with a set of hypothesis, the answers of the questions by the verification of the one of the set of hypotheses described.

- **Research question 1:** Is there a relationship between aptitude test and programming test?

H0: (There is no correlation between Aptitude test and programming test).

H1: (There is correlation between Aptitude test and programming test).

- **Research question 2:** What is the relationship between engineering department and programming department in aptitude test?

H0: (The distribution scores of engineering similar to distribution scores of programming in aptitude test).

H1: (The distribution scores of engineering are significantly different to distribution scores of programming in aptitude test).

- **Research question 3:** What is the relationship between engineering department and programming department in programming test?

H0: (The distribution scores of engineering similar to distribution scores of programming in programming test).

H1: (The distribution scores of engineering are significantly different to distribution scores of programming in programming test).

The analysis of the research questions by using the next statistical methods:

1) The analysis of the research Question 2 by performed the correlation and regression statistics.
2) Research Questions 3 and 4 were analyzed by performed the independent sample t-tests statistics.

3.5 ONLINE APTITUDE TESTS

Online aptitude testing is a common way to evaluate person's ability of succeeding in different situations. Most companies use these tests in the employment. These tests evaluate various abilities like problem solving, numerical ability, verbal ability and logical thinking etc. Each company create different test questions to be suitable with the work. Such as IT employment have test questions that evaluate persons' programming abilities. As well as, some universities and schools used online tests for the scholastic admission (Caroline, H, 2011).

Online aptitude tests have several advantages:

1) These tests are online, therefore, they can be taken from anywhere. The only two requirements for taking this test are having access to a computer and internet.
2) The online aptitude test is not time-consuming; most tests can be done in fifteen or twenty minute.

3) The results are generated in no time; anyone can get the results immediately.

These advantages make online aptitude testing suitable for initial screening.

3.6 DESIGN THE APTITUDE TESTS WEBSITE

The aptitude test website was built by using ASP.NET which was advanced by Microsoft to produce dynamic web pages. A free web framework for producing web sites by using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript and C#. HTML used for structuring pages content, CSS is for the design of the webpage and the appearance of the pages. However, JavaScript is used for interactive functionality, particularly to make dynamic pages.

The program (or platform) which has been used to write the codes is Microsoft Visual Studio, more specifically the 2012 version of it. It is an IDE (Integrated Development Environment) from Microsoft. It includes a code editor, and debugger. Also, it has other tools consist a web designer, forms designer which is Graphical User Interface, database designer and class designer. Microsoft Visual Studio can use a lot of languages, like; VB.NET, C, C#, C++, etc.

The online aptitude test has been designed for students who have a desire to enter to the computer departments to become computer programmers, this online test help them to know their programming abilities, for example, creativity, time management, problem solving skills, syntax, pattern recognition, ability to follow procedures, ability to follow instructions, attention to detail, logical thinking and basic mathematical skills. Figure 3.1 presents the main page from the website.

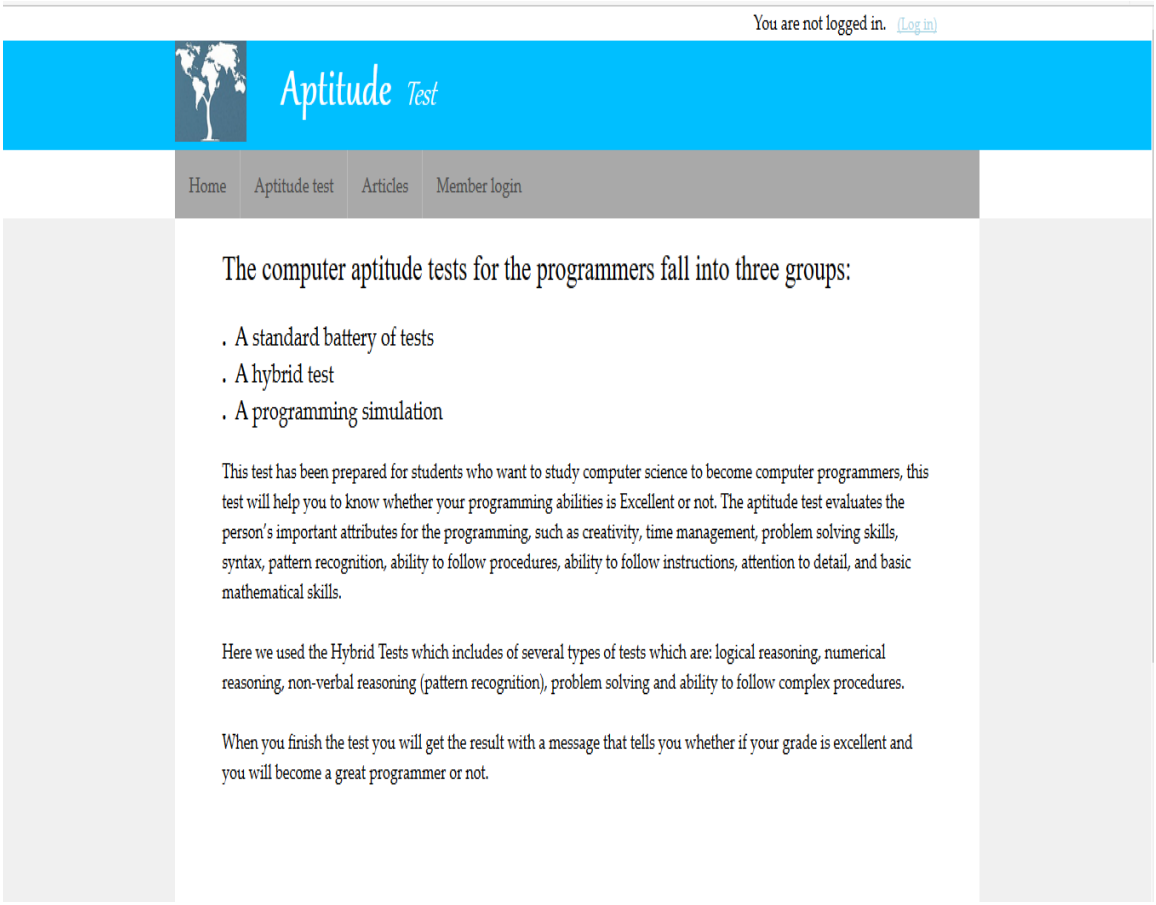


Figure 3.1 The main page.

The website has some articles about computer aptitude test (see figure 3.2).

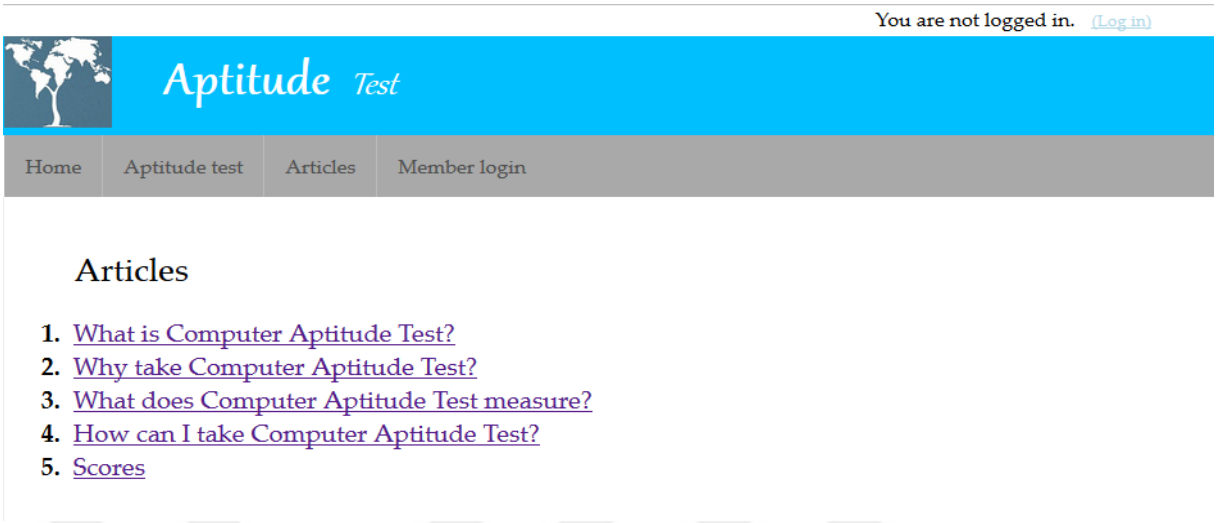


Figure 3.2 The articles page.

If anyone wants to do the exam, they need to register in the website, as presents in (figure 3.3). Whereas, if they already registered before they just need to login, as showed in (figure 3.4).

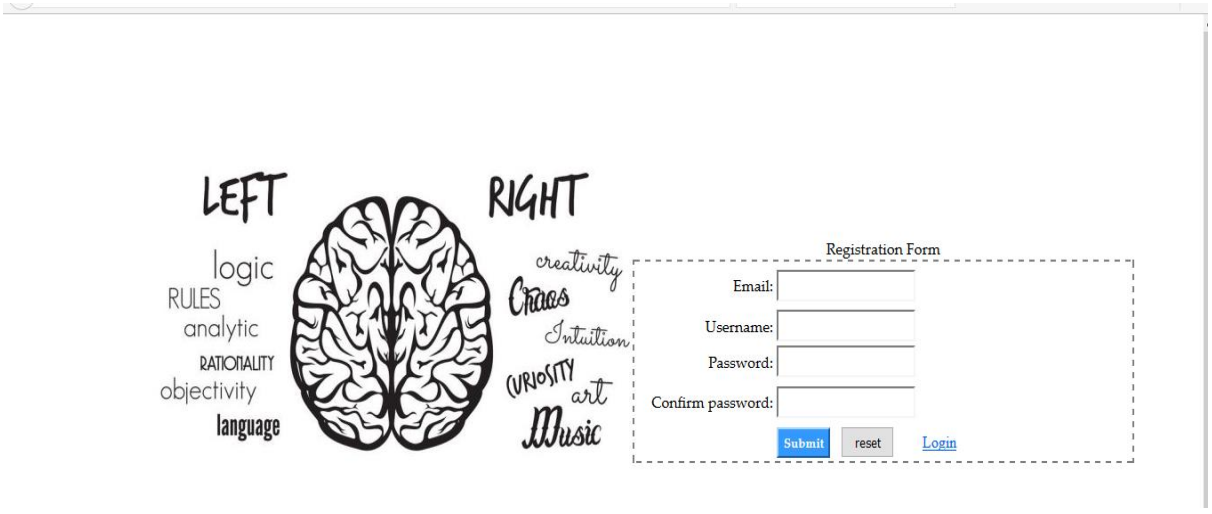


Figure 3.3 The registration page.

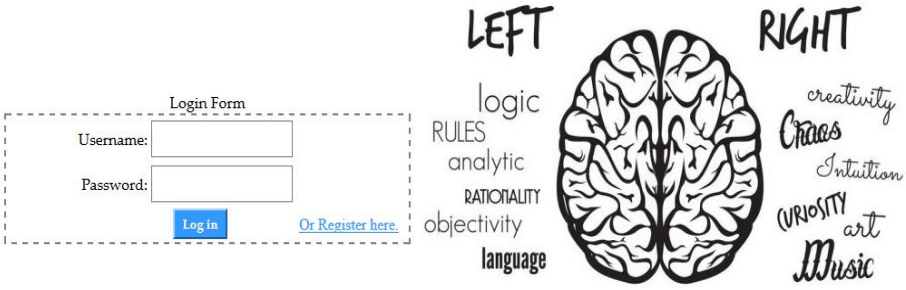


Figure 3.4 The login page.

The website have six types of aptitude tests which are general aptitude test, logical reasoning, problem solving, numerical reasoning, pattern recognition and ability to follow complex procedures which called (Hybrid tests), (see figure 3.5).

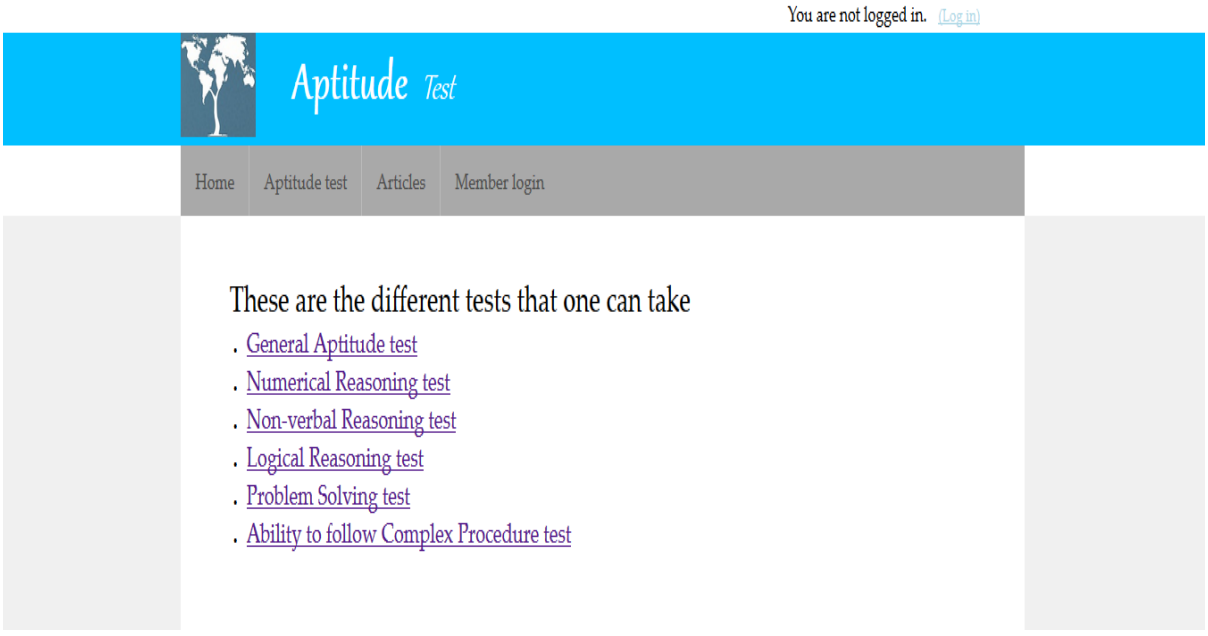


Figure 3.5 Types of aptitude tests page.

When you click on one of the previous tests, you will get the page that tells you what this test is, which ability the test measure, how much the questions are and the specified time to finish the exam, (see figure 3.6).

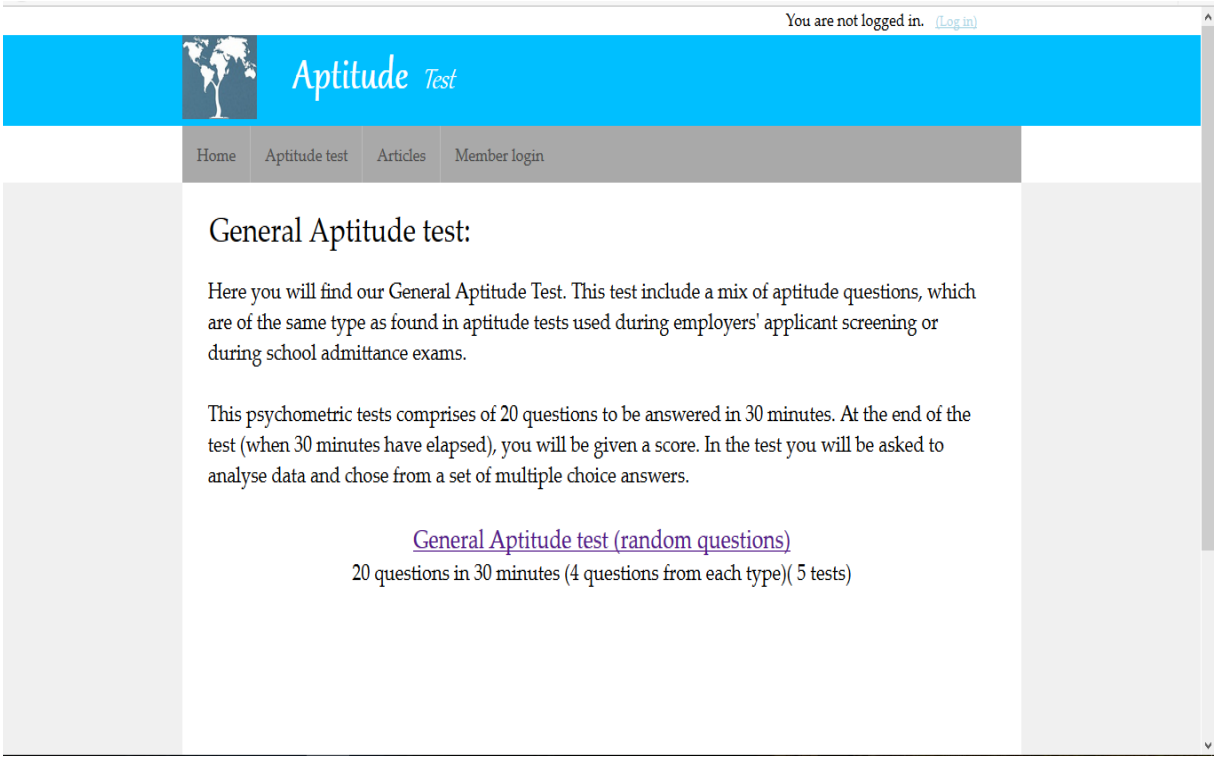


Figure 3.6 The page of test description.

When the test starts, you will get the first question which is Multiple Choice Question (MCQ) (see figure 3.7), at that time, the timer will start to count down, when the time is finished, you will get the score. Also you can go to the back question or next question, as well as you can end the exam and get the result before the timer finish.

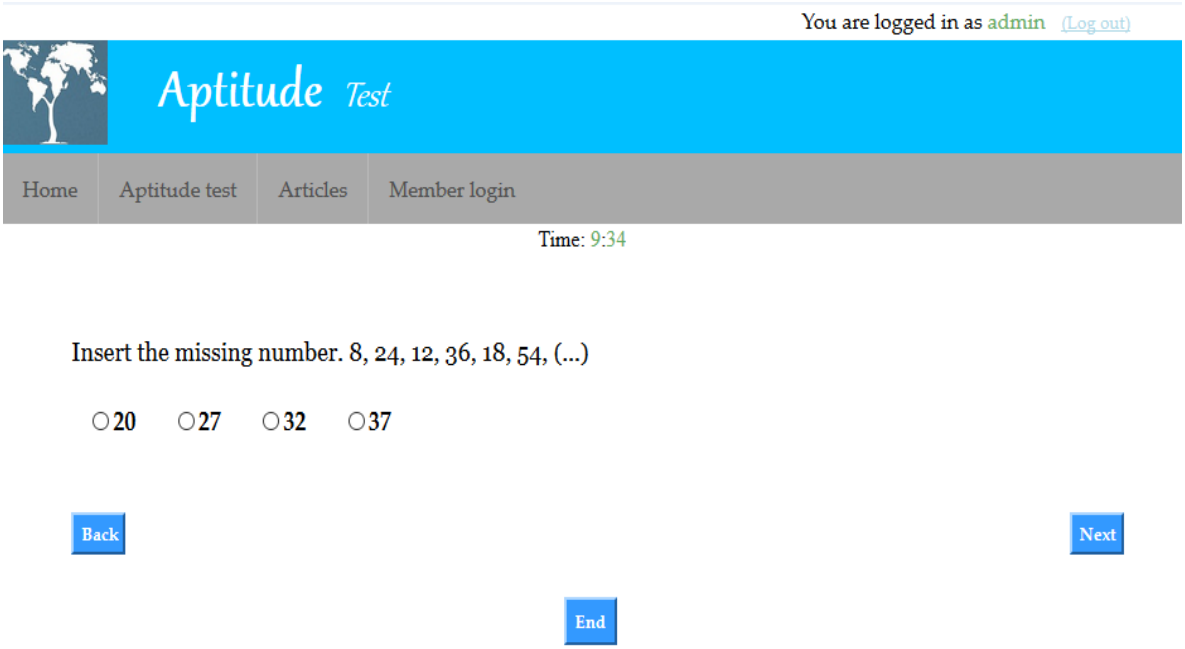


Figure 3.7 The page of test questions.

When the test is finished, you will get the following page (see figure 3.8), which tells you about your grade. Also, the number of correct answers, wrong answers and skipped questions. As well as, you will get a message telling you about your level in programming. Moreover, you can review your wrong answers.

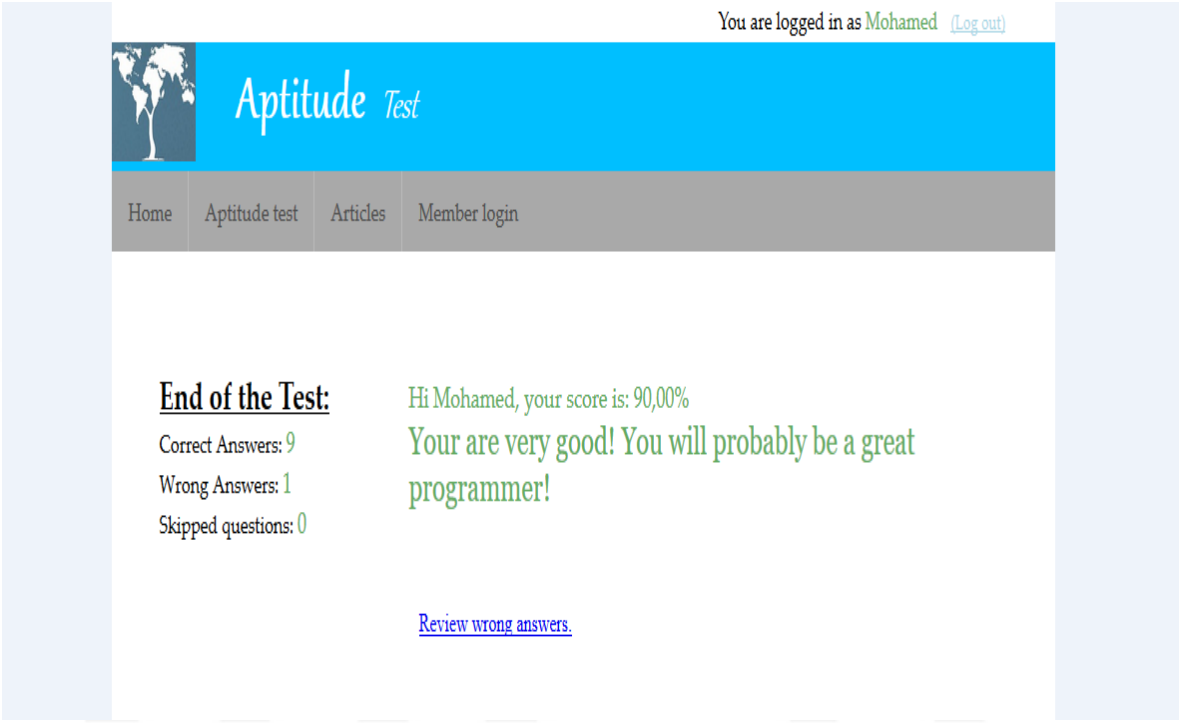


Figure 3.8 The test results page.

CHAPTER 4

ANALYSIS OF DATA

This chapter discusses the data analyses, collecting of the data and the statistical techniques utilized for the current study. Figures and Tables are shown to clarify the analysis of data in this chapter. The data collected includes student test scores in two tests which are the aptitude test and the programming test. These tests have been applied for two different departments; engineering department and programming department. Testing the effectiveness of the aptitude test, the following were performed: descriptive statistics, correlation, regression statistics, and independent sample t-tests. The analysis of data and gained of the results in this study by applying SPSS statistics 20.

Chapter 4 is structured as the following; it starts by small introduction. Then, it shows how the data were collected in this dissertation in section 4.1. After that explanation of the Statistical Methods used to Analysis of data is presented in Section 4.2. While Section 4.3 is the most important one as it includes the main results, this section presents the research questions given in Chapter 1 with hypothesis and answers by the statistical tests.

4.1 DATA COLLECTED

Data presented in this section were obtained from previously conducted aptitude test on the first year students, Department of Computer Science, Fatih University Istanbul. Therefore, the general aptitude test has been prepared from fifteen questions, which are divided in the following categories: four questions for numerical reasoning, four questions

for logical reasoning, three questions for problem solving, two questions for pattern recognition and two for ability to follow complex procedure. The test was applied on three groups; forty students from programming department in the first group and the test was in Turkish language. While the second group included students from engineering department, and this group consisted of two sub-groups; the first one (29 students) were tested in English language, and the second group (11 students) were tested in Turkish language. Finally, the grades from aptitude tests were compared with the grades from programming test, the results will be presented in this chapter in details.

4.2 STATISTICAL METHODS USED IN ANALYSIS

4.2.1 Correlation

The correlation statistics presented in Table 4.2 provide the correlation and significance level for each dependent and independent values. The correlations applied to measure the strength of the linear relationship of numerical values. The strength of linear relationship of two numerical values is determined by coefficient of the correlation, ρ, whose ranges from −1 to +1 (Pearson's Correlation, 2016).

- If the coefficient of the correlation is positive and closer to (1) that indicates to the positive and strong linear correlation;
- If the coefficient of the correlation is negative and closer to (-1) that indicates to the negative and strong linear correlation;
- If the coefficient of the correlation is (0) that indicates to no linear correlation;

4.2.2 Regression Analyses

Regression analysis is a method to decide if there is a linear relationship between variables (Campbell, Dan, and Sherlock Campbell, 2008). The regression equation by bivariate is easier than by Multivariate as in the following Eq. (4.1) (the regression

equation) which presented the connection between the variables, Y is indicates to the dependent variable. Whereas, X is indicates to the independent variable.

$$Y = B0 + B1X \tag{4.1}$$

In the simple regression equation, Y is a dependent variable, the symbols B_0 and B_1 are called regression parameters or coefficients, B_0 is called the intercept, B_1 is called the slop regression, and X is a predictor variable.

Aptitude test is the independent variable used to predict the programming test as in the Eq. (4.2).

$$\text{Programmin g test} = B0 + B1\,\text{Aptitude test} \tag{4.2}$$

4.2.3 Independent Sample T –Test

This statistical method is applied to compare means between groups, and if there are differences between each other. (Comparing Means Using T-tests, 2016).

4.2.4 Descriptive Statistics

This Statistical methods is applied to collect the data and analyze them, in order to describe the data and make the decision (CHAPTER 1 DESCRIPTIVE STATISTICS, 2016).

The descriptive statistics (see Table 4.1), provide the minimum score, maximum score, means scores and the Standard deviation for the scores of dependent and independent variables used in the current study. Figures 4.1 and 4.2 illustrate the aptitude test and the programming test frequencies.

Table 4.1 Descriptive Statistics.

	Number of students	Min	Max	Means	Standard Deviation
Aptitude test grade	80	7.7000	84.7000	47.915000	18.3100519
programming test grade	80	6.0000	98.0000	53.443750	22.7404221

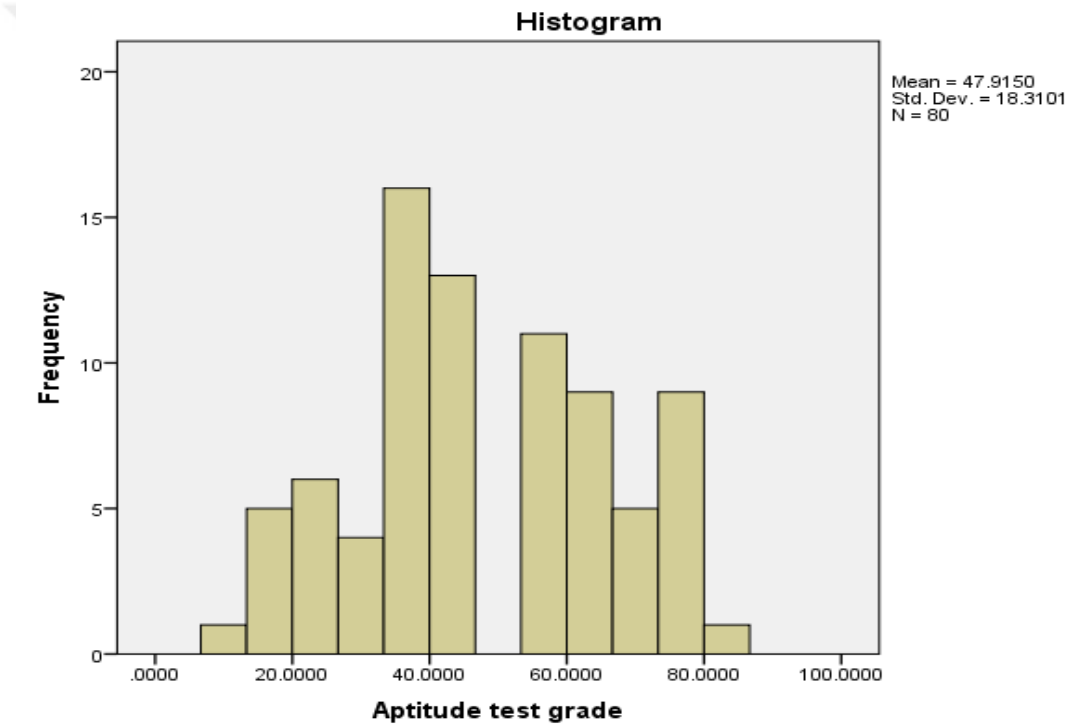


Figure 4.1 Aptitude test grade frequency.

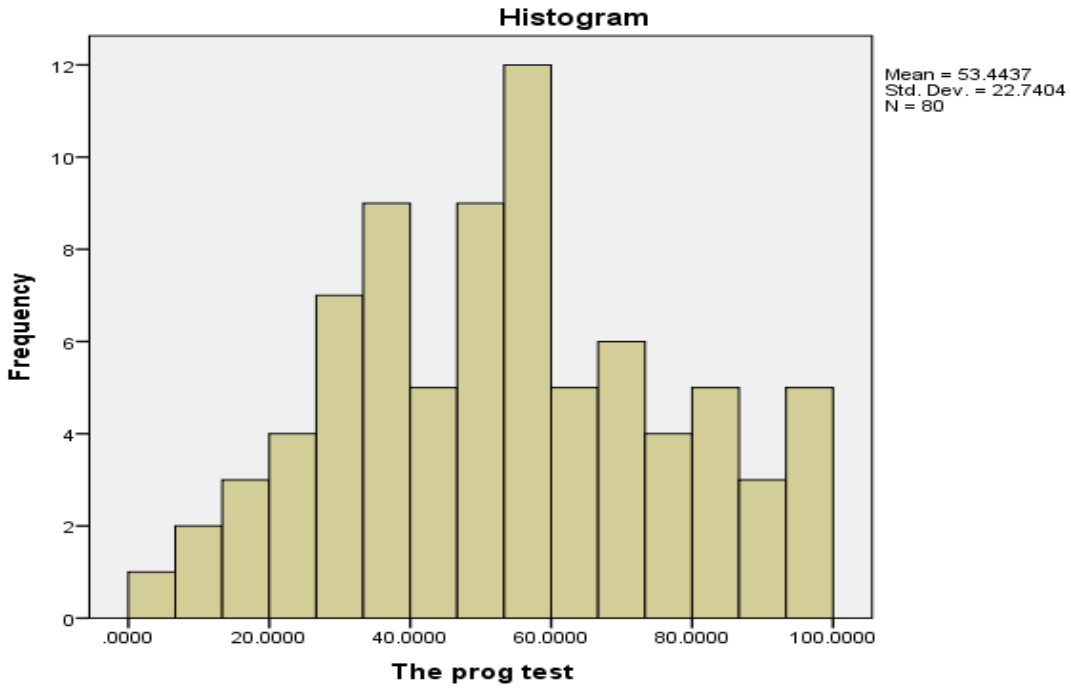


Figure 4.2 Programming test grade frequency.

4.3 RESEARCH QUESTIONS

Research question 1:

Is there a relationship between aptitude test and programming test?

Research Question 1 was analyzed using correlation and regression to find out if there is a relationship between aptitude and programming tests.

The correlation hypothesis in this case is:

H0: $\rho = 0$ (There is no correlation between Aptitude test and programming test).

H1: $\rho \neq 0$ (There is correlation between Aptitude test and programming test).

The correlation statistics (see Table 4.2), (the correlation coefficient =0.525) means the correlation coefficient is positive and strong, (Sig=0.00), p<0.01 means it has very strong evidence to believe H1. So, there is strong liner correlation between aptitude test and programming test.

Table 4.2 Correlations.

		Aptitude test grade	**programming test grade**
Aptitude test grade	**Correlation coefficient**	1	.525**
	Significant		.000
	Number of students	80	80
programming test grade	**Correlation coefficient**	.525**	1
	Significant	.000	
	Number of students	80	80

**. The correlation is strong at the 0.05 level and it is very strong at the 0.01 level.

Figure 4.3 is a scatter plot of the correlation between aptitude test and programming test and it shows a significant correlation between aptitude and programming tests.

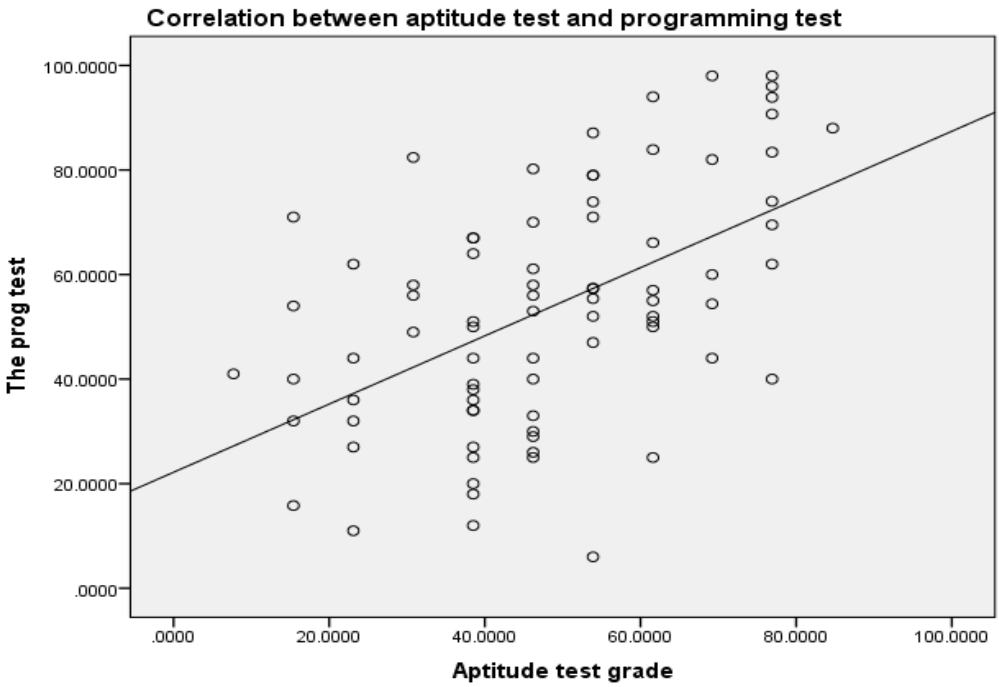


Figure 4.3 scatter plot of the correlation between aptitude test and programming test.

The regression hypothesis in this case is:

H0: B = 0 (The relationship between Aptitude test and programming test is not significant).

H1: B ≠ 0 (The relationship between Aptitude test and programming test is significant).

From table 4.3 ($B0$=22.192) which means when the aptitude test=0 the programming test=22.192, ($B1$=0.652) as in the following Eq. (4.3).

So:

$$\text{programmin g test} = 22.192 + 0.652 * \text{Aptitude test} \quad (4.3)$$

The (sig=0.00) and (α level = 0.05) so, sig < 0.05. That return the H1 hypothesis (The relationship between Aptitude test and programming test is significant), that means the Aptitude test statistically effects of the programming test.

Table 4.3 The Regression.

Model	Unstandardized Coefficients		Standardized Coefficients	T	Significant
	B	Standard Error	Beta		
Aptitude test grade (Constant)	22.192	6.134		3.618	.001
	.652	.120	.525	5.450	.000

From table 4.4, the (correlation coefficient R= 0.525) and (the R coefficient square= 0.276), which indicates that 27.6% of the variation of the Aptitude test is explained by the model. From table 4.5 the p value of Anova equal to p value of slop in the table 4.3 which is (sig= 0.00).

Table 4.4 Model Summary.

Model	R	R Square	Adjusted R Square	Standard Error	Durbin-Watson
1	.525[a]	.276	.267	19.4758432	.870

Table 4.5 ANOVA.

	Total of Squares	Degrees of Freedom	Means Square	F	Significant
Residual Total	11266.956 29586.060 40853.017	1 78 79	11266.956 379.308	29.704	.000[b]

Research question 2:

What is the relationship between engineering department and programming department in aptitude test?

Research question 3:

What is the relationship between engineering department and programming department in programming test?

Research Question 2 and 3 were analyzed using independent samples t-test in order to compare means between engineering and programming departments in aptitude test and programming test.

From table 4.6 For Aptitude test there are two groups engineering and programming, the number of students 40 for each group, the mean is different between engineering and programming 55.41 and 40.42 respectively, the standard deviation is slightly different between engineering and programming 17.93 and 15.55 respectively.

For programming test also there are two groups; engineering and programming departments and the number of students was 40 for each group. The mean and the standard deviation is slightly different between engineering and programming (57.80, 49.08) and (25.82, 18.48) respectively.

Table 4.6 independent sample T-Test (Statistical Groups).

THE GROUPS		Number of students	Means	Standard Deviation	Standard Error
Aptitude test grade	**ENGINEERING**	40	55.410000	17.9380958	2.8362620
	PROGRAMMING	40	40.420000	15.5597640	2.4602147
programming test grade	**ENGINEERING**	40	57.802500	25.8251566	4.0833158
	PROGRAMMING	40	49.085000	18.4825011	2.9223400

The hypothesis in this case is:

H0: $P_{eng} = P_{prog}$ (The distribution scores of engineering similar to distribution scores of programming).

H1: $P_{eng} \neq P_{prog}$ (The distribution scores of engineering are significantly different to distribution scores of programming).

It should compare t- test p value with our α level of significant which is (0.05).

If the p value of the t- test is bigger than (0.05) → H0 is not rejected.

If the p value of the t- test is smaller than (0.05) → H0 is rejected.

It should be known whether assume the variances are equal or not assume the variances are equal. From table 4.7 the Aptitude test (sig = 0.280) > 0.05 that means there is no differences, thus, in this case the equal variances assume is used. But, the programming test (sig=0.048) < 0.05 which indicates there is difference, thus, in this case the equal variances not assume is used

Table 4.7 Independent Sample T-Test (Variances Assumed).

		Variances Assumed		Means
		F	**Significant**	**T**
Aptitude test grade	**Assumed equal of variances**	1.181	.280	3.992
	Assumed not equal of variances			3.992
programming test grade	**Assumed equal of variances**	4.054	.048	1.736
	Assumed not equal of variances			1.736

From table 4.8, the Aptitude test (sig = 0.00) which is smaller than the p value (0.05), so the hypothesis 1 (H1) is selected meaning that the aptitude test results between engineering department and programming department are different. Whereas the programming test (sig = 0.087) which is bigger than the p value (0.05), so the null hypothesis (H0) is selected, that means programming test results between engineering department and programming department are not different.

The mean difference is the subtract means between engineering and programming groups. So in the aptitude test = 14.99 and in the programming test = 8.71.

Table 4.8 Independent Sample T-Test (Differences of Mean).

		Differences of Mean		
		Degrees of Freedom	**Significant**	**Difference Of Mean**
Aptitude test grade	**Assumed equal of variances**	78	.000	14.9900000
	Assumed not equal of variances	76.473	.000	14.9900000
programming test grade	**Assumed equal of variances**	78	.086	8.7175000
	Assumed not equal of variances	70.648	.087	8.7175000

From table 4.9, Confidence Interval of the Difference this confidence for mean different, 95% is actual different of mean engineering and programming is between (7.51- 22.46) in the Aptitude test and between (-1.29 – 18.73) in the programming test.

Table 4.9 Independent Sample T-Test (Confidence Interval).

		Differences of Mean		
		Standard Error	**95% Difference in Confidence Interval**	
			Lower	**Upper**
Aptitude test grade	**Assumed equal of variances**	3.7546023	7.5151610	22.4648390
	Assumed not equal of variances	3.7546023	7.5128100	22.4671900
programming test grade	**Assumed equal of variances**	5.0213085	-1.2791573	18.7141573
	Assumed not equal of variances	5.0213085	-1.2955682	18.7305682

Figure 4.4 illustrates the frequencies of aptitude test for engineering and programming groups and it shows that the aptitude test results between engineering group

and programming group are different. Whereas, Figure 4.5 shows the frequencies of programming test for engineering and programming groups and it shows that the programming test results between engineering group and programming group are not different.

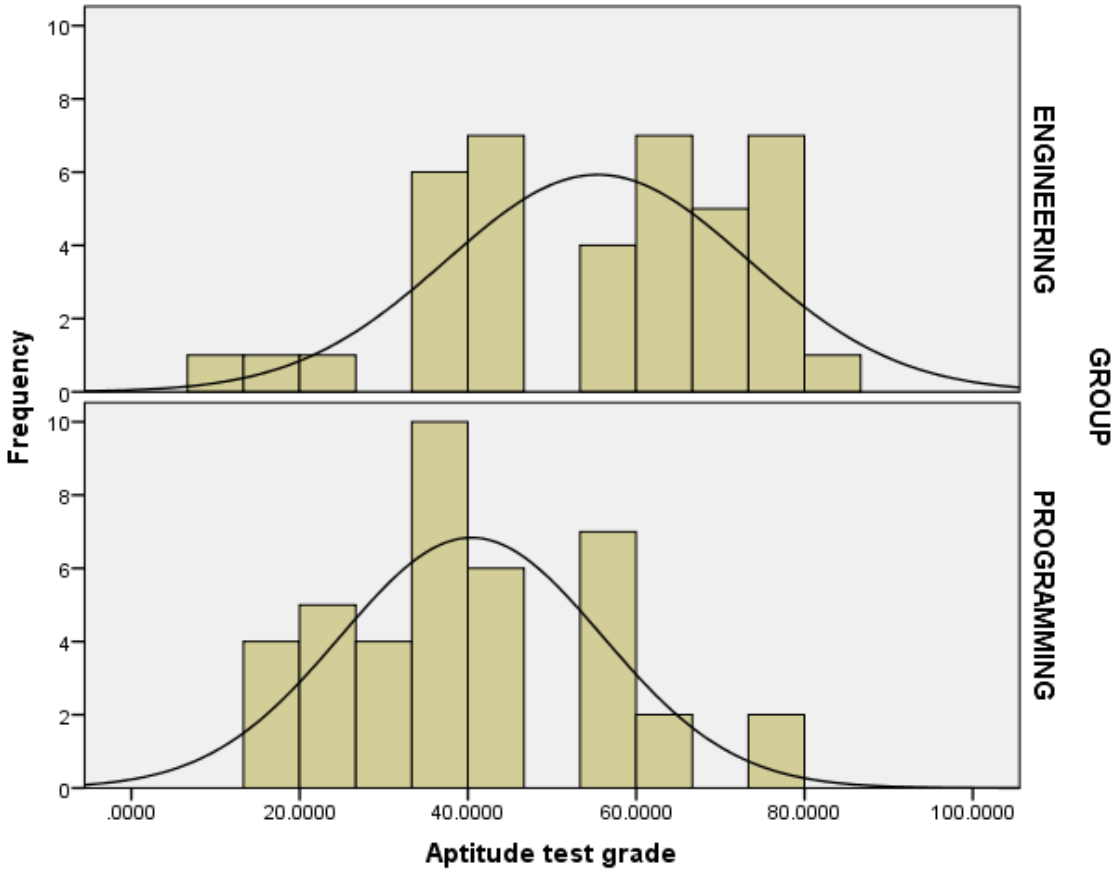


Figure 4.4 The frequency of aptitude test for engineering and programming groups.

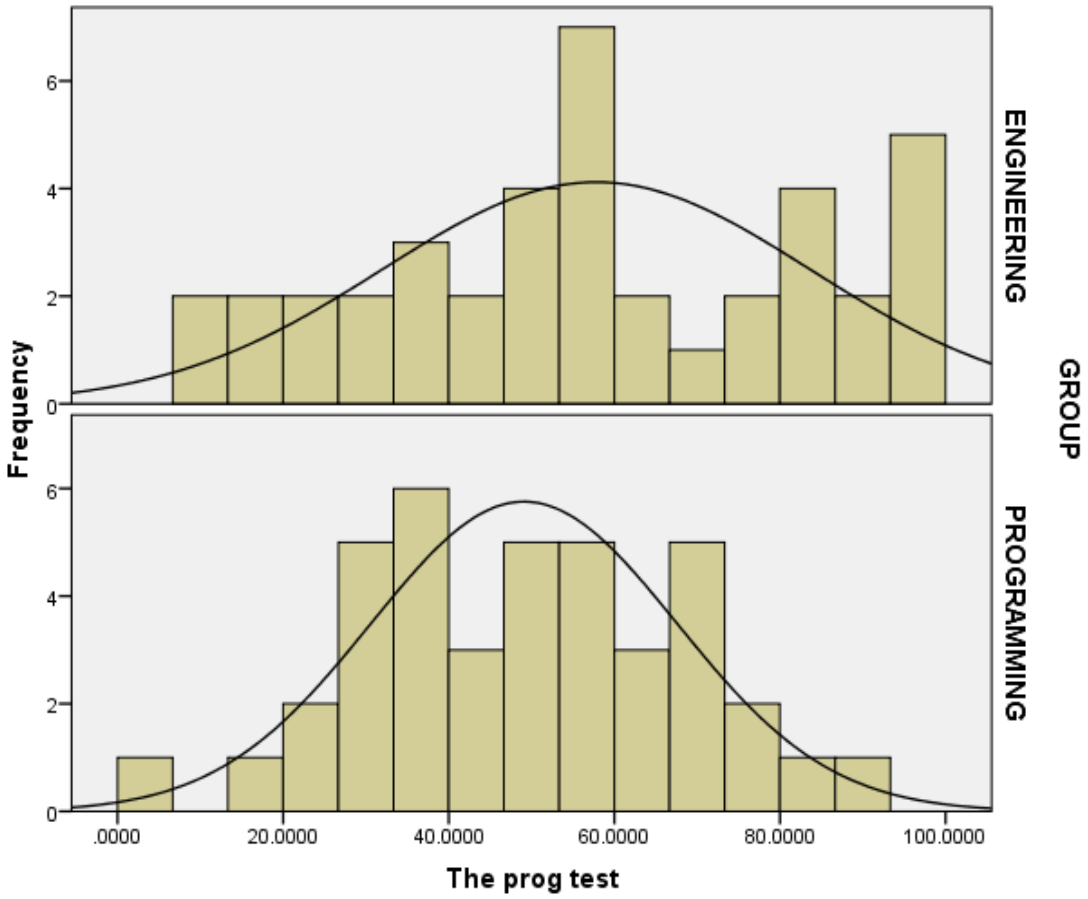


Figure 4.5 The frequency of programming test for engineering and programming groups.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Chapter 5 presents the main conclusions for the study, This chapter starts by given a summary of this project; by clarifying the main problem and explain how it was solved, and then presents key results from performing statistical analysis methods in section 5.1. Lastly, recommendations and future work will be discussed for future research into this regard in section 5.2.

5.1 CONCLUSIONS

The overall objective of this thesis is to know the students their programming ability before enter to computer science. Large number of students desire to study computer science, at the same time they cannot predict whether they have the abilities that can help them in learning and succeeding in programming. For this reason this study was performed in order to identify programing skills for students in early stage of learning computer science to avoid students' failure in programming.

A general introduction of the learning programming skills was first presented in Chapter 1 with some problems from learn programming for the students in the introductory of programming courses. These problems led to the main focus of this thesis which is determining the programming ability for students prior to programming course. This chapter also explains briefly the aptitude test and highlights the main problems and objectives.

The pedagogical literature was presented in learning and evaluating programming skills, types of abilities, types of tests and long discussion on computer programming aptitude tests. All these themes was adequately discussed in Chapter 2.

Chapter 3, a detailed description of the proposed test was presented by explaining ways to prepare and perform the proposed test on 80 students from two departments of computer science, in order to examine the students' success prediction in programming, by measuring of students' programming abilities that is the main aim of this research, by conducting an extensive study on computer programming aptitude tests.

Therefore, this study presented the computer aptitude test in Chapter 3, the computer aptitude test was used to evaluate students' programming abilities before they enter to university and start learning programming to avoid the students' failure. The current study was used the hybrid test particularly, because of it includes several types of general abilities tests. Generally, hybrid test consists of five main types of tests:

1) Non-verbal reasoning test (Pattern recognition test).
2) Problem solving test.
3) Logical reasoning test.
4) Numerical reasoning test.
5) Ability to follow complex procedure test.

Finally, this study answered three main research questions. Through conduct several statistical analysis methods which were descriptive statistics, correlation, regression statistics, and independent sample t-tests in Chapter 4 on the data collected from these tests, and revealed about interesting results. The following remarks are an overview of key results of the data analysis that was conducted in Chapter 4:

- The analysis results by using descriptive statistic to compare the aptitude test grades with the programming test grades which performed on the computer science students provided several parameters regarding the scores including minimum, maximum, means and the standard deviation.

- The statistical techniques were discussed to determine the linear relationship between aptitude test grades and programming test grades by used the correlation and regression, through compared the significant value with the p value as the Sig = 0.00, so, it is smaller than (0.01), therefore the hypothesis1 (H1) was selected, also the correlation coefficients (r = 0.525) showed that the relation between the aptitude test and the programming test was positive and strong. And proved that there are significant relationship between aptitude test and programming test. The aptitude test grades used to predict of programming test grades.
- Other technique used to compare means which was independent sample t-tests, the results showed that the results between engineering department and programming department in aptitude test are different, because of compared the significant value with the p value, the Aptitude test the Sig = 0.00 which is smaller than the p value (0.05), so the hypothesis 1 (H1) is selected. Whereas, the results between engineering department and programming department in the programming test are not different, also due to compared the significant value with the p value, the programming test the Sig = 0.087 which is bigger than the p value (0.05), so the hypothesis 0 (H0) is selected.

After proving that aptitude test success for used as a predictor of success in computer science courses. The online aptitude test was widely presented in Chapter 3. The website has been designed for students to help them to determine their programming abilities, like creativity, time management, problem solving skills, logical thinking, pattern recognition, basic mathematical skills, attention to details, ability to follow procedures etc. This website has six types of tests which consists the previous mentioned types in addition to the general aptitude test. Moreover, when the test ends the website gives the result immediately with a message explaining the examiner programming level.

5.2 FUTURE WORK

In spite of the results for the current study has proven the effectiveness of the programming aptitude test (PAT), there are some improvements that can still be made, by increase developed in a several of ways.

This section presents in short several interesting research topics that need to be further investigated. As in the following points:

- One of the improvements can be done on this thesis is develop the questions in the aptitude test. The hybrid test was used to measure programming ability, which consists five main types of tests as previous mentioned. So, the test questions which will be developed by searching for another types of hybrid tests that measure other various programming abilities.
- Another improvement which should be taken into consideration is develop the website by add other features. Any future work can utilize this web design and add more programming tests and features to develop it by the right way.
- Another important observation that we must take into account is the reliability. The programming aptitude test has proved through the results to be a successful test which used to evaluate of a variety of programming skills targeted toward the novice programmer. Because of time constraints, the test was applied only on two groups. So, the tests which were done by this dissertation may well be re-evaluated again using different programming tests with other programming groups to increase the reliability.

There is clearly much work to be done in the area of aptitude tests.

REFERENCES

"Advantages and Disadvantages of Conducting Tests For Selection of Employees." Your Article Library com The Next Generation Library. 2014. Web. 11 Apr. 2016. http://www.yourarticlelibrary.com/employee-management/advantages-and-disadvantages-of-conducting-tests-for-selection-of-employees/25959/

Alahakoon, P. M. K, A. I. Thilakasiri, and P. S. R. Peiris. "A Hybrid Testing System for ICT Skills Assessment of Students." International Journal of Scientific and Research Publications 2.12, University of Peradeniya Sri Lanka (2012). Web.

Alan Blackwell, Thomas Green, Chris Roast, and Peter Robinson. Cognitive models of programming-like activity. USA (New York), Pages 910{911, 2002. http://doi.acm.org/10.1145/506443.506654.

"Aptitude Assessment." Aptitude Assessment. 2013. Web. 22 Mar 2016. http://www.theworksuite.com/id15.html

"Aptitude Tests; A Historical Perspective." Personality and Aptitude Career Tests, 2004. Web. 14 Mar 2016. http://www.personality-and-aptitude-career-tests.com/aptitude-tests.html

"Aptitude Test Definition | Investopedia." Investopedia. 2010. Web. 09 Apr. 2016. http://www.investopedia.com/terms/a/aptitude-test.asp

"Aptitude Tests - Introduction." Aptitude Tests. Web. 09 Apr. 2016. http://www.psychometric-success.com/aptitude-tests/aptitude-tests-introduction.htm

Bennedsen, Jens and Michael E. Caspersen. "Abstraction Ability as an Indicator of Success for Learning Object-oriented Programming?" ACM SIGCSE 38.2 (2006): 39. Web.

Bereiter, C and E. Ng. Three Levels of Goal Orientation in Learning. 1991, Pages 243-271, Vol.1.

Besie, C., et al., An Examination of Age, Race, and Sex as Predictors of Success in the First Programming Course. Pages 51-64. 2003.

Byrne, Pat, and Gerry Lyons. "The Effect of Student Attributes on Success in Programming: UK, Canterbury, ACM, Pages 49 52, 2001.

Campbell, Dan, and Sherlock Campbell. "Introduction to Regression and Analysis of Data.". 28 Oct. 2008. Web. 25 Feb. 2016. http://statlab.stat.yale.edu/workshops/IntroRegression/StatLab-IntroRegressionFa08.pdf

"Can Aptitude Tests Really Predict Employee Success?" Human Resources com RSS. Web. 11 Apr. 2016. http://www.humanresources.com/1565/can-aptitude-tests-really-predict-employee-success/

Caroline, H. "The Benefits of Online Aptitude Testing." Ezinearticles. 14 Apr. 2011. Web. 20 Apr. 2016. http://ezinearticles.com/?The-Benefits-of-Online-Aptitude-Testing&id=6179008

Caspersen, Michael E., Jens Bennedsen, and Kasper Dalgaard Larsen. "Mental Models and Programming Aptitude." ACM SIGCSE Bulletin 39.3 (2007): 206. Web.

Chamillard, A. T., and Jay K. "Using Lab Practica to Evaluate Programming Ability." (2001). ACM SIGCSE Bulletin 33.1 .Web.

"CHAPTER 1 DESCRIPTIVE STATISTICS." Mypolyuweb. Web. 25 Feb. 2016. http://www.mypolyuweb.hk/machanck/lectnotes/c1_des.pdf

"Comparing Means Using T-tests." University of Vermont. Web. 25 Feb. 2016. http://www.uvm.edu/~dhowell/fundamentals7/SPSSManual/SPSSLongerManual/SPSSChapter7.pdf

"Computer Programmer." What Does a Computer Programmer Do? Web. 04 Apr. 2016. https://www.sokanu.com/careers/computer-programmer/

"Computer Programming." Wikipedia. Wikimedia Foundation. Web. 12 Mar 2016. https://en.wikipedia.org/wiki/Computer_programming

Cross, E. “The behavioral styles of computer programmers”. USA. Maryland, WA.1970.

Dan, F. "Standardized Testing." Time. Time Inc., 2009. Web. 21 Mar 2016. http://content.time.com/time/nation/article/0,8599,1947019,00.html

"Different Types of Tests." MIT Center for Academic Excellence: Mastering Tests. Web. 06 Apr. 2016. http://web.mit.edu/uaap/learning/test/testtypes.html

"Exam Questions" Centre for Teaching Excellence. 2012. Web. 06 Apr. 2016. https://uwaterloo.ca/centre-for-teaching-excellence/teaching-resources/teaching-tips/developing-assignments/exams/questions-types-characteristics-suggestions

Fincher, S. “What are we doing when we teach programming?” (1999), Pages 41-5.

Fletcher, Stephen H. Cognitive Abilities and Computer Programming. Research/Technical (143). Los Angeles, Ca, (1984).

Huoman, J. “Predicting programming aptitude”. University of Joensuu, Finland. Department of Computer Science, Master's Thesis. 1986.

"Individual and Physical Abilities." Management Consulting Courses. Web. 3 Apr. 2016. http://managementconsultingcourses.com/Lesson17Individual&PhysicalAbilities.pdf

Kathryn D. Sloane and Marcia C. Linn. “Instructional Conditions in Pascal Programming Classes”. 1988. Pages 207-235.

Kolb, D. “Leaning Style Inventory”. 1985.

Lister, R., et al., “A Multi-National Study of Reading and Tracing Skills in Novice Programmers”. 2004.

Mayer, D.B. “A computer manager speaks: \my problems are...”. ACM. New York, USA, Pages 117-125. 1964.

Mayer, D.B and Stalnaker, A.W. “Selection and evaluation of computer personnel” National Conference, ACM. Las Vegas, NV, USA, Pages 657-670, 1968.

Mayer, R. E. “A psychology of how novices learn computer programming”. Pages 121-141 Computing Surveys, (1981).

McCoy, L.P. and Burton, J.K. “The Relationship of Computer Programming and Mathematics in Secondary Students”. Pages 159 166. (1988).

McCoy, L.P. “Literature relating critical skills for problem solving in mathematics and computer programming”. (1990). Pages 48-60.

Mccracken, Michael, Tadeusz Wilusz, Lynda Thomas, Vicki Almstrum, Mark Guzdial, Cary Laxer, Danny Diaz, Yifat Ben-David Kolikant, Ian Utting, and Dianne Hagan,. "A Multi-national, Multi-institutional Study of Assessment of Programming Skills of First-year CS Students." (2001). Web.

Nathan, E. "The Computer Boys Take Over." The Computer Boys Take Over. Web. 16 Mar 2016. http://thecomputerboys.com/?p=369

"Numerical Ability Tests." Web. 18 Apr. 2016. http://www.psychometric-success.com/aptitude-tests/numerical-aptitude-tests.htm

Organ, D. W. “Predicting success in school and at work”. (2001).

"Parts of an IQ Test." IQ Test Experts. Web. 18 Apr 2016. http://www.iqtestexperts.com/iq-test-parts.php

PBS. PBS, 1995. Web. 21 Mar 2016. http://www.pbs.org/wgbh/pages/frontline/shows/sats/where/history.html

Pea, R. “Language independent conceptual bugs in novice programming”. (1986). Pages 25-36.

"Pearson’s Correlation." Statstutor. Web. 25 Feb. 2016. http://www.statstutor.ac.uk/resources/uploaded/pearsons.pdf

Pennington, N and Grabowski, B. “The tasks of programming”. Academic Press, London, UK. Pages 45–62. 1990.

Ramalingam, V and Wiedenbeck, S. “Development and validation of scores on a computer programming”. 1998. Pages 367-381.

Ransdell, S. “Predicting college success: the importance of ability and non-cognitive variables”. (2001) Pages 357-364.

Reinstedt, R. N. Beulah C. Hammidi, Sherwood H. Peres, and Evelyn L. “Ricard. Programmer performance prediction study”. Santa Monica, Calif., 1964.

Rountree, N. et al. “Interacting factors that predict success and failure in a CS1 course”. Leeds, U K. ACM. 2004.

Rountree, N. et al. “Predictors For success in studying CS”. ACM. Norfolk, Virginia, USA. 2004.

Saeed Dehnadi. “A Cognitive Study of Learning to Program in Introductory Programming Courses”, 2009. Middlesex University, London, Doctor’s Thesis of Philosophy.

Saeed Dehnadi. "Testing Programming Aptitude." (2006). Middlesex University, UK. Pages 22-37. Web.

Serrano, C, Luis Miguel, Carlos-María Alcover, Maximiliano Paredes-Velasco, and J. Ángel Velazquez-Iturbide. "An Evaluation of Students’ Motivation in Computer-supported Collaborative Learning of Programming Concepts." Pages 499-508. 2014. Web.

Simon, et al., “Predictors of success in a first programming course”. 2006.

Soloway, E., Greenspan, J. Bonar, J., and Ehrlich, K., “What do novices know about programming?”. (1982). Pages 27-54.

Spohrer, J., and Soloway, E. “Novice mistakes: Are the folk wisdoms correct?”. (1986). Pages 624-632.

Tony, J. "On the Difficulty of Learning to Program." Jenkins on the Difficulty of Learning to Program. 23 Sept. 2002. Web. 28 Mar 2016. http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

Tukiainen, M and Mönkkönen, E. “Programming aptitude testing as prediction of learning to program”. London, 2002. Pages 45-57,

"Types of Tests." Good Luck Exams. Web. 06 Apr. 2016. http://www.goodluckexams.com/types-of-tests/

Wang, Tiantian, Yuying Wang, Xiaohong Su, Kuanquan Wang, and Peijun Ma. "Ability-training-oriented Automated Assessment in Introductory Programming Course." Computers & Education (2011). Pages 220-26. Web.

"What Does a Computer Programmer Do? Careers and Degrees." What Does a Computer Programmer Do, How to Become a Computer Programmer Careers and Degrees. Web. 04 Apr. 2016. http://www.yourfreecareertest.com/what-does-a-computer-programmer-do.html

"What Is an Ability? Definition and Meaning." Web. 02 Apr 2016. http://www.businessdictionary.com/definition/ability.html

"What Is an Aptitude Test - Beginner's Guide to Psychometric Tests – Job Test Prep." Job Test Prep. Web. 09 Apr. 2016. https://www.jobtestprep.co.uk/beginners-guide-aptitude-tests#technical-test

"What Is Aptitude? Definition and Meaning." BusinessDictionary.com. Web. 02 Apr 2016. http://www.businessdictionary.com/definition/aptitude.html

"What Is Aptitude Test? Definition and Meaning." BusinessDictionary.com. Web. 02 Apr 2016. http://www.businessdictionary.com/definition/aptitude-test.html

"Why Programmer Aptitude Test Reject Programmers for Jobs?" Personality and Aptitude Career Tests. 2004. Web. 17 Apr. 2016. http://www.personality-and-aptitude-career-tests.com/programmer-aptitude-test.html

Wilson, B.C. and S. Shrock, “Contributing to success in an introductory computer science course”: Charlotte, North Carolina, U S. 2001.

Winrow, B. “The Walden programmer analyst aptitude test”. 1999, http://www.ddj.com/documents/ s=894/ddj9914b/9914b.htm

Wolfe, J. M. “Perspectives on testing for programming aptitude”. Pages 268-277. 1971.

APPENDIX A

COMPUTER APTITUDE TEST

1) Insert the missing number?

1, 4, 9, 16, 25, 36, 49, (….)

2) Think of number, squares it, take away 7, multiple by 3, added 8, and finally divide it by 7. To give the answer of 5. What is the starting number?

3) If the code for (ASIA) is (CTJC). What is the code of (AFRICA)?

4) In box there are 8 red, 7 blue and 6 green balls. One ball is picked up randomly. What is the probability that it is neither red nor green?

5)

	A	**B**	**C**	**D**	**E**	**F**
1	5	10	2	4	0	0
2	2	5	7	12	0	0
3	9	3	6	8	0	0
4	4	8	15	11	0	0

Step 1: multiply A3 by C4 and store it in E1.

Step 2: multiply B4 by D2 and store it in F3.

Step 3: if the value of E1 is larger than F3 swap their contents, otherwise leave them as they are.

Step 4: subtract F3 from B1 then add 1 to C1.

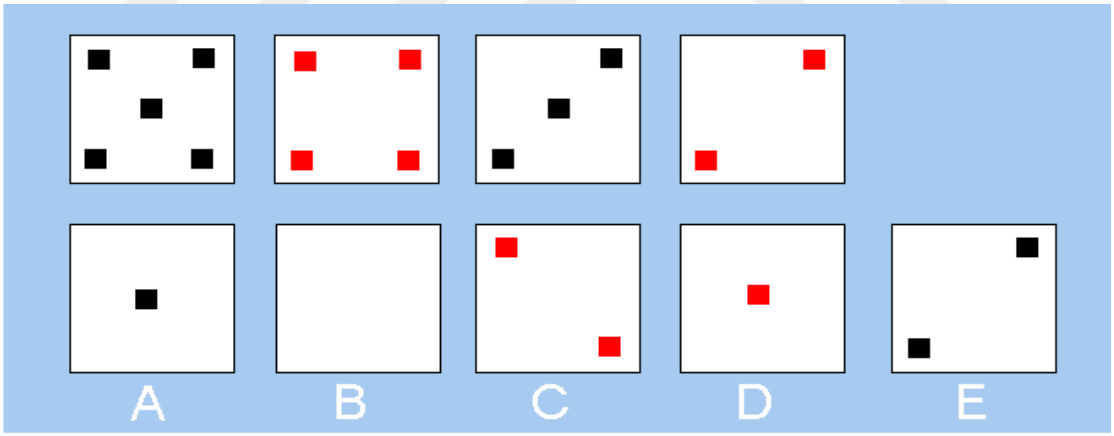
Repeat step 4 until the value of C1 equal to 5.

What is the value of F3?

6) Three friends they sat next to each other's, And each one wearing different T-shirt colors, Ahmed was sit left of Ali, but not necessarily next to it, the white T-shirt was to the right of the Green T-shirt, the blue T- shirt was to the left of the Omer, Omer was sit the left of Ali.

What was the order of them?

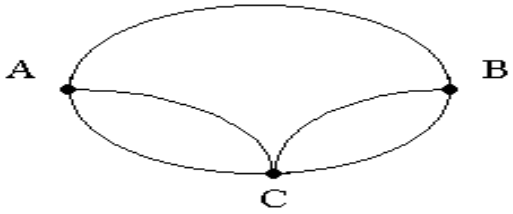
7)



8) What is the missing letter in this sequence?

a z b ? c v

9)



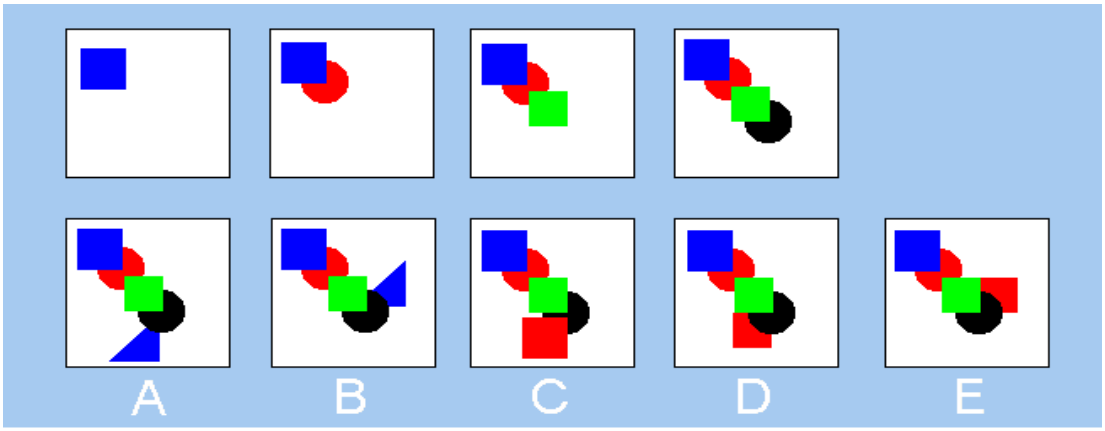
Ali has to visit towns B and C in any order. The roads connecting these towns with her home are shown on the diagram. How many different routes can she take starting from A and returning to A, going through both B and C (but not more than once through each) and not travelling any road twice on the same trip?

10) Which is the following fraction is the largest?

7/8, 13/16, 31/40, 63/80

11) If DAC=12, BAC=9, And DCCh=25. What is the value of (AMAR)?

12)



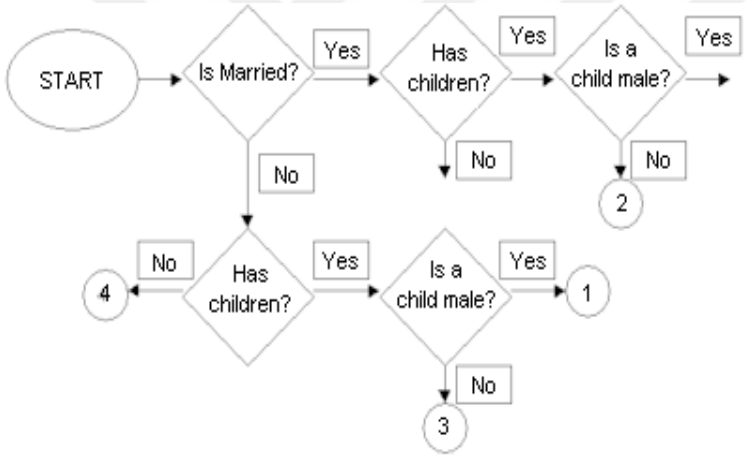
13) Is two years older than (B) who is twice as old as (C). If the total of the ages of A, B and C be 27. How old is B?

14)

Jklmsqdr	Jklmsqdr
Olgbehiz	Olgbehiz
Cqmdfdoj	Cqmdfdoj
ydfsjnop	ydfsjmop
Klvmojis	Klvmojis
qaspoany	qaspoany

Above there are 6 rows containing two identical sets of characters, but in one row there is one character in the two sets is different. Which row has the different?

15) Consider the following flow chart for a customer:



The person in one is:

A. Married, with children.
B. Married, with at least one son.
C. Unmarried, with at least one daughter.
D. Unmarried, with at least one son.
E. Unmarried, with no children.

APPENDIX B

Bilgisayar Yetenek Sınavı

1) Eksik olan sayıyı giriniz?

   1, 4, 9, 16, 25, 36, 49, (….)

2) Bir sayı düşünün, karesini alın, uzaktan 7 alın, 3 ile çarpın, 8 ekleyın ve son olarak bunu 7 ile bölün. 5.'nın cevabını verebilmek için. Başlangıç sayısı neydi?
3) Eğer (ASIA) için kod (CTJC) ise, (AFRICA)'nın kodu nedir?
4) Kutunun içinde 8 tane kırmızı, 7 tane mavi ve 6 tane yeşil top var. Bir top rastgele çıkarılıyor. Tahmini olarak onun kırmızı veya yeşil'e yakınlığı nedir?
5)

	A	**B**	**C**	**D**	**E**	**F**
1	5	10	2	4	0	0
2	2	5	7	12	0	0
3	9	3	6	8	0	0
4	4	8	15	11	0	0

1. Adım: A3'u C4 ile çarp ve E1'e ekle.

2. Adım: B4'u D2 ile çarp ve F3'e ekle.

3. Adım: eğer E1'ın değeri F3'tan büyük ise içeriklerini değiştir. (takas yap). Yoksa onları şuanki durumlarından ayır.

4. Adım: B1'den F3'u çıkar sonra C1'e 1ekle.

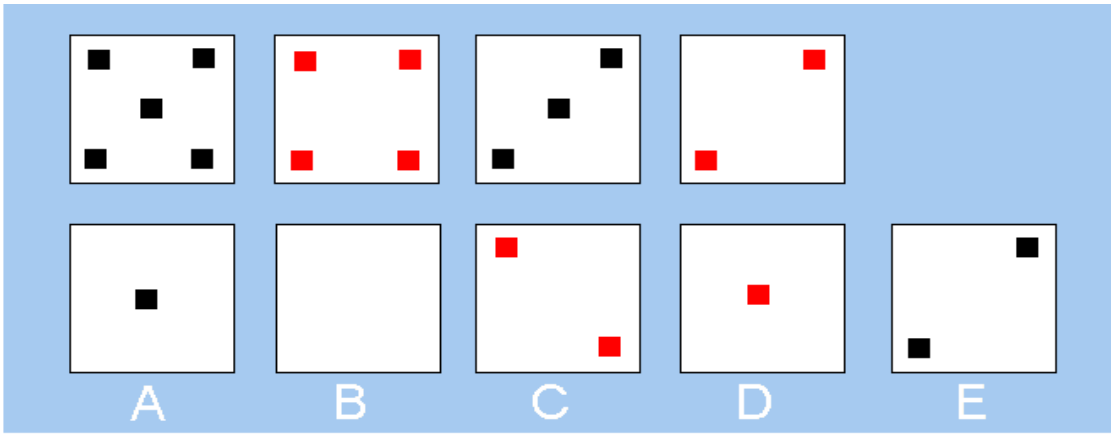
Adım 4'ü C1 değeri 5 olana kadar tekrarla.

Bu durumda F3'ün değeri nedir?

6) arkadaş birbirine sırayla oturmuş. Ve her biri farklı renkte T-şört giymiş. Ahmad, Ali'nin son tarafında oturmuş, ama onu geçmesi gerekli veya zorunlu değil. Beyaz T-şörtlü yeşil T-şörtlünün sağ tarafında oturmuş. Mavi T-şörtlü Ömerin sol tarafında oturmuş. Ömer de Ali'nin sol tarafında oturmuş.

Onların sıralamaları nasıl?

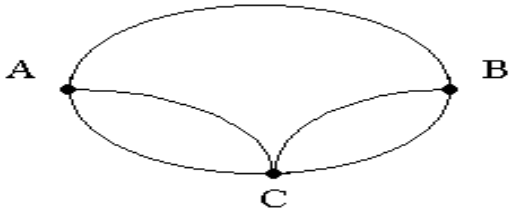
7)



8) Harf dizisindeki eksik harf nedir?

a z b ? c v

9)



Ali her hangi bir siparişte B ve C şehirlerini ziyaret etmek istiyor. Diagramda gösterilen sokak bu şehirleri onun evine bağlıyor. A'dan başlayıp tekrar A'ya dönebilmesi için kaç tane farklı yol kullanabilir?

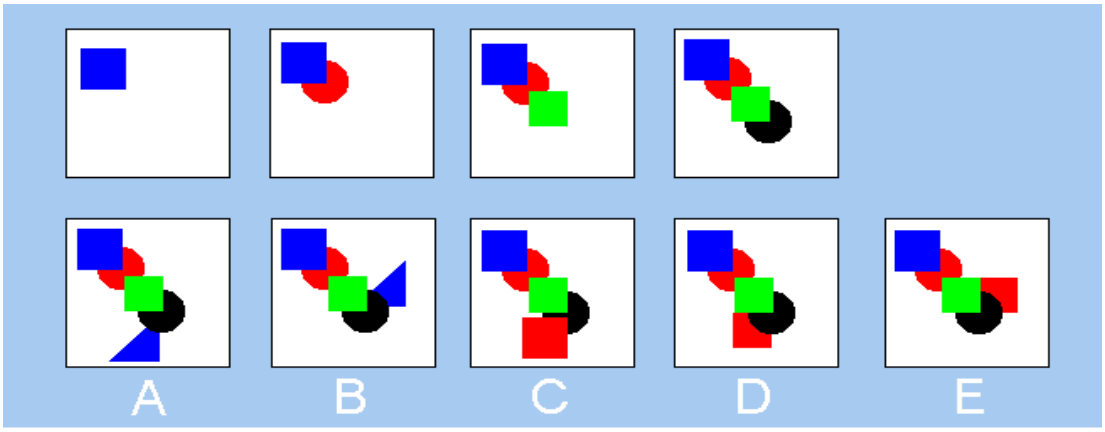
B ve C'nin içinden gidebilir (ancak her birinin içinden sadece bir defa gidebilir) ve aynı yönde bir yoldan iki defa geçemez?

10) Aşağıdaki kesirlerin hangisi en büyüktür?

7/8, 13/16, 31/40, 63/80

11) Eğer DAC=12, BAC=9, VE DCCh=25. (AMAR)'ın değerleri nedir?

12)

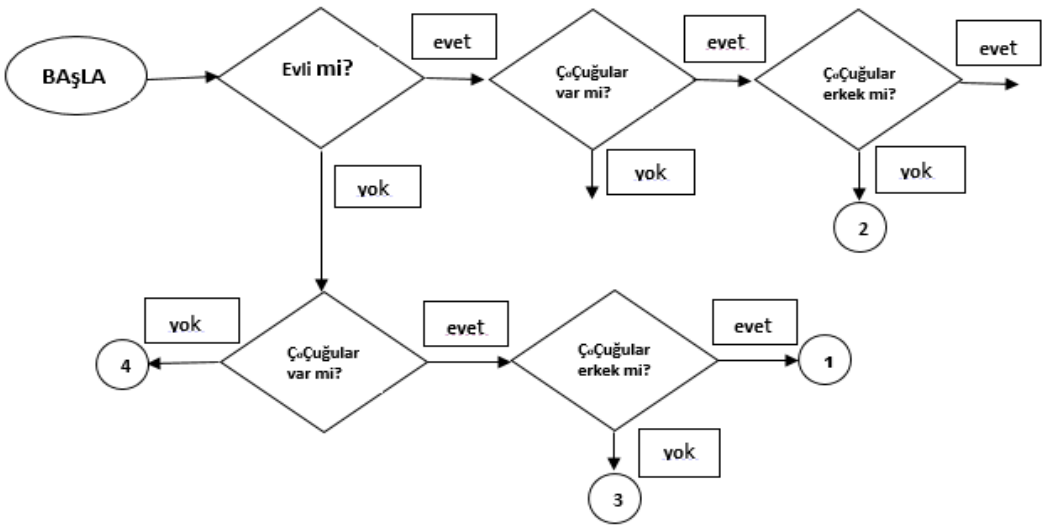


13) A, B'dan iki yaş büyüktür, ikisi de C'nın iki kat yaşında. Eğer A, B ve C'nın toplam yaşları 27 ise, B kaç yaşındadır?

14) Jklmsqdr Jklmsqdr
Olgbehiz Olgbehiz
Cqmdfdoj Cqmdfdoj
ydfsjnop ydfsjmop
Klvmojis Klvmojis
qaspoany qaspoany

Yukarıdaki 6 satırda bulunan kümeslerin içeriği özdeş (benze) karakterlerden oluşuyor. Ama bir satırdaki setlerin içindeki karakterler farklıdır. Hangi satır farklıdır?

15) Aşağıdaki akış şemasını bir müşteri için düşünün:



Birindeki şahıs:

A. Evli, çocuklu.
B. Evli, en az bir oğlu var.
C. Evli değil, en az bir kızı var.
D. Evli değil, en az bir oğlu var.
E. Evli değil, hiç çocuğu yok.