DEVELOPING A DISTANCE EDUCATION COURSE USING JAVA AND JAVASCRIPT LANGUAGES

NAZİME TUNCAY

Eastern Mediterranean University

Developing a Distance Education Course Using the Facilities of Java and JavaScript Languages

by

Nazime Tuncay

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements
for the Degree of

Master of Science
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
July 2002

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Zeka MAZHAR

Director

I certify that this thesis satisfies all the requirements as a thesis for the Degree of Master of Science.

Prof.Dr.Ahmet ERİŞ

Chairman

We certify that we have read this thesis and that in our opinion, it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Prof.Dr.Mehmet C.OKUR

Supervisor

Examining Committee

1. Prof Dr.Mehmet C.OKUR

2. Assoc.Prof Dr.Rza Bashirov

3. Assoc.Prof.Dr.Rashad. Aliyev

CERTIFIED TRUE PHOTOCOPY

Director
Office of Higer Education and Foreign Relations
Ministry of National Education, Youth and Sports
Turkish Republic of Northern Cyprus.

Nicosia 19 / 07 / 2012

MINISTRY OF NATIONAL EDUCATION, YOUTH AND SPORTS * T.R.N.C. *

Ahmet Simitcioğlu
Inspector
Higher Education and
Foreign Affairs Department

T. C. LEFKOŞA BÜYÜKELÇİLİĞİ
TURKISH EMBASSY - NICOSIA

Regardless of the text itself, it is hereby certified that the signature appearing above is that of... Ministry of Education, Youth and Sports Turkish Republic of Northern Cyprus and seal is that of the said Ministry.

NICOSIA 24 / 7 / 2012
A1273265.

Fulya T. ÇANGA
İkinci Katip
Second Secretary

T.C. LEFKOŞA BÜYÜKELÇİLİĞİ

True photocopy of the original.

Kuzey Kıbrıs Türk Cumhuriyeti
104
TASDİK MEMURU

Doğa Zeki

CONTENTS

ABSTRACT.....v

ÖZET.....vi

Acknowledgements.....vii

List of Table.....ix

List of Figures.....x

1 INTRODUCTION.....1

2 DISTANCE EDUCATION.....4

2.1 Drawbacks of Traditional Classroom Education 4

2.2 Distance Education and Distance Learning.....4

2.3 Distance Learners.....6

2.4 Comparison Between Distance Learners and Traditional Learners...7

2.5 Distance Education Delivery Systems.....8

2.6 Distance Education in Turkey.....9

2.7 Distance Education in T.R.N.C 10

3. ABOUT THE JAVASCRIPT LANGUAGE.....11

3.1 What is JavaScript?.....11

3.2 Capabilities of the JavaScript Language.....12

3.3 JavaScript is Different than Java.....12

4. ABOUT THE JAVA LANGUAGE.....14

4.1What is Java?.....14

4.2 Java is Object Oriented.....16

4.3 Java Exceptions.....16

4.4Java is Multithreaded.....17

4.5Graphical Programming with Java.....17

4.5.1 Label.....22

4.5.2 TextField.....23

4.5.3 TextArea.....24

4.5.4 TextComponent.....25

4.5.5 Buttons.....26

4.5.6 Checkbox and CheckboxGroup.....27

4.5.7 Choice.....28

4.5.8 List.....29

4.5.9 Containers.....31

5. ONLINE JAVA COURSE 35

5.1 The relevance of Java and JavaScript in Distance Education Course Development.....35

5.2 Preparation of the Course.....35

5.2.1 Course Outline.....36

5.2.2General Structure of the Course.....37

5.2.3 Using the Graphical Programming Capabilities of Java.....42

5.2.4Transfering Text, Applets and Applications to the Microsoft FrontPage.....48

5.2.5 Including the Images and the Outputs of Java Programs to the Online Java Course....52

5.2.6 The Preparation of Exercises....53

5.2.7 The Preparation and Evaluation of Quizzes....57

6 CONCLUSION....70

7 APPENDIX Sample HTML Codes....71

REFERENCES....88

ABSTRACT

This thesis is about preparing a distance education course using Java and JavaScript language facilities.

This thesis makes the following contributions: First, main concepts of the Distance Education and the Distance Learning have been presented and then, a brief summary of the Java and JavaSript language features are given. After that, the Java language features that are relevant to online course development have been considered. An online Java course has been developed and the relevant features of the Java and JavaScript languages have been discussed.

ÖZET

Bu tez, Java ve JavaScript dillerini kullanarak, uzaktan eğitim dersi hazırlamakla ilgilidir.

Bu tezin başlıca katkıları şunlardır: Öncelikle uzaktan eğitim ve eğitimci kavramları anlatılmış, ve JavaScript dilinin kısa bir özeti verilmiştir. Sonra, uzaktan eğitim dersinin hazırlanmasında kullanılabilecek olan Java dil özellikleri anlatılmıştır. Tez kapsamında bir uzaktan Java dili eğitimi dersi hazırlanmış ve bununla ilgili olan Java ve JavaScript dillerinin özellikleri tartışılmıştır.

Acknowledgements

I would like to thank to my supervisors Assoc. Prof. Dr. Reneta P. Barneva and Prof. Dr. Mehmet C. Okur for their guidance and continuous encouragement during my study.

Thanks to my mother Katriye Tuncay for her endless love, support and encouragement during my study.

List of Table

Table 3.1 Comparisons between Java and JavaScript languages.....13

List of Figures

Figure 4.1 Compressed Hierarchy of the AWT 21

Figure 5.1 Back button 38

Figure 5.2 Up button 38

Figure 5.3 Forward button 38

Figure 5.4 Chapter 4.2 from the online notes 38

Figure 5.5 Pascal label 39

Figure 5.6 Comparison with the Pascal language 39

Figure 5.7 Summary of chapter 1 40

Figure 5.8 Outline of chapter 1 41

Figure 5.9 NazButton applet 42

Figure 5.10 Output of the NazButton applet 43

Figure 5.11 Frames on the online Java tutorial 44

Figure 5.12 Buttons arranged with GridLayout 45

Figure 5.13 WhatIs applet 46

Figure 5.14 Output of the WhatIs applet 47

Figure 5.15 Note applet 48

Figure 5.16 Output of the Note applet 48

Figure 5.17 Surrounding the chapter headings with black rectangles 49

Figure 5.18 Changing the font of the heading when pointed by mouse 50

Figure 5.19 HTML Codes of Welcome HTML on the Microsoft Frontpage 51

Figure 5.20 Normal View of the Welcome Applet on the Microsoft Frontpage 51

Figure 5.21 Output of the Welcome Applet.....52

Figure 5.22 Image of the flowers application output using Paintshop Pro7.0.....52

Figure 5.23 Image of the definition applet output using Paintshop Pro7.0.....53

Figure 5.24 Exercise for chapter 2.....54

Figure 5.25 End of the exercises for Chapter 2.....55

Figure 5.26 Answers of the Exercises in Chapter 2.....56

Figure 5.27 End of the Answers of the Exercises in Chapter 2.....57

Figure 5.28 Preperation of Quiz with Hot_Patatoes.....58

Figure 5.29 Exporting the questions prepared to HTML.....59

Figure 5.30 Saving the files created at the end of the process of exporting to HTML.....60

Figure 5.31 Message that appears after saving the files that JBC created.....60

Figure 5.32 First question of the multiple-choice quiz from the chapter 1.....61

Figure 5.33 Response given to a wrong answer.....62

Figure 5.35 Response given to a correct answer.....63

Figure 5.36 Feedback given to a wrong answer.....64

Figure 5.37 Finding the correct answer after second trial.....65

Figure 5.38 Finding the correct answer after fourth trial.....66

Figure 5.40 Chapter 1 short answer quiz.....67

.Figure 5.41 Response given to a correct answer in the short answer quiz.....68

Figure 5.42 The action of the "Show Answer" button.....69

Figure 1 First page of the Javapage.....73

Figure 2 Second page of the Javapage.....74

Figure 3 Third page of the Javapage.....75

Figure 4 First Page of the Contents.....76

Figure 5 First page of the exercises of Chapter1.....77

Figure 6 Second page of the exercises of Chapter1.....78
Figure 7 Third page of the exercises of Chapter1.....79
Figure 8 Fourth page of the exercises of Chapter1.....80
Figure 9 Fifth page of the exercises of Chapter1.....81
Figure 10 Multiple-choice quiz of Chapter2.....82
Figure 11 First Page of the Quiz2c.....82
Figure 12 Second page of the Quiz2c.....83
Figure 13 Third page of the Quiz2c.....84
Figure 14 Fourth page of the Quiz2c.....85
Figure 15 Fifth page of the Quiz2c.....86
Figure 16 Sixth page of the Quiz2c.....87
Figure 17 Seventh page of the Quiz2c.....88
Figure 18 Eight page of the Quiz2c.....89
Figure 19 Quiz2w.....91

1 INTRODUCTION

Day by day, **Distance Education** is taking more place in our lives. Universities around the world are adding Distance Education programs to their schedules and the number of distance learners are getting increased every day. Today, one has an alternative of taking a degree on a specific subject at his/her home, if he/she has a computer connected to the Internet at home instead of going to school. There are numerous online courses available on most subjects on the Internet. There exists also many sites which help instructors to build distance education courses like *ecolledge* and *webct*. The main question that everybody asks is: "What is the best way to develop Distance Education Lessons?"

JavaScript is an easy to learn scripting language. It makes the web pages attractive and appealing. Rather than presenting traditional static information, which is what happens when HTML alone is used, web pages can be brought to life by using JavaScript. However, JavaScript is a small language with limited features and less reliable compared with Java. There exist lots of sites on the web that provide us with JavaScript programs to prepare distance education courses. One of these is *Hot-Potatoes* where quizzes can be prepared.

Java is generally regarded as the most strategic language to develop applications for the web. Java is an object-oriented program that provides better quality software. It breaks down a programming code into smaller components to make it

easier to produce programs. The Java language becomes increasingly important every day due to its qualities of security, being robust, and being portable.
With Java, complex, interactive programs can be written which is important in business world. Many sites provide us Java buttons with animation and many other exciting Java applets to include on our homepages. There exists lots of sites on the web which provide us with Java programs to prepare distance education courses.

Last semester, using the previous version of the online Java course notes, Java lectures are given in the department of Applied Mathematics and Computer Science at EMU. At the end of the semester an evaluation form for the Java lecture notes were distributed to the students. The findings of this evaluation are used to improve the structure of the course. On the online Java course some of the basic features of Java and JavaScript is used. In this thesis this online course is investigated.

In the online Java course, *Microsoft FrontPage and Hot-Patotoes* access JavaScript facilities. They provide the users with the necessary JavaScript programs to include into the web pages.

This thesis consists of four chapters. First chapter is focused on *Distance Education.* The first part, discusses the drawbacks of the traditional classroom education. Second part, presents the concepts of distance education and distance learning. Third part deals with distance education delivery systems.

The second chapter is about *JavaScript.* First part introduces the JavaScript language. In the second part, the capabilities of the JavaScript language are explained. Then, the difference of JavaScript from Java is explained.

In the third chapter, basic features of *Java* language are presened. In the first part the definition of Java is given. In the second part, it is claimed that Java is safe and

secure. In the third, part it is stated that Java is object oriented. Fourth part deals with Java exceptions. Fifth part explains that Java is multithreaded. Sixth part is about graphical programming with Java. A brief definition of Label, TextField, TextArea, TextComponent, Buttons, Checkbox, CheckboxGroup, Choice, List and Containers and their methods is given.

In the fourth chapter, the *online Java course* is discussed. Here, first the relevance of Java and JavaScript in distance education course development is explained. Then, in part two, steps of preparing the online Java course are considered and course outline is given. General structure of the course is explained. Some examples, in which graphical programming capabilities of Java is used, are given. The methods followed for transferring the text, the applets and the applications to the Microsoft FrontPage are explained. Then, the software used for the preparation of the exercises and quizzes for the course are presented. The evaluation processes of the quizzes are also explained in this part.

The appendix contains some of the HTML codes from the online Java course.

DISTANCE EDUCATION

2.1 Drawbacks of Traditional Classroom Education

In **traditional classroom education**, students have no immediate access to research tools such as libraries. The scope of what can be discussed in a given class period is limited to what lies inside the heads of individuals in the class and whatever tools happen to be in the classroom.

Specially in rural areas, finding the talent to teach classes can be very difficult. Even when such talent is found, often the small number of students makes holding a class cost-prohibitive. Many rural states have tried to solve this problem through video-conferencing. Using video-conferencing technology, students come together in a single or, in well-funded instances, a few locations to learn from a teacher in another location. Video-conferencing requires a lot of expensive equipment plus many support personnel to make a class happen.

2.2 Distance Education and Distance Learning

The history of Distance Education goes back more than 100 years in Europe, Africa, and Asia. Distance Education is part of all degree types, from the A.A. to the Ph.D., and is an option in most majors, and at hundreds of universities worldwide.

Distance Education is an instructional delivery system which connects learners with educational resources. It is a kind of education which takes place when students and instructors are far away. When the method and technologies used are appropriate to the instructional tasks, there is student-to-student interaction, and when there is timely teacher-to-student feedback Distance Education can be very effective. Distance Education speeds the process of transferring the knowledge of instructors to their students, enabling the learning to take place faster than the traditional education.

Distance Education is for people who want to learn a new skill, or just pick up a few new ideas for the fun of learning. Additionally, Distance Education is an exciting and growing part of public and private schools from elementary level through high school in many areas such as math, science, and languages. Shared courses offered via satellite, fiber optic cable and videocassette connect schools in the same town, or great distances apart. Courses can be one-way or two-way. Also, pre-university Distance Education, commonly known as home-schooling, is a type of Distance Education. Typically, a home-schooled child is taught by his or her parents, however, sometimes, the child is tutored at home in part or completely via modem or TV course.

For most students, Distance Education remains primarily a books and paper proposition, as well as an independent study effort, but is made more convenient with the delivery of information via the Internet, the web, online services, telephone, TV, satellite courses, VHS tapes, cassette tapes, records, radio and CD-ROM. Many Distance Education programs are correspondence courses where books, materials, and coursework are exchanged through the mail. However, there are courses which are completely electronic, such as interactive multimedia courses, and group learning in virtual classrooms. These options are growing in number and scope. Distance Education also involves finding alternative ways to earn college credits, such as through

challenge examinations. Mature students are particularly adept at testing out of courses by demonstrating subject mastery through a single examination. Earning college credit for life and work experience through the creation of a student portfolio is another option common to Distance Education programs. Many people complete long unfinished Bachelor's degrees through the use of portfolios.

There is scheduling flexibility in Distance Education. Many courses start when the student is ready, not just in September and January. Students can learn at any time of the day or night and there is no waiting for when a certain course will be offered at the traditional school. Distance Education students can take any course anytime, and sometimes, even help design the course they want to take. Some Distance Education programs have residency requirements ranging from a single week-end to short campus periods.

Distance Learning is the process of learning, which takes place during Distance Education, it is the result of Distance Education. Distance Education and Distance Learning terms are two important and different concepts. Distance Learning is the volitional control of learning by student rather than distance instructor. What is important is how the Distance Education can take place so that better Distance Learning can be achieved.

2.3 Distance Learners

Distance Learners could be interested in changing careers or they might simply want to expand their knowledge base for work or personal reasons. They might want to expand their cultural background, learn a new language, or start a degree program that was postponed due to family or career needs.

The people who need Distance Education are usually adults, for whom the convenience and appeal of earning college credits on all degree levels with minimal domestic disturbance is a great attraction. Parents, caregivers, working adults, and all busy people find Distance Education to be an excellent way to combine studies and living. People who prefer to study solo can do so with Distance Education, whilst those who enjoy group learning can find Distance Education classes online. People who live in remote areas or where weather can be extreme, those who have disabilities that make commuting difficult and many others just for the fun of it turn to Distance Education as the only practical way to earn college credits.

Most Distance Education students are mature adults over the age of 25. Some Distance Education programs cater exclusively to people over age 30. Others have no age limits, and will admit otherwise qualified teens and gifted children.

2.4 Comparison Between Distance Learners and Traditional Learners

Students who take Distance Education courses are successful since they are voluntarily seeking for further education and they are generally older than the students who take the traditional courses. These students who take distance education courses are older and most of them works during traditional classroom hours. They require flexible learning schedules and demand professional development opportunities and classes to help them keep up with today's ever-changing work environment.

Learners are more motivated if they are in frequent contact with the instructor. The use of technologies such as fax machines, computers and telephones can also provide learner support and interaction opportunities. Distance Education courses to the learner include accessible training to students in rural areas. Students may complete

their course of study without suffering the loss of salary due to relocation. Students are exposed to the expertise of the most qualified faculty.

Distance Education courses and its technologies require extensive planning and formative evaluation is necessary for an effective distance education couse. Learners benefit significantly from a well-designed syllabus and presentation outlines. One who prepares a distance education course should be able to use the equipment and should be aware of those techniques proven effective in the Distance Education environment. Like traditional learners distant learners require support and guidance to make the most of their distance learning experiences. Researches done on this area, show that learners value timely feedback regarding course assignments, exams, and projects and learners benefit significantly from their involvement in small learning groups. These groups provide support and encouragement along with extra feedback on course assignments. Most importantly, the groups foster the feeling that if help is needed it is readily available.

2. 5 Distance Education Delivery Systems

There are two categories of Distance Education delivery systems, synchronous and asynchronous.

Synchronous instruction requires the simultaneous participation of all students and instructors. The advantage of synchronous instruction is that interaction is done in "real time." Forms of synchronous delivery include Interactive TV, audiographics, computer conferencing, IRC, and MOO.

Asynchronous instruction does not require the simultaneous participation of all students and instructors. Students do not need to be gathered together in the same location at the same time. Rather, students may choose their own instructional time

frame and gather learning materials according to their schedules. Forms of asynchronous delivery include email, listservers, audiocassette courses, videotaped courses, correspondence courses, and WWW-based courses.

Asynchronous instruction is more flexible than synchronous instruction. Moreover, in the case of telecommunications such as email, asynchronous instruction allows and even may encourage community development. The advantages of asynchronous delivery include student choice of location and time, and (in the case of telecommunications such as email) interaction opportunities for all students. A disadvantage to consider with email-based interaction is the considerable written exchange, which could really pile up.

2.6 Distance Education in Turkey

Technologies used to deliver distance education programs in Turkey are designed to reach the masses. The two primary forms of distance education are the **Open Education Faculty (OEF)** at *Anadolu University* in Eskisehir and the **Open High School (OHS)** through the Ministry of National Education.

The OEF, which been operating since 1982, delivers undergraduate degree programs and other programs to Turks throughout Turkey, Europe, and the Turkish Republic of Northern Cyprus. The OHS is a widespread secondary education program, which has been operating since 1992. The purpose of the OHS is to allow traditional and non-traditional students, who for one reason or another have not completed secondary school, with an opportunity to earn a high school diploma. The OHS curriculum is the same as for traditional high school students. The technologies for both programs include specially designed textbooks and other printed materials including newsletters and bulletins, television and radio broadcasts; technologies for the OEF

include videotapes and face-to-face lectures at universities throughout the country. Examinations for both programs are offered in a variety of locations, usually in cities and larger towns. Computer-aided instruction, while not typically used in distance education, is being introduced into schools nationwide.

There are online degree programs also in *METU(Middle East Technical University)*, and Anadolu University for about 2 years.

2.7 Distance Education in T.R.N.C

There are recent studies to develop distance education courses in T.R.N.C. One example is the *EMU(Eastern Mediterranean University) Online*. (http://emuonline.emu.edu.tr) One can find information about currently offered online courses by clicking on the Demo Page link. Also there is a 2-year Information Management online program which is approved by the Turkish Higher Educational Council (YOK) and will start in October 2002.

3 ABOUT THE JAVASCRIPT LANGUAGE

3.1 What is JavaScript?

JavaScript is a simple scripting language for HTML. Script languages generally take longer to process than compiled languages, but are very useful for shorter programs. JavaScript is an interpreted language (i.e. it does not need to be compiled) that has characteristics of Object Oriented languages, e.g. objects, properties, methods. JavaScript is an **object-based language**, but is not an object-oriented language. It has a syntax based on C. JavaScript language is a programming language that adds *interactivity* to our Web pages.

JavaScript comes in two flavors, the popular **Client Side** version that runs in a users browser, or the **Server Side** version that can run on Netscape Servers. There are three browsers that support Client Side JavaScript: Netscape Navigator (since version 2), Microsoft Internet Explorer (since version 3) and Opera.

The browsers themselves extend the JavaScript language with additional objects, for example images, windows, frames. These objects, their methods and properties are part of what is known as the **Document Object Model (DOM)**. Each browser has its own subtle version of the DOM. As browsers get mature and new versions are released more and more features are added to the browsers DOM.

3.2 Capabilities of the JavaScript Language

JavaScript creates an active user interface, has validated user input on forms, creates custom HTML pages on the fly, and controls Web browser. It is used in web site development to do things such as: automatically change a formatted date on a web page, cause a linked-to page to appear in a popup window, cause text or a graphic image to change during a mouse rollover.

JavaScript allows client side processing to occur within a document without the need for processing on the server. For example, JavaScript can be used to validate a form before it is sent to the server. This can avoid lengthy delays whilst validation is carried out on the server, only for a response to return stating that the form has been completed incorrectly. JavaScript can do this **before** the form is sent. JavaScript can be **useful for very small applications** that run on the web client or server.

JavaScript cannot read or write files on the local system or on the server. Also, it cannot start/stop applications on local system or on server. What's more, JavaScript cannot run in all browsers.

3.3 JavaScript is Different than Java

JavaScript uses some of the same ideas found in Java, the compiled object_ oriented programming derived from C++. However, Java and JavaScript are two different languages. JavaScript is a simple scripting language for HTML created by Netscape. Whereas, Java is a full complete programming language and platform created by Sun Microsystems and can be used to write any standard application, not just HTML. JavaScript, which originated at Netscape, is interpreted at a higher level, is

easier to learn than Java, but lacks some of the portability of Java and the speed of bytecode.

It is possible for the two languages to communicate with each other in Netscape Navigator using a technology called LiveConnect. Some of the differences which web designers and programmers must know are summarized in Table 3.1[17]

Feature	Java	JavaScript
Creators	Sun Microsystems	Netscape
Type of Language	Byte Compiled Object Oriented	Smaller Object Oriented type Interpreted Scripting Language
Stand alone applications	Can be created	Cannot be created
Applet Creation	Possible	No
File input/output	Can read and write files from and to server but nothing can be written on clients machine	Can write cookies to clients machine, and embed *.js files from the server
Run Time Libraries	JVM (Java Virtual Machine) Java Runtime Lib required	JavaScript runtime libraries required
These libraries are built into Java and JavaScript compatible browsers		
Image Manipulation	Possible right up to pixel color level	No image manipulation possible, other than image replacement
Animation	Possible	Limited
Drawing	Possible	No
Multi threaded programs	Possible	No
Networking / Sockets	Possible	No
Cross Platform	More stable and reliable	Less Reliable
The problems with cross platform compatibility in JavaScript arise mainly due to differences in the object models implemented in the two major browsers - Netscape Navigator and Microsoft Internet Explorer		
Tools Need to write and test	Java Development Kit (JDK)	Nothing Special is needed, you can get by with Notepad in Windows
Compilation	Needs to be compiled before executing	No need for compilation

Table 3.1 Comparisons between Java and JavaScript

4. JAVA LANGUAGE

4.1 What is Java?

Java is a high-level, third generation programming language like C, Fortran, Smalltalk, Perl, Pascal and many others.

Java can be used to write computer **applications** that calculate spreadsheets, process words and fill out forms. Also computer applications such as playing games, chatting in real time, and getting continuously updated data and many other areas can be written by using Java.

Special programs called **applets** can be written in Java that can be downloaded from the Internet and played safely within a web browser. An important property of a Java applet is that it does not write to your hard disk without your permission. Thus, as it cannot write to an arbitrary address in the memory it could not introduce a virus into your computer.

Java was the first way to include inline sound and animation in a web page. With Java having music that plays in the background on a page or having cartoon style animations, having multi player interactive games is possible.

Java is an excellent teaching language and an excellent choice to learn programming. It is easy to become fluent with Java. The compiler-cycle is much shorter

than the other languages since it is interpreted. The run time environment provides automatic memory allocation and garbage collection thus there is less for the programmer to think about.

Java has various security features including the automatic garbage collection, the elimination of pointer arithmetic and the Java interpreter. These serve the dual purpose of making the language **simple for programmers** and **secure for users**. Both Java applets and applications are much safer in practice than code written in traditional languages. Even the code from trusted sources is likely to have bugs. But, Java programs are much less susceptible to common bugs involving memory access than are programs written in traditional languages. The Java runtime environment provides a fairly robust means of trapping bugs before they bring down its users system. Generally users of programming languages have many more problems with bugs than they do with deliberately malicious code. Although Java users are not protected from out and out malicious code, they are largely protected from programmer errors. Applets implement additional security restrictions, which are accomplished through the java.lang.SecurityManager class, that protect users from malicious code too. java.lang.SecurityManager class is subclassed to provide different security environments in different virtual machines. Unfortunately, implementing this additional level of protection does somewhat restrict the actions an applet can perform.

It is less possible to write a Java program that will crash your system since *Java provides automatic memory allocation and garbage collection*. Memory is allocated (there are constructors which allocate the memory on the heap but these are transparent to the reader) and then it is decollated by the garbage collector when it is no longer needed. Thus, users of Java can use the web without worrying that a Java applet will format their hard disk or introduce a virus into their system.

Java is a platform for application development. (A platform is some combination of hardware and system software.) Java programs are compiled to a bytecode format that can be read and run by interpreters on many platforms including Windows xx, Windows NT, and other Unix platforms.

4.2 Java is Object Oriented

Almost everything in Java is a **class**, a **method**, or an **object**. Only the most basic primitive operations and data types (int, for, while) are at a sub-object level. Java code can be executed in an environment that prohibits it from introducing viruses, deleting or modifying files, or otherwise performing data destroying and computer crashing operations. Java can be compiled on a fly with a **Just-In-Time compiler (JIT)**; to code that rivals C++ in speed.

4.3 Java Exceptions

Problems that arise in a Java program generate *exceptions* and *errors*. An *exception* is an object that defines an unusual or erroneous situation. This erroneous situation may be an exception that is thrown by a program or the runtime environment, and can be caught or handled appropriately. Note that, an error is similar to an exception, except that and generally represents an unrecoverable situation, and should not be caught.
In order to ignore the exceptions, it is necessary to indicate the compiler that you are aware of the problems that might occur by declaring the exception in the throws clause of the method. It can be thrown to indicate that the situation has occurred. Exception then can be designed to catch and process the exception when it is thrown.

4.4 Java is Multithreaded

Sequence of steps executed one at a time is called *thread.* A thread can perform task independent of other threads. Also threads can share access to objects. Operations such as compiling a program, printing a file, and receiving electronic mail messages over a network are performed concurrently by computers. Generally, programming languages provide only a simple set of control structures that enable programmers to perform one action at a time and then proceed to the next action after the previous one is finished. Java makes congruency primitives available to the programmer. Multithreading is the capability of executing a portion of a program concurrently with other threads. Each thread in the program represents a separate sequential flow that can occur with other threads. A single Java program can have many different *threads* executing independently and continuously. For example, three Java applets on the same page can run together with each getting equal time from the CPU with very little effort on the part of the programmer!

4.5 Graphical Programming with Java

Applet has a graphics object associated with it, which can be accessed within *paint( )* since it is provided as an argument. Anything that is wanted can be drawn with *Graphics* object's methods. Some of the methods of the Graphical class are:

- *void clearRect (int x, int y, int width, int height): clears a rectangle* by painting it in the background color. The arguments are integers. (x,y) denotes the coordinates of the upper left corner.
- *void draw3DRect (int x, int y, int width, int height, boolean raised):* draws a rectangular border of the specified dimensions, shaded in the background color to look three-dimensional. A boolean type has either of two values, called true and

false. If the argument raised has the value true the border is painted as if it were raised out of the surface and if it has the value false, it appears as if it is depressed.

- *void drawArc (int x, int y, int width, int height, int starAngle, int arcAngle):* draws an *unfilled arc* within the rectangle specified by x, y, width and height. The arc begins at the angle, in degrees, specified by the argument starAngle, and extends to arcAngle
- *void drawLine (int x1, int y1, int x2, int y2):* draws a *line* from the point with coordinates (x1, y1) to the point with coordinates (x2, y2).
- *void drawOval (int x, int y, int width, int height):* draws an *unfilled oval* within the bounding rectangle specified by the arguments. Here instead of the dimensions of the oval itself, the dimensions of the smallest rectangle, which can enclose the oval, are specified. The oval is drawn as large as it can be to touch the rectangle's edges at their centers.
- *void drawRect (int x, int y, int width, int height):* draws an *unfilled rectangle* starting from the position (x, y).
- *void drawRoundRect (int x, int y, int width, int height, int arcWidth, int ArcHeight):* draws an *unfilled rounded* rectangle. Here (x,y) gives us the position and width and height give us the height of the rectangle. ArcWidth and arcHeight specify the width and height, respectively, of the rounded corners.
- *void drawString (String s, int x, int y):* draws a String of characters with its baseline starting at coordinates x and y.
- *void fill3DRect (int x, int y, int w, int h, boolean raised):* draws a filled rectangular border of the specified dimensions, shaded in the background color to look three-dimensional.
- *void fillArc (int x, int y, int w, int h):* draws a *filled arc* within the rectangle specified by x, y, width and height.

- *void fillRoundRectangle(int x, int y, int w, int w, int h, int aW, int aH):* draws a *filled rounded* rectangle.
- *void setColor (Color c):* sets the current drawing color to the color specified in the argument.

The Color class's objects represent colors. This class contains thirteen constant values that can be used when you want: black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white or yellow. Colors are used to make programs nicer. Every color is created from an RGB (Red/Green/Blue) value. A RGB value is a triple of whole numbers(r, g, b) defining the amount of red, green and blue color. Java enables a programmer to choose from 256 x 256 x 256 colors.

The Font class represents the fonts used for the textual displays. There are five basic fonts: "Sans Serif", "Serif", "Monospaced", "Dialog", "Dialog Input" (Serif is the font style we know as Times New Roman). There is Font class constants, which are used to specify the style of the font. These are: Font.BOLD, Font.PLAIN, and Font.ITALIC. Java provides three classes that are useful for manipulating collections of dimension values: Point Class, Dimension Class and Rectangle Class. Java also has methods to draw outlined and filled ovals, arcs.

Components in Java are graphical user interface (GUI) widgets(window gadgets) like checkboxes, menus, windows, buttons, text fields, applets, and more. In Java, all components are subclasses of **java.awt.Component**. Component class is the super class of many AWT (Abstract Windowing Toolkit) objects. It is an abstract class, hence can not be instantiated. It can be painted on the screen as well as receive input events. GUI's are built from GUI components (also called widgets). GIU component is a visual object, which we interact via the mouse or the keyboard. **java.awt** provides classes that are used to create GUI components.

The following are some of the examples of the components:

- *Color getBackground( )*: returns the Color object used for the background color of this Component.
- *Rectangle getBounds( )*: returns a Rectangle object that describes the bounding rectangle of this object.
- *Font getFont( )*: returns the Font object that is currently being used by this Component.
- *Color getForeground( )*: returns the Color used for the foreground of this object.
- *Graphics getGraphics( )*: returns the Graphics object associated with this Component.
- *Point getLocation( )*: returns the anchor point of this component.
- *Dimension getMinimumSize( ), Dimension getMaximumSize( ), Dimension getPreferredSize( ), Dimension getSize( )*: returns the current width and height of this Component.
- *void hide( )*: makes this Component invisible.
- *boolean isShowing( )*: returns true if and only if this Component can be currently seen on the screen.
- *boolean isVisible( )*: tests for the property of visibility.
- *void paint(Graphics g)*: causes this Component to be painted, using the graphics object in the argument.
- *void repaint( )*: tells the system that this Component needs to be drawn as soon as possible.
- *void setBackground(Color c)*: sets the background color of this Component.
- *void setBounds(int x, int y, int width, int height)* and *void setBounds(Rectangle r)*: changes the size and location of this Component.

- *void setFont(Font f)*: sets the font of this Component to f.
- *void setForeground(Color c)*: sets the foreground color of this Component.
- *void setLocation(int x, int y)* and *void setLocation(Point p)*: moves this Component so that its anchor is at the coordinates given by the arguments.
- *void setSize(int w, int h)* and *void setSize (Dimension d)*: changes the size of this Component, keeping the current anchor.
- *void setVisible(boolean isVisible)*: if the argument is true, this method makes the Component visible on the screen, otherwise makes it invisible.
- *void update(Graphic g)*: called by the system when the screen needs to be redrawn.

The steps for adding components to the applet is: Declaring the component, initialising the component, adding the component to the layout. Component class is the super class of many AWT objects. It is an abstract class, hence cannot be instantiated. Component class represents something that has a position and a size. Fig. 4.1[13] shows the compressed hierarchy of java.awt.

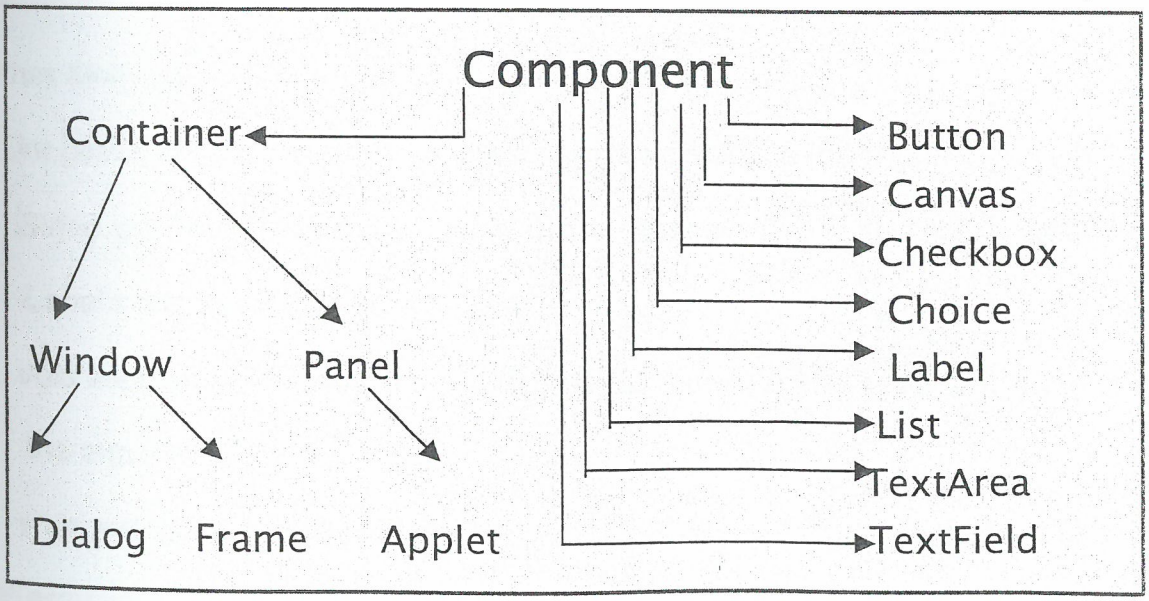

Figure 4.1 Compressed hierarchy of the AWT

Label, TextField, TextArea and TextComponent are widgets that can be used to dial with the display of the text.

4.5.1 Label

A Label is a component for placing text in a container. Its primary use is to label other components. java.awt.Label is the simplest component. A Label is one line of read-only text.

Constructors:

- *Label( )*: creates a Label object with center alignment and no text.
- *Label (String Label)*: creates a centre-aligned Label object displaying the label text given in the argument.
- *Label (String label, int alignment)*: creates a Label with the given text .The alignment of the text specified by using one of the three class constants Label.LEFT, Label.RIGHT or Label.CENTER.

Other Methods:

- *int getAlignment( )*: returns an integer corresponding to the alignment of this Label.
- *String getText( )*: returns a String object whose value is the same as that of this Label's text.
- *void setAlignment(int alignment)*: sets the alignment of this Label to the one given by the argument.
- *void setText(String label)*: sets the text of this label to the label given by the argument.

4.5.2 TextField

TextField is a component that presents the user with a single line of text. By default the text is editable to the user, but it is possible to make the text not editable by the user.

Constructors:

- *TextField( )*: creates a TextField object with empty text and a default width determined by the system.
- *TextField (int columns)*: creates a TextField with empty text, wide enough to hold columns characters.
- *TextField (String text)*: creates a TextField object with content given by the argument string.
- *TextField (String text, int columns)*: creates a TextField object with content and width given by the arguments.

Other Methods:

- *Boolean echoCharIsSet( )*: returns the current echo char, if any it has been set.
- *int getColumns( )*: returns the current width, in columns, of this TextField.
- *char getEchoChar( )*: sets the current echo character to that provided in the argument.
- *Dimension getMinimumSize( )* and *Dimension getPreferredSize( )*: returns a Dimension object with width and height set to the minimum or preferred size, respectively of this TextField.
- *int setColumns(int columns)*: sets the width of this TextField to be enough to hold column characters.
- *void setEchoChar(char c)*: sets the current echo character to that provided in the argument.

4.5.3 TextArea

TextArea is a component that presents the user with multiple lines.

Contructors:

- *TextArea( )*: creates a TextArea of default size, with empty text.
- *TextArea(int rows, int columns)*: creates a TextArea with empty text, with room for rows of text, each of which has room for columns characters.
- *TextArea(String text)*: creates a TextArea, containing the argument String.
- *TextArea(String text, int rows, int columns)*: creates a TextArea containing the argument String, of size specified by the rows and columns of arguments.
- *TextArea(String text, int rows, int columns, int scrollbars)*: allows the program to specify whether the TextArea will have the horizontal and / or vertical scrollbars.

Some More Methods:

- *void append(String str)*: changes the text of this object by appending the characters in the argument to the end of the current text.
- *int getColumns( )*: returns the current width, in columns, of this TextArea.
- *Dimension getMinimumSize( )* and *Dimension getPreferredSize( )*: returns a Dimension object with width and height set to that of the minumum or prefered size, respectively, of this TextArea.
- *int getRows( )*: returns the current height, in rows, of this TextArea.
- *int getScrollbarVisibility( )*: returns one of the four class constants, indicating which of the scrollbars are visible.
- *void insert(String str, int position)*: inserts the argument string into the current text, at location given by the position argument
- *void replaceRange(String str, int start, int end)*: replaces the current text between positons start and end with the argument string.

- *void setColumns(int columns)*: changes the number of columns in this TextArea.

4.5.4 TextComponent

TextComponent contains many of the methods common to the TextArea and TextField. Methods are the following:

- int getCaretPosition( ): returns the position measured in number of characters.
- *String getSelectedText( )*: returns a copy of the part of the text that has been selected.
- *int getSelectionEnd( )*: returns the position, in characters at the end of the selected text.
- *int getSelectionStart( )*: returns the position, in characters at the start of the selected text.
- *String getText( )*: returns a copy of the current contents of the component.
- *boolean isEditable( )*: returns true if the user can modify the text and false otherwise.
- *void select(int start,int end)*: selects all characters in the text, from position start to position end, highlighting the selected area in whatever way is appropriate for the system which it is running.
- *void selectAll( )*: selects and highlights all the characters in the text.
- *void setCursor(int position)*: moves the cursor to the indicated position in the text. The position argument must be greater than or equal to 0.
- *void setEditable(boolean canEdit)*: if the argument is true, the user can modify the text; if it is false, the user can not edit the text.
- *void setSelectionEnd (int position)* and *void setSelectionStart (int position)*: allows the program to set the end and start the selected portion of the text.
- *void setText(String text)*: sets the current text to be the one given by the argument.

4.5.5 Buttons

Buttons are simple objects. They are instances of the java.awt.Button class, a subclass of java.awt.Component. Buttons are created with the Button (String label) constructor. They are added to the layout and an ActionListener is added to it. When it is pushed an action event is generated.

Constructors:

- *Button( )*
- *Button (String label)*

Methods:

- *String getLabel( )*: returns the copy of the label text.
- *void setLabel (String label)*: allows us to change the label text.
- *void setText(String theLabel)*: changes the ButtonLabel.
- *public void addNotify( )*: creates the Button's peer object that is the native widget that has the appearance of a Windows button or a Mac button or a Motif button or whatever.
- *public void setActionCommand (String command), public String getActionCommand ( )*: modifies the command string sent along with the ActionEvent. By default this is the label of the button, but it can be changed.
- *public void addActionListener (ActionListenerl)*: registers another object as one, which should receive ActionEvents, fired by the Button.
- *public void removeActionListener (ActionListener l)*: deregisters the ActionListener object l so it will no longer receive action events fired by this Button.
- *public abstract void actionPerformed (ActionEvent e)*: does something as a result of the ActionEvent, the button press fired.

4.5.6 Checkbox and CheckboxGroup

A Checkbox is a graphical user interface that has an on state (true) and an off state (false). A CheckboxGroup can contain a set of Checkboxes in which only one may be true at a time.

Constructors of Checkbox( ) are the following:

- *Checkbox( )*: creates a Checkbox with no label and false state.
- *Checkbox (String Label)*: constructs a Checkbox with the given label and initial false state.
- *Checkbox (String label, boolean state)*: constructs a Checkbox with the specified label and state.
- *Checkbox (String label, boolean state, CheckboxGroup group), Checkbox (String label, CheckboxGroup group, boolean state)*: specifies the state and the label, and in addition specifies the group of Checkboxes to which this one will belong.

Other Methods of the Checkboxes are:

- *CheckboxGroup getCheckboxGroup( )*: returns the Checkbox group, if any, to which this Checkbox belongs.
- *String getLabel( )*: paired with setLabel( ), returns a string containing the label text of this Checkbox.
- *boolean getState( )*: returns the current state (true or false) of this Checkbox.
- *void setCheckboxGroup (CheckboxGroup group)*: sets the Checkbox group to which this object belongs.
- *void setLabel (String label)*: sets the label of this Checkbox.
- *void setState (boolean state)*: sets the state of this Checkbox.

Constructor of CheckboxGroup is:

CheckboxGroup

Methods of CheckboxGroups are:

- *void setSelectedCheckbox (Checkbox c)*: returns the Checkbox in this group, if any, which is in the true state.
- *Checkbox getSelectedCheckbox( )*: has the same effect as if the user had clicked on the Checkbox in the argument- it sets the state of c to true and turns of any other true Checkbox in this group.

4.5.7 Choice

The Choice is used to select a single string via pop-up list of choices. Only one choice can be selected at a time.

Constructor of Choice is :

- *Choice( )*

Methods of Choice are:

- *void add(String item)*: adds a new item with the given label to the list of this Choice object.
- *int getItemCount( )*: returns the number of items currently in this Choice object.
- *String getItem(int index)*: returns the text of the item in position index in the list.
- *int getSelectedIndex( )*: returns the index of the current selection.
- *String getSelectedItem( )*: returns the text of the current selection.
- *void insert(String item, int index)*: inserts a new item string in the list in the given position.
- *void remove(int index)*: removes the item at the given position.
- *void removeAll( )*: removes all items from this Choice.

- *void select(int index)*: selects an item by its position.
- *void select(String item)*: selects an item ,if any, which matches the string given in the argument.

4.5.8 List

The List component allows the selection from a scrolling list of Strings. The List can be set up to allow either one selected item or multiple selected items. Unlike Choice List allows the user to see more than one object at a time. The List take up more screen real estate than the Choice.

Constructors are the following:

- *List( )*: creates a default-sized List object in the single selection mode
- *List(int rows)*: creates a List object in single-selection mode, with a number of rows given by the argument. These rows are the visible to the user.
- *List(int rows, boolean multipleSelections)*: creates a List object with the specified number of rows. If the boolean argument is true, the List is in multiple-selection mode, and if the argument is false, the created list is in multiple selection mode.

Methods are:

- *void add(String item), void add(String item, int index)*: adds a new item with the given label to this List object. If the index is not specified, the item is added to the bottom; otherwise the existing items at and after position index are shifted down one row to make room for the new item.
- *boolean allowsMultipleMode( ):* returns true if this List is in multiple-selection mode.
- *void removeAll( )*: removes all items from this List object.
- *int getItemCount( )*: returns the number of items in this List.
- *void delItems(int start, int end)*: removes the range of items from this List.

- *void deselect(int index)*: turns off the selection of the item in position index.
- *String getItem(int index)*: returns the text of the item in position index in the List.
- *Dimension getMinimumSize( ), Dimension getMinimumSize(int rows), Dimension getPreferredSize( ), Dimension getPrefferedSize(int rows)*: returns a Dimension object with width and height set to that of the minimum or prefered size, respectively, of this List.
- *int getRows( )*: returns the number of rows in this List object.
- *int getSelectedIndex( )*: returns the index of the current selection. This method returns –1, if no item is currently selected or if more than one item is selected.
- *int getVisibleIndex( )*: returns the argument that was used in the last call to makeVisible() or –1 if no call to makeVisible() has been made previously.
- *boolean isIndexSelected( )*: returns true if the item in position index is selected; otherwise returns false.
- *boolean isMultipleMode( )*: returns true if the List is in multiple-selection mode; otherwise returns false.
- *boolean isIndexSelected( )*: returns true if the item in position index is selected; otherwise returns false.
- *void remove(int index)*: removes an item in position index.
- *void remove(String item)*: searches for an item matching the String argument, and, if there is one it is removed.
- *void removeAll( )*: removes all items from the List.
- *void replaceItem( )*: replaces the item in position index with the given by the argument newItem.
- *void select(int index)*: selects the item in position index.
- *void setMultipleMode(boolean m)*: returns the multiple-selection mode on if the argument is true and sets the mode to single-selection if the argument is false.

4.5.9 Containers

The Container class is a subclass of the Component. Thus every Container object has access to all the Component methods. Containers are used in order to group Components. There is no limit on the number of Components a Container may hold. A Container has referances to each Component it contains and each Component has a reference to the Container (its parent) which it belongs.

An applet is a container. Applets provide a ready-made container and a default LayoutManager, a FlowLayout. Other containers include windows, frames, dialogs, and panels. Every container has a LayoutManager that determines how different components are positioned within the container. Different LayoutManager classes use different rules to place components. java.awt. LayoutManager is an interface. Five classes in the java packages implement it: FlowLayout; BorderLayout; CardLayout; GridLayout ; GridBagLayout.

Container Organization Methods are:

- *void add(Component c):* adds the Component in the argument to the end of the list of the Components already in this Container.
- *void add(Component c, int position):* adds the Component in the argument to the position specified in the argument.
- *void add(String name, Component c):* adds the Component in the argument according to the convenience of some LayoutManagers like BorderLayout and CardLayout. Containers can not be added to itself ! And, if one attempts to add Component which is already there, it will be removed and reinserted, generally in a different position.

- *Component getComponent(int position):* returns the Component in the indicated list of Components in this Container.
- *Component getComponetAt(int x, int y):* returns the Component at the local coordinates (x, y).
- *Component getComponetAt(Point p):* returns the Component at the point p.
- *int countComponents( ):* returns the number of Components currently in the object.
- *Container getParent( ):* returns the Container that holds this Component.
- *Boolean isAnchestorOf(Component c):* returns true if this Container is parent of c. Otherwise returns false.
- *void remove(int index), void remove(Component c):* remove a Component from this Container's list. Note that these methods look only at the top-level list and do not look down the containtment hierarchy.
- *void removeAll( ):* removes all Components fron this Container.

AWT defines two different kinds of containers, panels and windows. Panel is used to store a collection of objects without creating borders. It subdivides the drawing area into separate rectangular pieces. Panels are subclasses of java.awt.Panel. A panel is contained inside another container, or perhaps inside the web browser's window. Panels do not stand on their own. Applets are also panels.

Windows are subclasses of java.awt.Window. A window is a free-standing, native window. Window class is the superclass of two the classes: Frame and Dialog. Window appears as a rectangular area that is displayed on top of any existing applet, browser or, in fact anything that is already showing on the screen. By using the java.awt.Window class and its subclasses freestanding windows can be created. Stand-alone windows can also be used to build applications that use the AWT instead of the command-line or to provide additional space for applets.

A **frame** is an instance of a subclass of java.awt.Frame. It represents a normal, native window. Frames are very useful in more complex applications. Frames separate different functions or data into different windows. Everything you need to create and work with frames is contained in the java.awt.Frame class.

Constructors of the Frame are:

- *Frame( )*: creates a new Frame with no title.
- *Frame (String Title)*: creates a new Frame with the specified title.

Other Methods are:

- *boolean isResizable( )*: returns true whether the user can reseize the frame and returns false if it is not resizable.
- *void setResizable (boolean canResize)*: allow you to set whether the Frame is resizable or not.

A **dialog** is a subclass of java.awt.Dialog. It is a transitory window that exists merely to impart some information or get some input from the user. Dialogs are used for simple user input or for quick alerts to the user. They can be can be modal or models; may be resizable or not resizable; may be given a title or not given a title.

Constructors are the following:

- *Dialog (Frame Parent)*: Creates a new, untitled, modeless Dialog with the specified parent.
- *Dialog (Frame parent, boolean isModal):* Allows us to specify whether the Dialog is modal or not.
- *Dialog (Frame parent, String title)*: creates a modeless Dialog with the given title.
- *Dialog (Frame parent, String tile, boolean isModal)*: creates a Dialog with the specified properties

Other Methods are:

- *boolean isModal( )*
- *void setModal (boolean isModal)*
- *boolean isResizable( )*
- *void setResizable(boolean canResize)*

In order to add **menus** to a frame window, an instance of the MenuBar class can be created and can be added to the frame using the setMenuBar method.

5 ONLINE JAVA COURSE

5.1 The Relevance of Java and JavaScript in Distance Education Course Development

Java applications and applets are well-suited toward the modularization and distribution needed to be able to use resources on the Internet and reference other applications or application pieces to perform common tasks. Also, Java was the first to include inline sound and animation in a web page.

Java is simple and **secure**. It has various security features including the automatic garbage collection, the elimination of pointer arithmetic and the Java interpreter. Java has no operating system-unique extensions or variations. Since it is interperted, the compiler cycle is much shorter than the other languages. Java applets can be downloaded from the internet and played safely within a web browser. They will run on almost any operating system without requiring recompilation.

JavaScript can be useful for very small applications that run on the web client or server. It is a programming language that adds interactivity to our web pages.

5.2 Preparation of the Course

The online Java language course that was prepared for this thesis is on http://brahms.emu.edu.tr/tuncay. It is designed for a first course in computing using the

Java programming language and the principles of object oriented technology. Little programming experince is assumed. If students already have programming background, some of the language concepts and syntax can be covered more rapidly.

Some comparisons between Java and Pascal programming language are included in the course. People who do not know Pascal language could just skip those parts and this would not constitute any obstacles in learning the Java language using this course.

There exists many examples and answered exercises throughout the course. Quizzes including multiple choice, true/false and short answer questions are also available. At the end of each quiz, students are evaluated and correct answers are given.

5.2.1 Course Outline

The course is organised in five chapters, each containing a number of subheadings. In **chapter 1**, concept of object oriented programming and the benefits of the Java language is presented. How to write simple Java programs, applets and applications, and how to compile and execute these are explained.

In **chapter 2**, "Tokens, Operators and Expressions" describes the tokens of the language from which statements are constructed. This chapter explains how the operations are used to build expressions, and how expressions are evaluated.

In **chapter 3**, "Control Flow" discusses selection (if, if/else and switch) and repetition (while, for and do/while) structures. Also, the statements labelled break and continue are explained.

In **chapter 4**, "Classes and Objects" describes the classes, methods, fields and objects in Java. In this chapter, the String class, the random class, abstract classes, nested classes, garbage collection, packages, interfaces are discussed. This chapter also explains that extended classes and methods that can overload or override.

In **chapter 5,** "Graphical Programming" describes graphics context and graphics objects in Java. In this chapter creation of applets and applications with graphical user interfaces are explained. "Basic Graphical User Interface Components" Textual Widgets, Active Widgets, Containers, Windows, Frames, Dialogs, Menus, Label codes are discussed.

5.2.2 Geneneral Structure of the Course

The text is first prepared in Microsoft Word. (Then it is transferred to Microsoft FrontPage and the necessary corrections are made there since working in Microsoft FrontPage is easy.)

To make the pages attractive, and easily followed by the students, various colours are used. "Example box" have light blue color, "syntax boxes" have blue color, "pseudecodes" have orange color and answers of the exercises pages have green color and so on.

To provide easier passing between the pages, some buttons are arranged throughout the course. There exist three buttons at the top of each page of chapters.(which are shown in the Figs. 5.1-5.3) Back button links to the previous section, forward button links to the next section and up button links to the beginning of the current chapter.

Figure 5.1 Back button

Figure 5.2 Up button

Figure 5.3 Forward button

Fig. 5.4 shows a picture of the chapter 4.2 from the online notes. Here the back, up and forward buttons; white colored "syntax box" with grey border and blue colored "example box" can be seen. Fig. 5.6 is another example of the color arrangement, designed for taking the attention of readers.

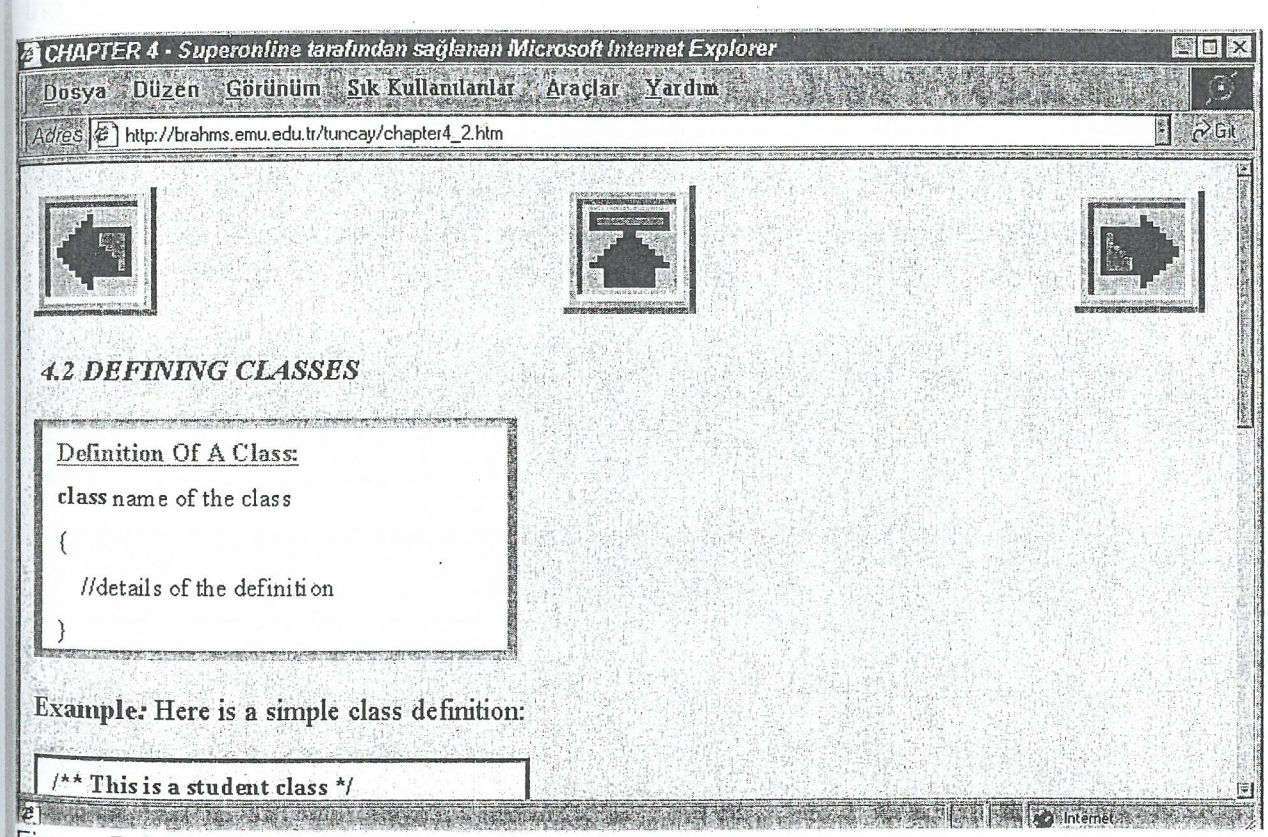

Figure 5.4 Chapter 4.2 from the online notes.

There are comparisons between the Pascal and Java languages throughout the text. When there are some similar statements in the **Pascal** language to the Java language, they are stated in a seperate text book with a label, shown in the Fig. 5.5

Figure 5.5 Pascal label

Thus, the students who know the Pascal language can make comparisons with the Java language which helps them to learn the language faster.

Fig. 5.6 shows chapter 3.5 from the online Java course; where the syntax of the "for" structure in Java and then the syntax of the "for" structure in Pascal is given.

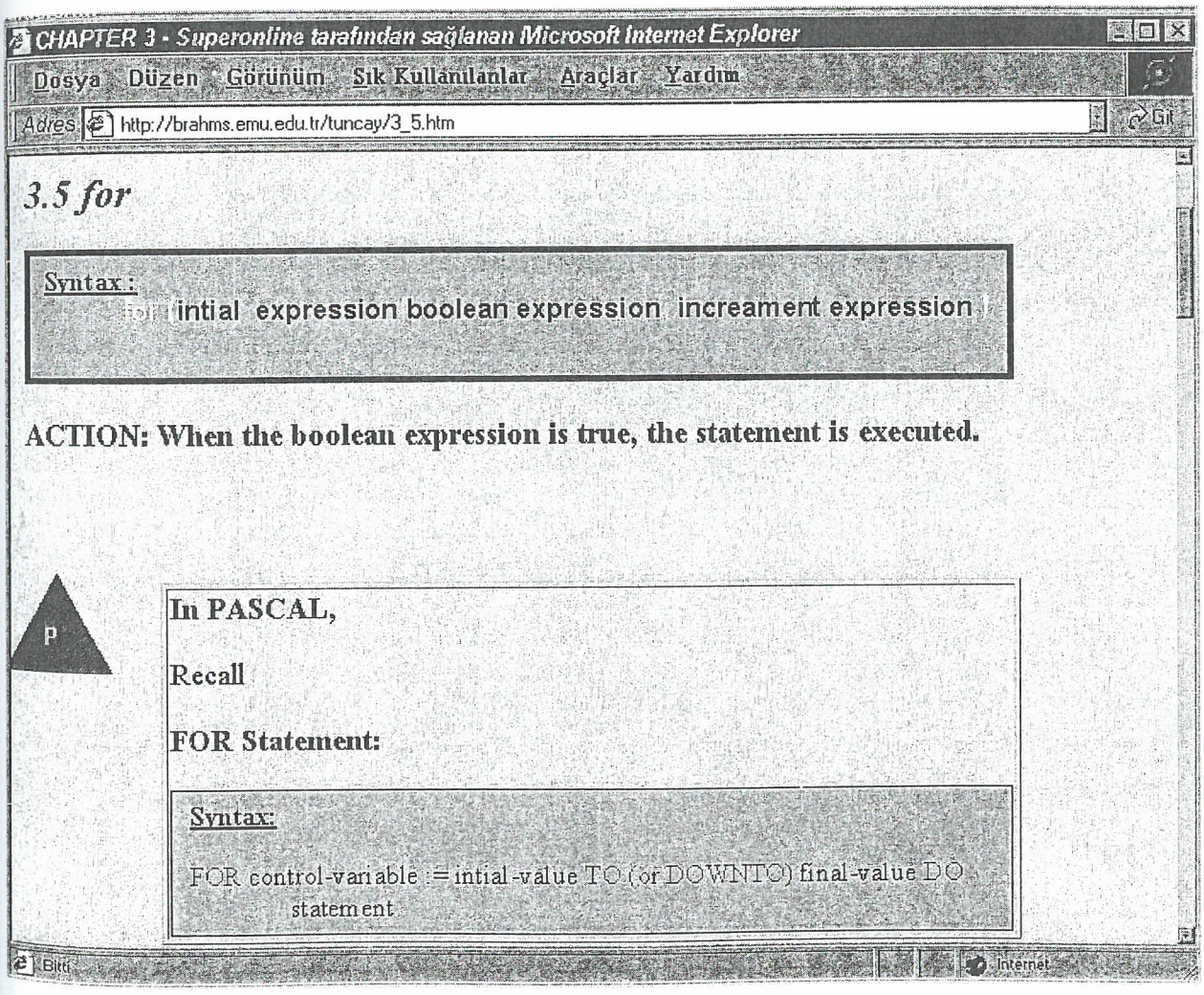

Figure 5.6 Comparison with Pascal language

Programming concepts are thaught by representative **examples**, which are carefully choosen and represented in an easy-to-follow style. Each example, includes the source code and a sample run.

There are also **summary sections** at the end of each chapter on the online Java course which reviews the important subjects that students should understand and remember. It also reinforces the key concepts that are stated in the chapter. Fig. 5.8 shows an example of the summary section of chapter 1. As shown in the Fig. 5.7, each summary section in the course contains only the **up button** which links to the chapter 1 outline. On chapter 1 outline, one can reach to all the chapter 1 subsections and from the back, forward and up buttons that this chapter 1 contains to the other chapters.

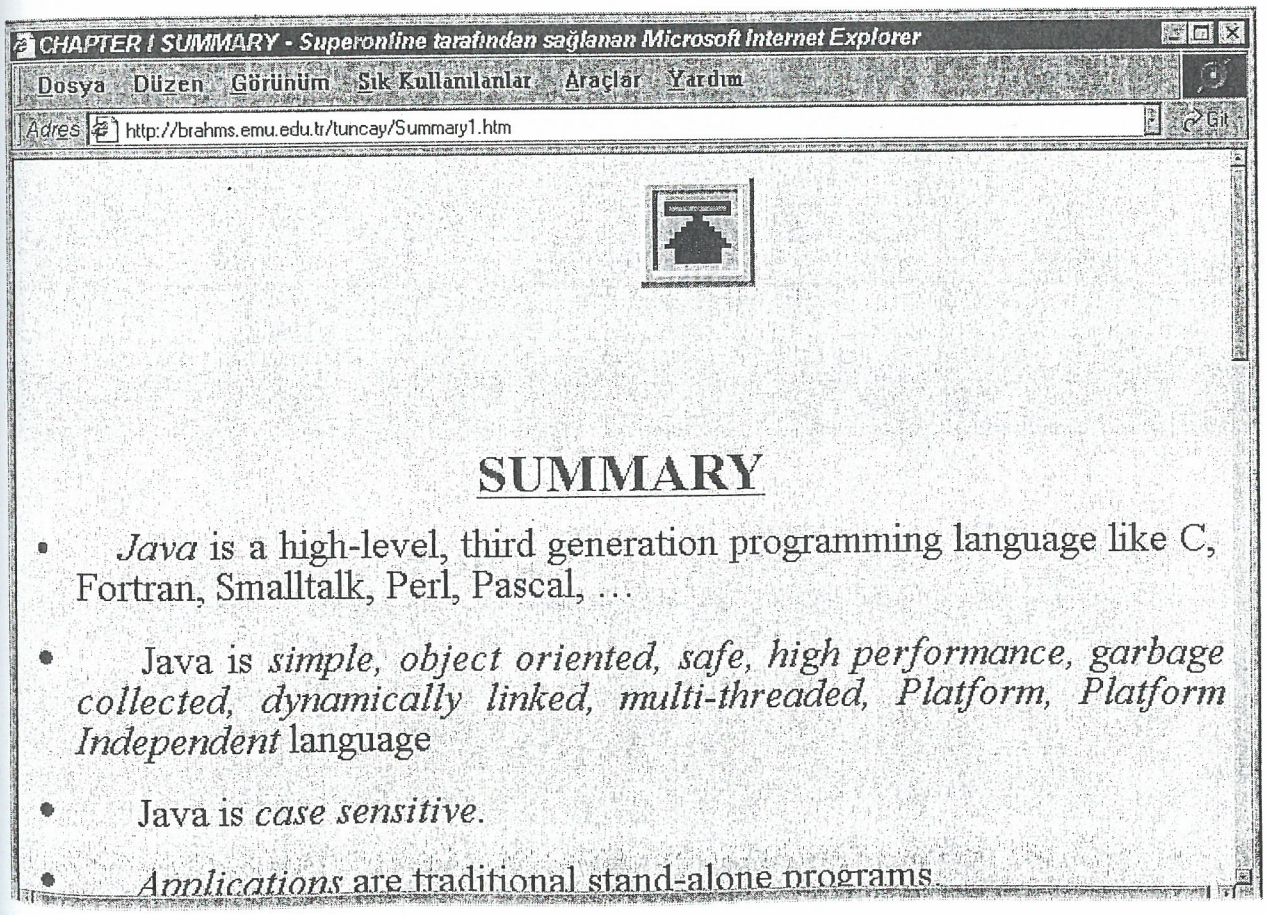

Figure 5.7 Summary of chapter 1

There are "chapter outlines" for each chapter which helps the students to easily follow the notes. "Chapter 1 outline" is shown in Fig. 5.8.

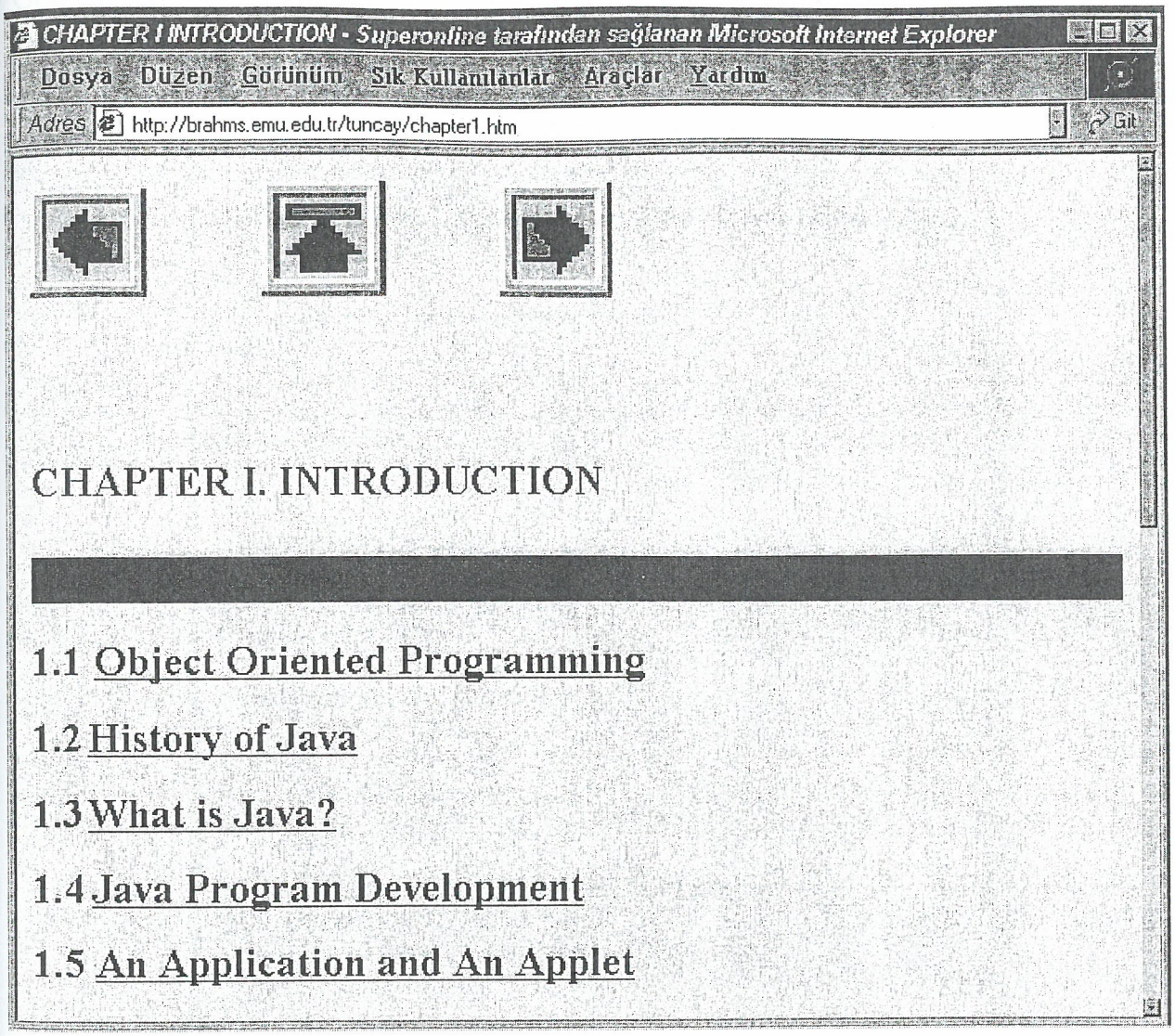

Figure 5.8 Outline of the chapter 1

Exercises , which comes after the summary section in the course, provide students with opportunities to apply the skills on their own.

At the end of each chapter, students are provided with multiple choice and short answer **quizzes** which are designed to evaluate their progress.

5.2.3 Using Graphical Programming Capabilities of Java

One can use Java Buttons, TextAreas, TextApplets,....on the online courses for purposes of displaying any information that is needed. NazButton applet is one of the buttons on the online notes which is illustrated in Fig. 5.9

```
import java.applet. *;
import java.awt. *;
public class NazButton extends Applet {
 Button
    c1 = new Button ("OUTPUT");
    Font font1 = new Font ("Helvetica", Font.BOLD, 14);
public void init (){
setLayout (null);
 add (c1);
  c1.reshape (12,24,60,26);
  }
public void paint (Graphics g){
c1.setFont (font1);
c1.setBackground (Color.yellow);
c1.setForeground( Color.black);
 g.drawRect (7,19,70,42);
 g.drawRect (8,20,70,40);
 g.setColor (Color.white);
 g.fillRect (9,21,67,38);
}
public boolean action (Event evt, Object arg){
 if (arg.equals("OUTPUT"  ) )
  getAppletContext( ). showStatus ("OUTPUT");
   return true;
 }
                                    }
```

Figure 5.9 NazButton applet

This applet creates a button c1 with label "OUTPUT"; 3 rectangles are drawn around this button and they are filled with white color. One can change the parameters here to get a different output. It is possible to obtain buttons with different colors (i.e. one can change the yellow and black colors, in Fig. 5.9 with one of the following colors: white, orange, gray, pink, cyan, magenta, lightGray, darkGray, red, green, blue). Also it is possible to obtain buttons with different sizes by changing the sizes of the rectangles drawn. The output of NazButton is shown in Fig. 5.10.

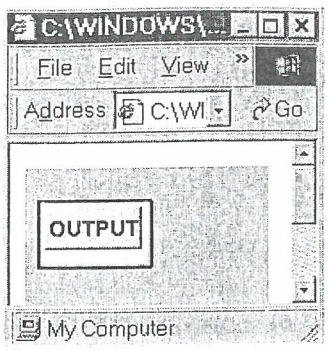

Figure 5.10 Output of the NazButton applet

Some kind of links in an applet can be implemented, so that a certain URL is loaded when the user clicks on a link. For this we need to have some mouse routines which handle the mouse click. There are lots of such buttons on the online Java course.

If you add the following statements to the NazButton applet, you can get a button that loads the web site http://brahms.emu.edu.tr/tuncay/javapage.htm, when clicked.

```
URL myURL= null;
...
try {
    myURL = new URL("http://brahms.emu.du.tr/tuncay/javapage.htm");
  } catch (MalformedURLException e){ }
```

To load a new document showDocument( ) method is used. If the URL specified isn't valid, an exception is thrown which you got to catch. If you look into the API documentation provided by Sun, you will see that there are different possibilities. One of them is showDocument(URL url). If only one window/frame is used this method can be used. This method loads the new page to the window which contains the applet. On the other hand, if you want to load it to a frame you must use showDocument(URL url, String target). This method takes the name of the target frame as a second argument.

On the online Java course two frames is used. The course notes appears on the frame at the top. As shown in the figure 5.11, frame at the bottom of the page contains seven Java buttons: Button with label HOME links to http://brahms.emu.edu.tr/tuncay and the other buttons with labels CONTENTS, CHAPTER1, CHAPTER2, CHAPTER3, CHAPTER4 and CHAPTER5 loads to the respective Java tutorial chapters to the frame at the top.

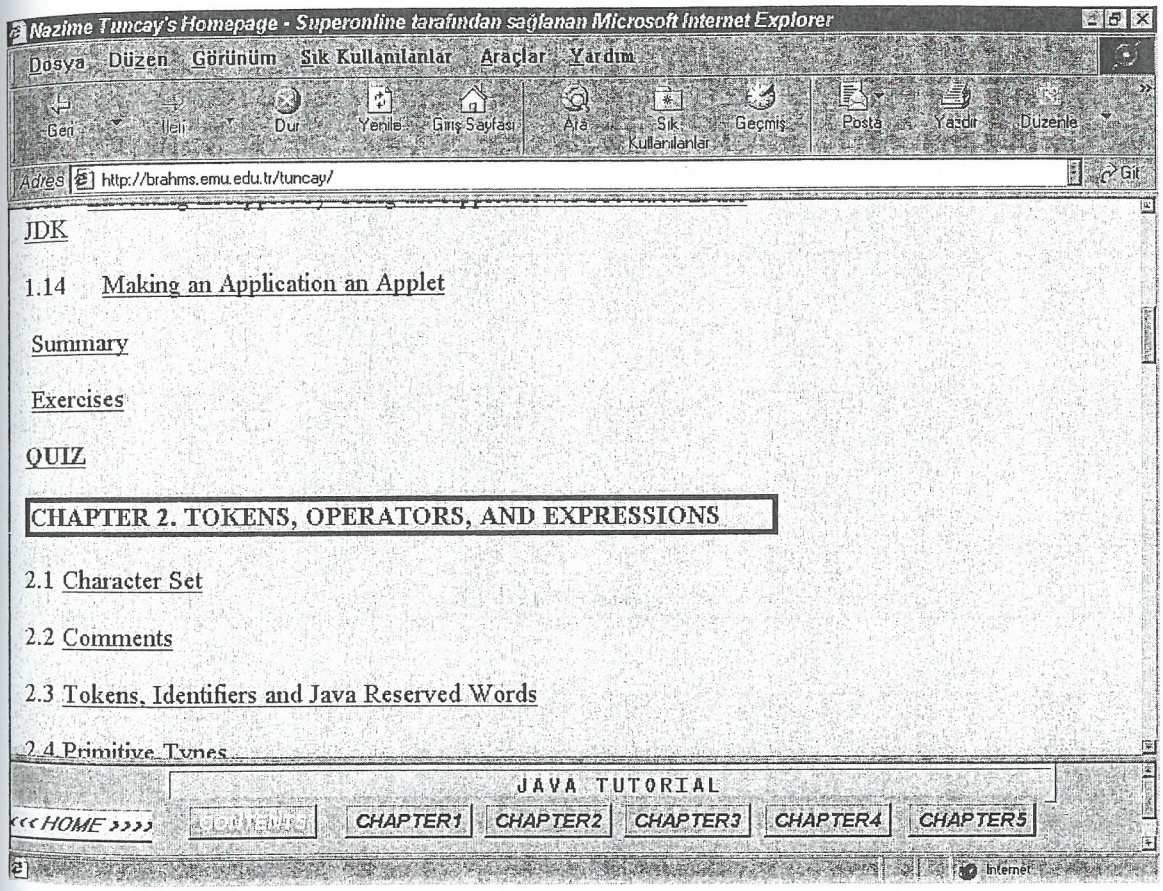

Figure 5.11 Frames on the online Java tutorial

Layout managers determines how components will be arranged when they are added to a container. So you can use FlowLayout, GridLayout, BorderLayout, CardLayout , GridbagLayout managers to arrange your buttons. On the online Java tutorial GridLayout is used to arrange buttons since it is a layout manager that arranges components into a grid of rows and columns. Figure 5.12 shows one example of the

buttons arranged with GridLayout. The statements creates a grid layout with 5 rows, 1 columns, a horizontal gap of 4 pixels, and a vertical gap of 4 pixels are the following:

```
GridLayout g1=new GridLayout(5,1,4,4);
 setLayout(g1);
```

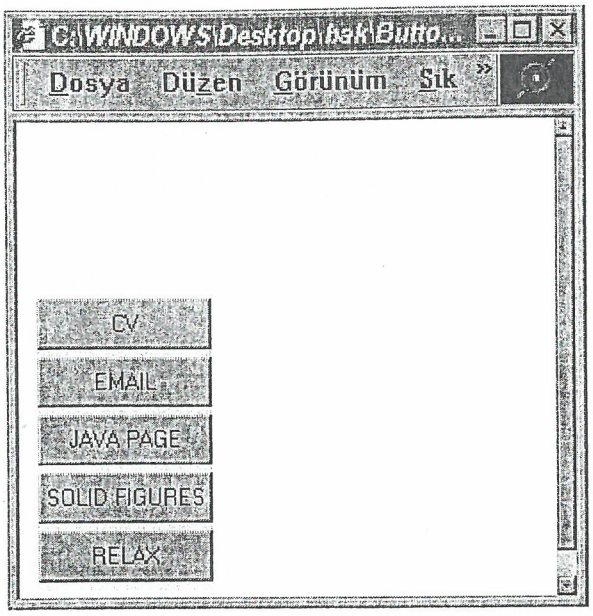

Figure 5.12 Buttons arranged with GridLayout

On the online Java course, there is also an applet called WhatIs (shown in Fig. 5.13).This program is written for displaying the definitions of Distance Education, Java and JavaScript. In this applet the components TextArea, Checkboxes and the CheckboxGroup is used. This program is written for displaying the definitions of Distance Education, Java and JavaScript. WhatIs class contains two textareas *t* and *m*. As the argument, *t* contains the following: *t1*, which is the string "WhatIs...", 1 row and 10 columns. *m* contains as the argument " ", 5 rows and 32 columns. *g* is defined as a CheckboxGroup. It contains three Checkboxes: *ans1* with label Distance Education, ans2 with label Java, ans3 with label JavaScript. For taking the choice from the student

and displaying the definitions of the terms being choosen, event action is used. According to the event an action is performed.

```
import java.awt. *;
import java.applet.*;
public class WhatIs extends Applet {
Font font1=new Font ("Helvetica", Font.BOLD, 14);
 Font font2=new Font ("Helvetica", Font.BOLD+Font.ITALIC, 12);
 String t1="What is...."
 TextArea t= new TextArea (t1, 1,10);
 TextArea m = new TextArea (" ", 5,32);
 CheckboxGroup g = new CheckboxGroup();
 Checkbox
  ans1 = new Checkbox("Distance Education ?",g, false),
  ans2 = new Checkbox("Java ?", g, false),
  ans3 = new Checkbox("JavaScript ?",g,false);
 public void init(){
  t.setFont(font1);
  ans1.setFont(font2);
  ans2.setFont(font2);
  ans3.setFont(font2);
  setBackground(Color.cyan);
  t.setEditable(false);
  m.setEditable(false);
  add(t);
  add(ans1);
  add(ans2);
  add(ans3);
  add(m);
 }
public boolean action (Event evt, Object arg){
 if (evt.target.equals(ans1)){
  m.appendText("\nDistance education is an education in\n which teacher and
student are\n");
  m.appendText(" separated by distance and time.\n");
} else
if (evt.target.equals(ans3)){
  m.appendText("\n JavaScript is Netscape's cross-platform,\n");
  m.appendText("  object-based scripting language for client\n  and server
applications.\n");
}else
if (evt.target.equals(ans2)){
   m.appendText("\nJava is a high-level, third generation\n");
   m.appendText("programming language\n");
} else
 return super.action(evt, arg);
  return true;
 }
}
```

Figure 5.13 WhatIs applet

This program can be used to display any information that is needed other than Distance Education, Java and JavaScript, by changing the "**bold words**" in Fig.5.13

An illustration of this process which "Distance Education" is choosen, is given in Fig. 5.14. The definition of the "Distance Education" appears in the textArea under the checkboxes.

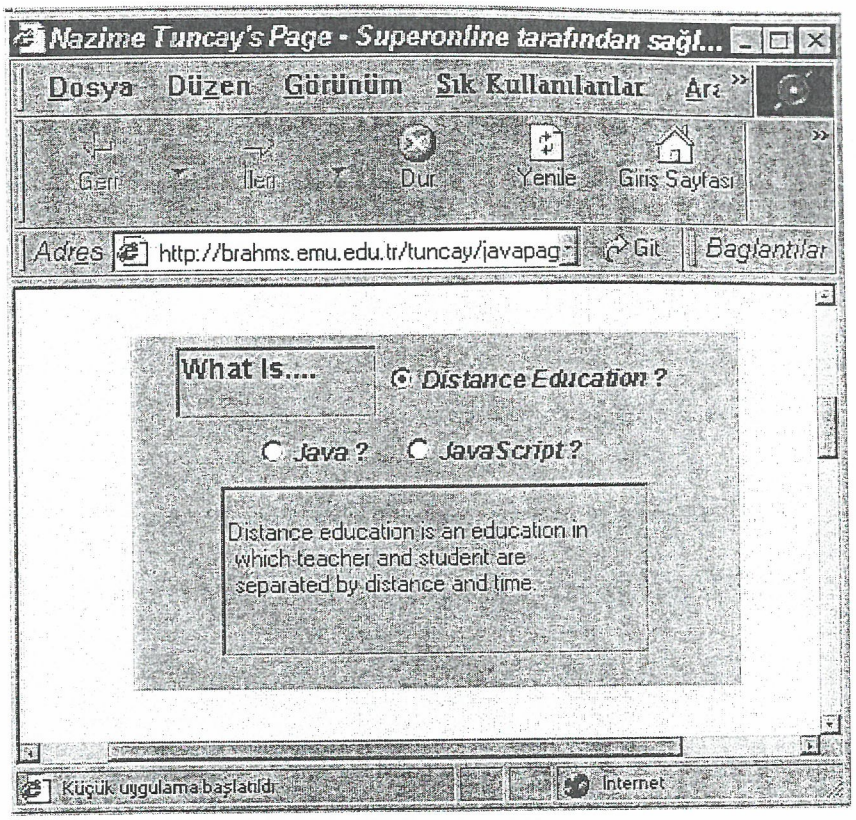

Figure 5.14 Output of the WhatIs applet

For the purpose of drawing attention of something or giving some piece of information, Frame object from java.awt class can be used.

In the note applet(See Fig. 5.15) used on the online Java course, a Frame appears informing that Chapter 5 will be available in three days.

It is possible to display any other message, other than "Chapter 5 will be available in three days!", only by changing the "**bold words**" in the applet , shown in Fig. 5.15

```
import java. applet.*;
import java.awt. *;
public class Note extends Applet
{
  Frame myFrame = new Frame("NOTE:");
  TextArea area1 = new TextArea(4,15);
public void init( )
{
 area1.appendText ("Chapter 5 will be available in three days!");
 MyFrame.setBackground (Color. white);
 MyFrame.add ("Center", area1);
 MyFrame.reshape (50,150,220,100);
 MyFrame.show( );
  }
}
```

Figure 5.15 Note applet

In the Note class *myFrame* is a Frame object with the label "NOTE: ". *area1* is a TextArea with 4 rows and 15 columns. In this area "Chapter 5 will be available in three days!" is displayed. The output of the note applet is shown in Fig. 5.16

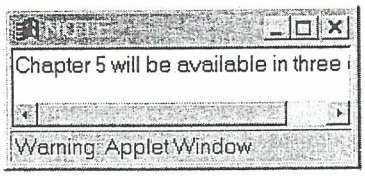

Figure 5.16 Output of the Note applet

5.2.4 Transfering Text and Applets to the Microsoft FrontPage

Microsoft FrontPage is a widely used software product, which can be used to manipulate and correct web page contents easily. The text which is prepared in Microsoft Word, is transfered to the Microsoft Frontpage to use its features like viewing *html format*(See Fig.5.19) and viewing the way the page appears on the browser(*preview*). Microsoft Frontpage also helps us to use dynamic html effects. These are *Javascript* programs for the purposes of surrounding a heading with a colored

rectangle when the mouse points that heading or changing the color/ font of a word, which has a hyperlink, when clicked or pointed by a mouse.

On the online Java course, the chapter headings are surrounded with black rectangles. An example is presented in Fig. 5.17

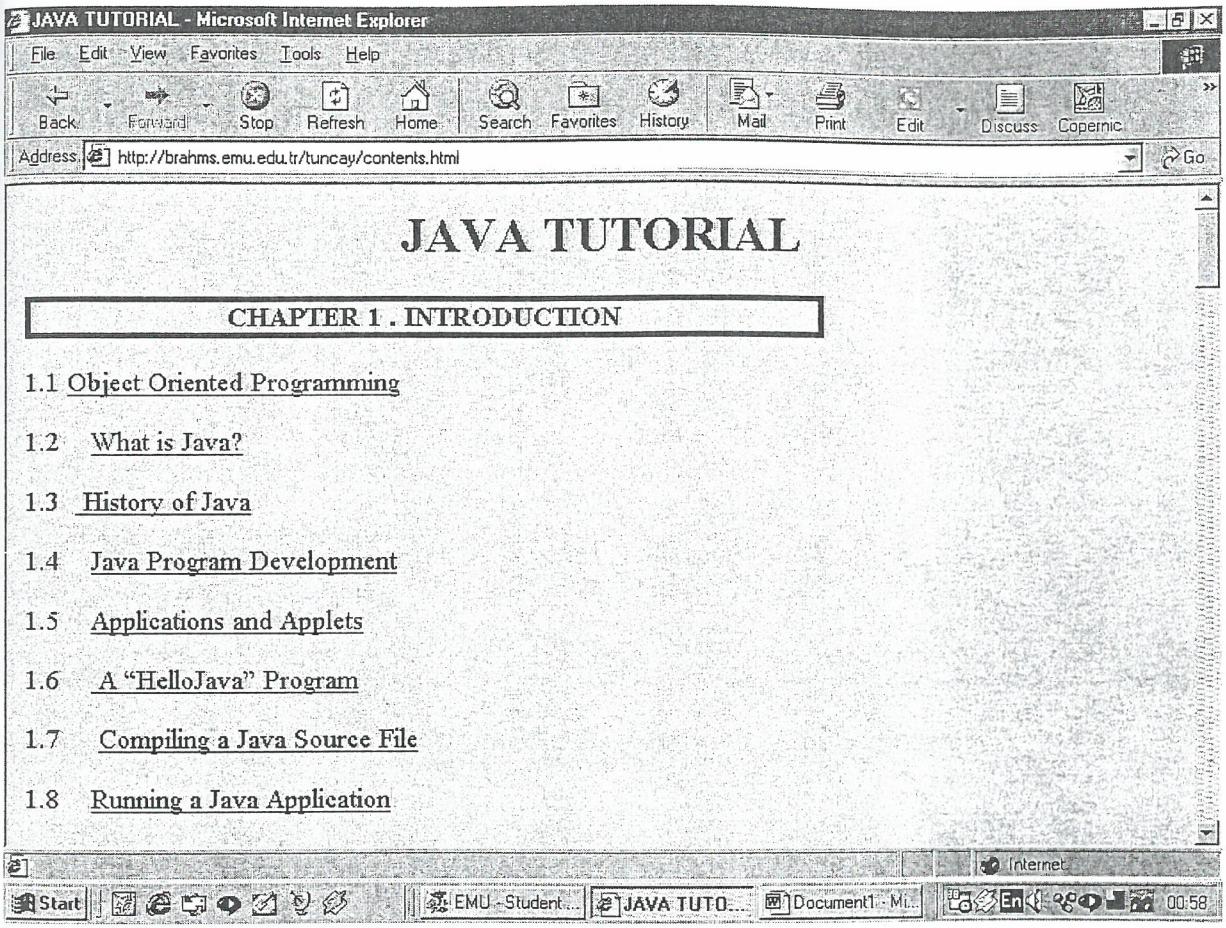

Figure 5.17 Surrounding the chapter headings with black rectangles

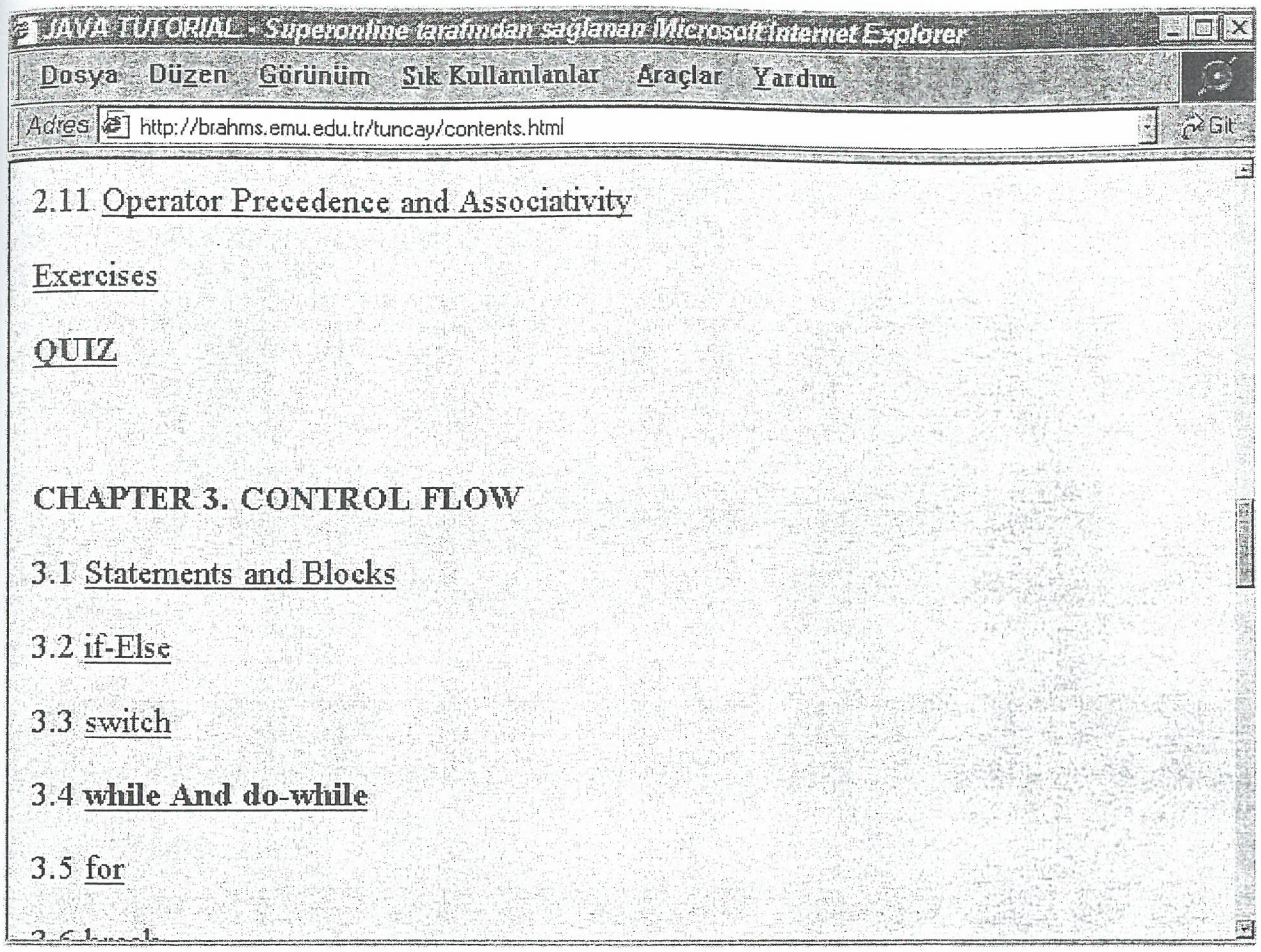

Figure 5.18 Changing font of the heading when pointed by mouse

If one points with the mouse to the chapter subheadings, their font will change.

To include java applets to the webpages html language is used which appears on the Microsoft Frontpage as shown in the Fig. 5.19. It is possible to arrange an applet anywhere in a web page by including the following codes in the html codes of the Microsoft FrontPage:

```
<applet code="NameOfTheApplet.class" width=250 height=250>
</applet>
```

Here one can change the NameOfTheApplet, width and height. Writing necessary html codes on Microsoft Frontpage helps us to arrange the Java applets inside the text easily. Fig. 5.21 is an ilustration of this process with the Welcome applet.

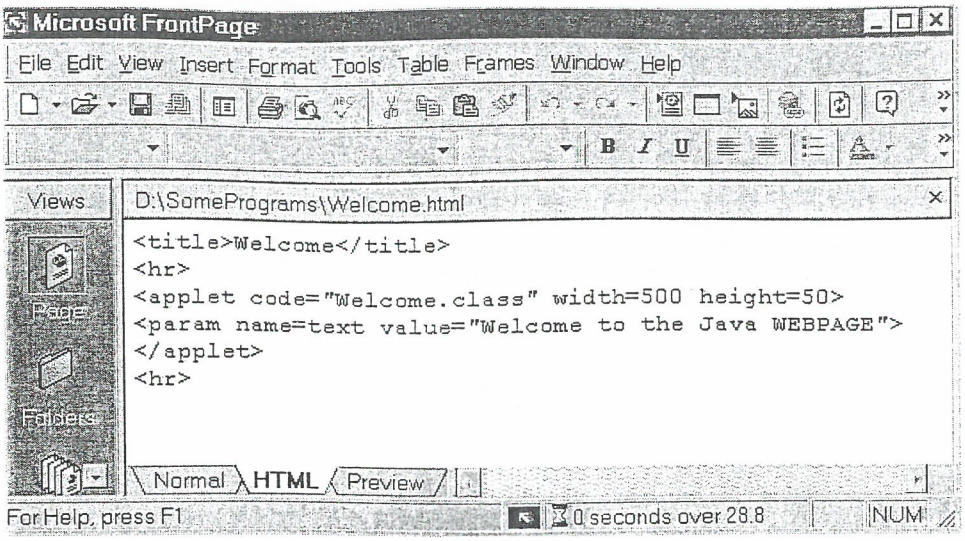

Figure 5.19 HTML codes of the Welcome.html on Microsoft Frontpage

In the Normal view of Microsoft Frontpage an icon appears. By moving this icon, one can arrange the place of an applet in a webpage easily. Fig. 5.20 shows the normal view of Welcome.html

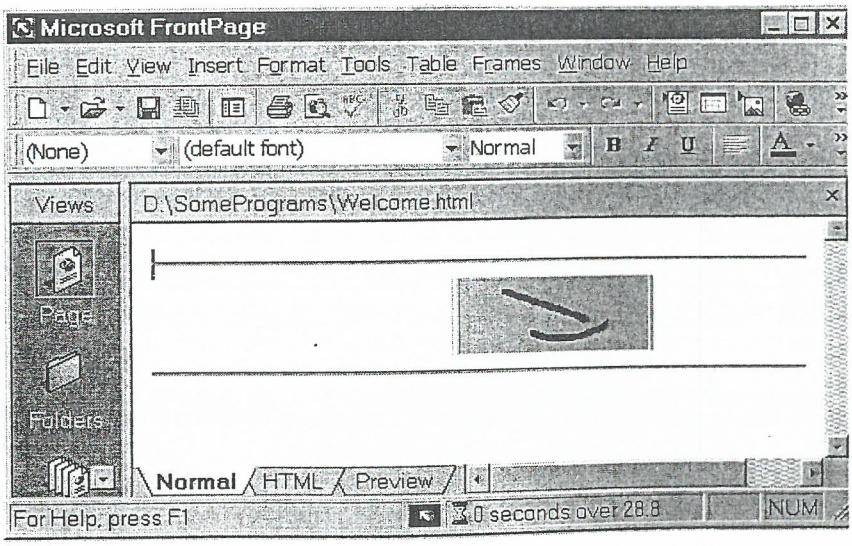

Figure 5.20 Normal view of the Welcome applet on the Microsoft Frontpage

Also, it is possible to see the way that the output of a Java applet appears on a web browser by *preview* in Microsoft Frontpage. *Preview* of Welcome applet is shown in Fig. 5.21

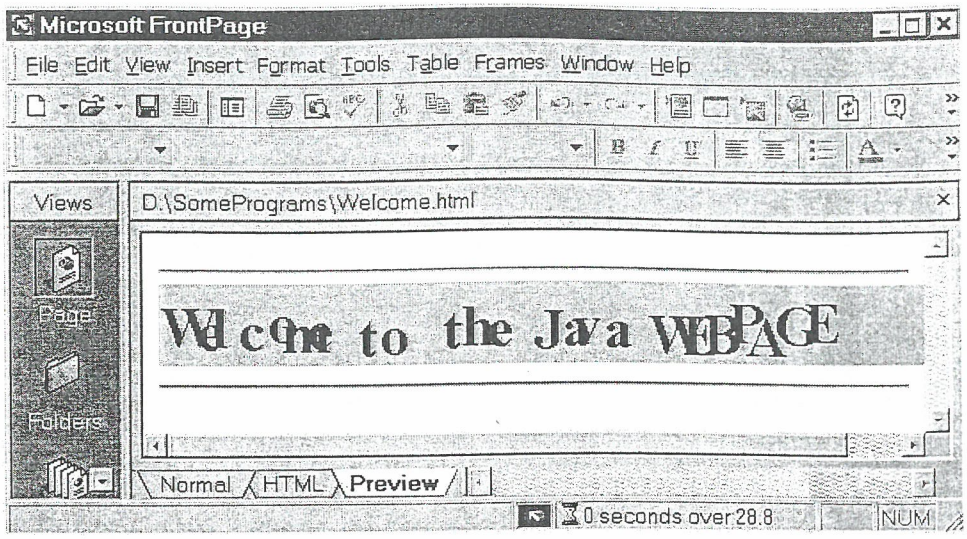

Figure 5.21 Preview of the Welcome.html

5.2.5 Including Images of the Outputs of Java Programs

The outputs of the Java application and applet examples are included on the online course. This images are obtained by using *Paintshop_Pro 7.0* screen capture and image facilities. Two examples of the images of the Flowers application and the Definition applet outputs, obtained by using PaintShop_Pro 7.0, can be seen in the Figs. 5.22-5.23

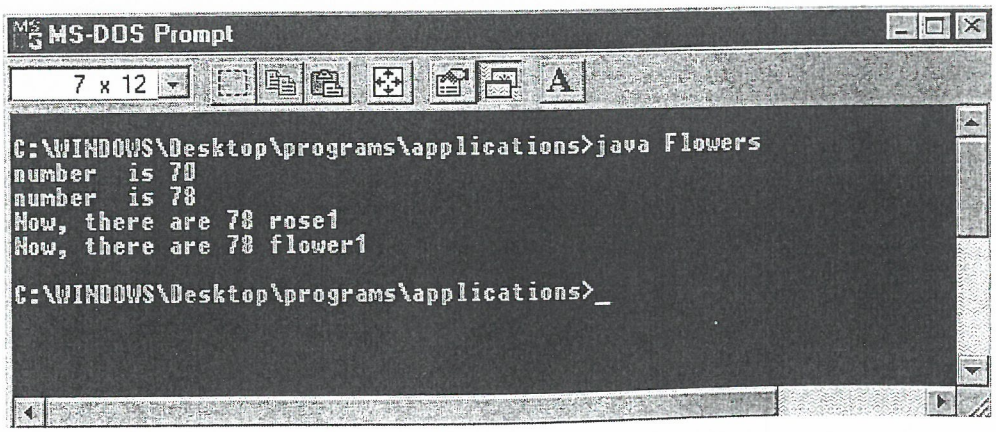

Figure 5.22 Image of the output of the flowers application, which is taken using Paintshop Pro 7.0.

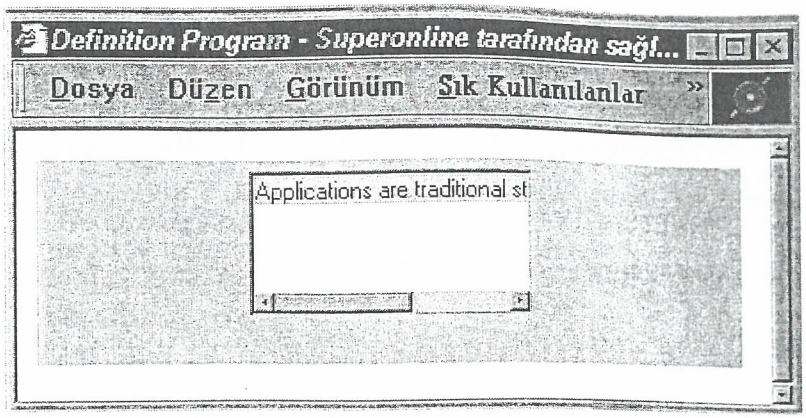

Figure 5.23 Image of the output of definition applet which is taken using Paintshop Pro 7.0

5.2.6 The Preparation of Exercises

At the end of each chapter there are a number of exercises. These exercises are first prepared on Microsoft Word and then transfered to the Microsoft Frontpage. At the end of the exercises there exists a Java button to view the answers. Figs. 5.24-5.25 shows the first page and last page of exercises for the chapter 1.

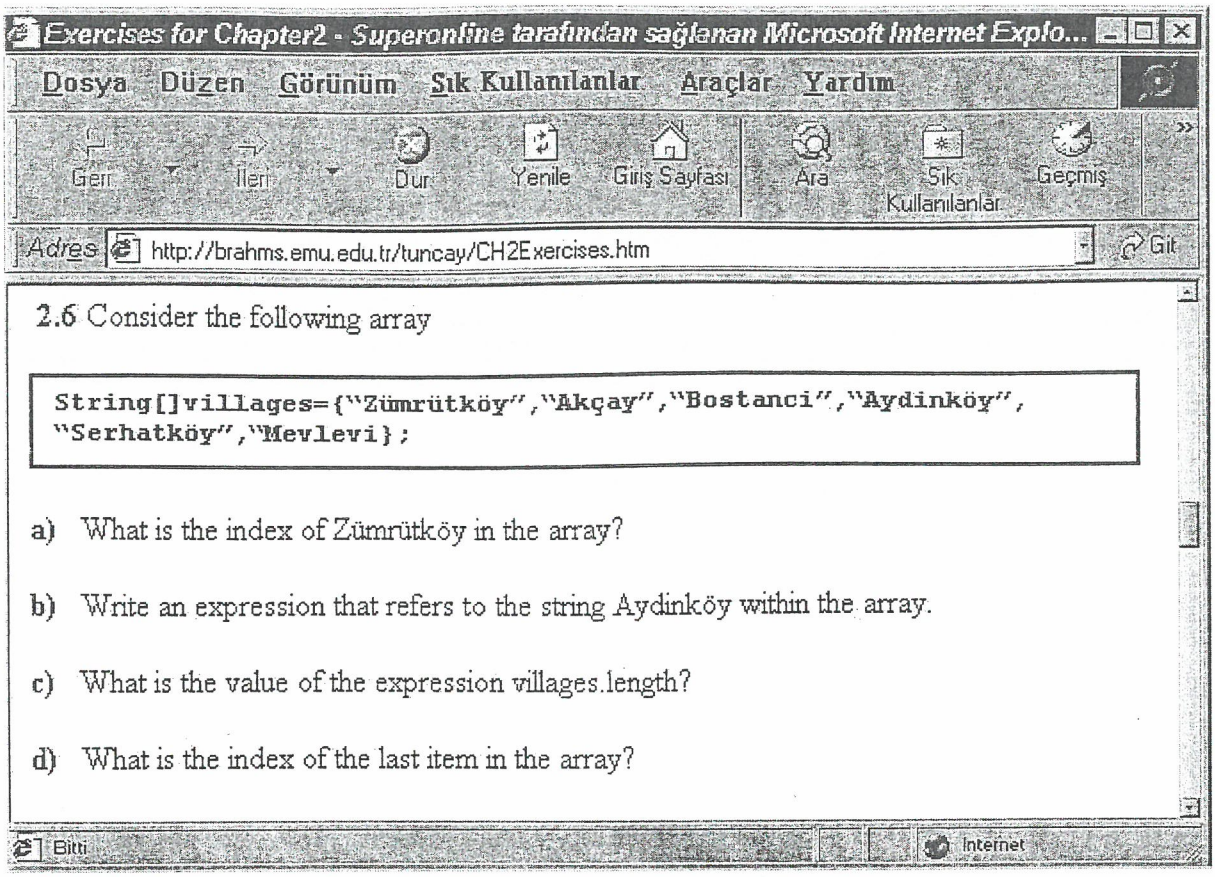

Figure 5.24 Exercises for chapter 2

There are three buttons at the end of the exercises: Back, forward and answer. Back button links to the current chapter notes and forward button links to the next chapter notes. If answers button is clicked, another page that contains the answers appears.

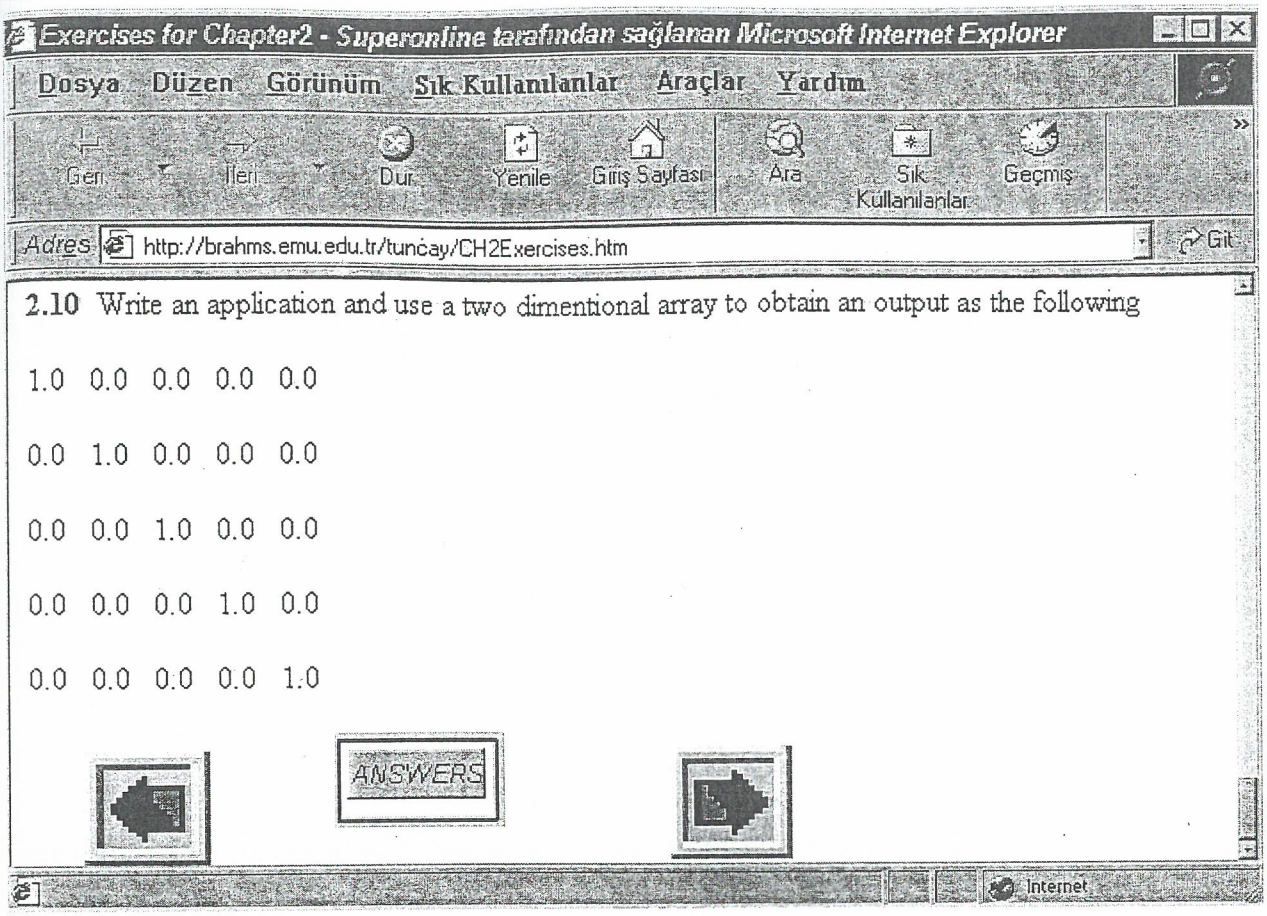

Figure 5.25 End of the exercises for chapter 2

The first page of the answers for chapter 2 is shown in Fig. 5.25

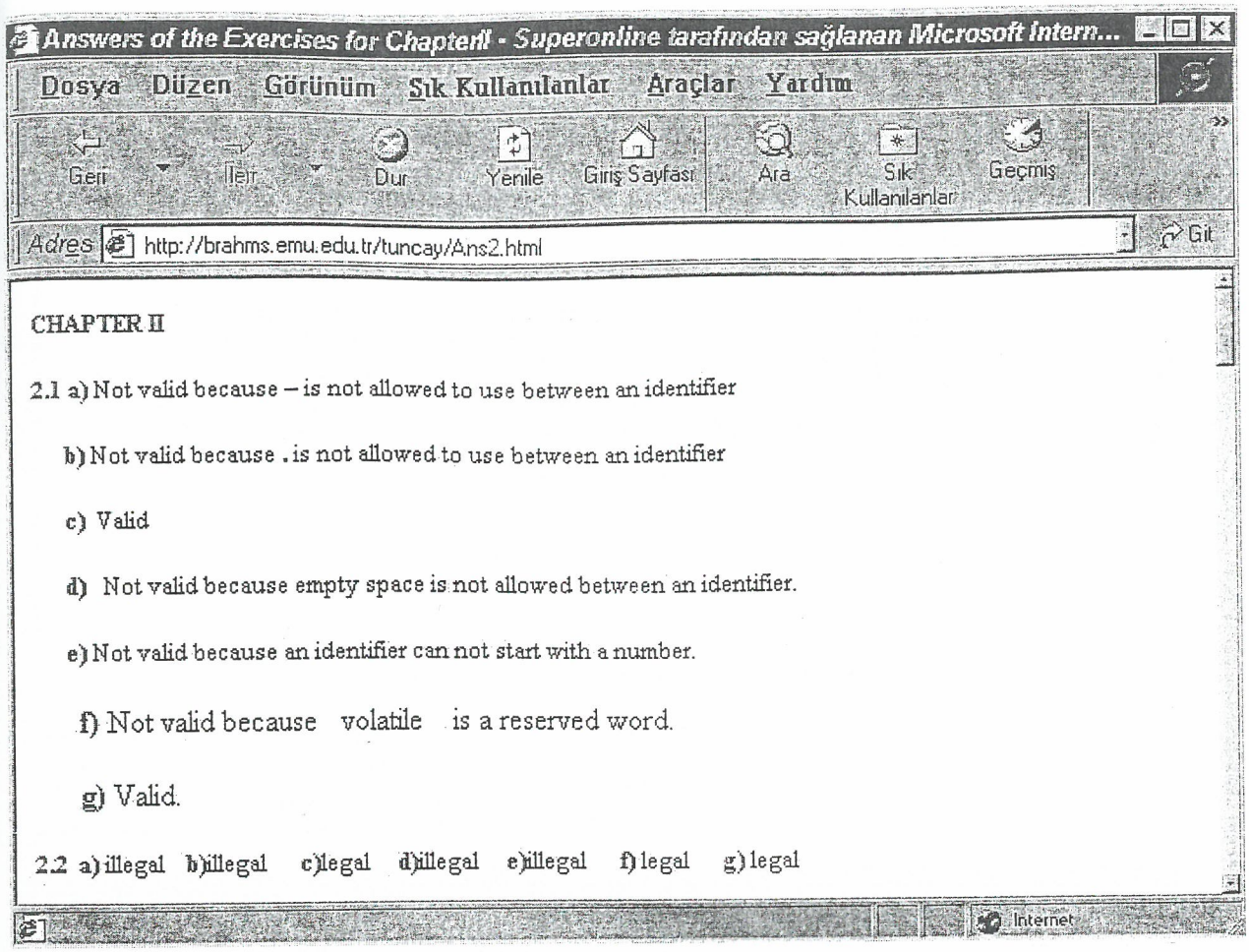

Figure 5.26 Answers of the exercises for chapter 2

There exists three buttons at the end of the answers. These are Back, forward and answer buttons. Back button links to the current chapter notes and forward button links to the next chapter notes. If Questions button is clicked, another page that contains the quetions appears.

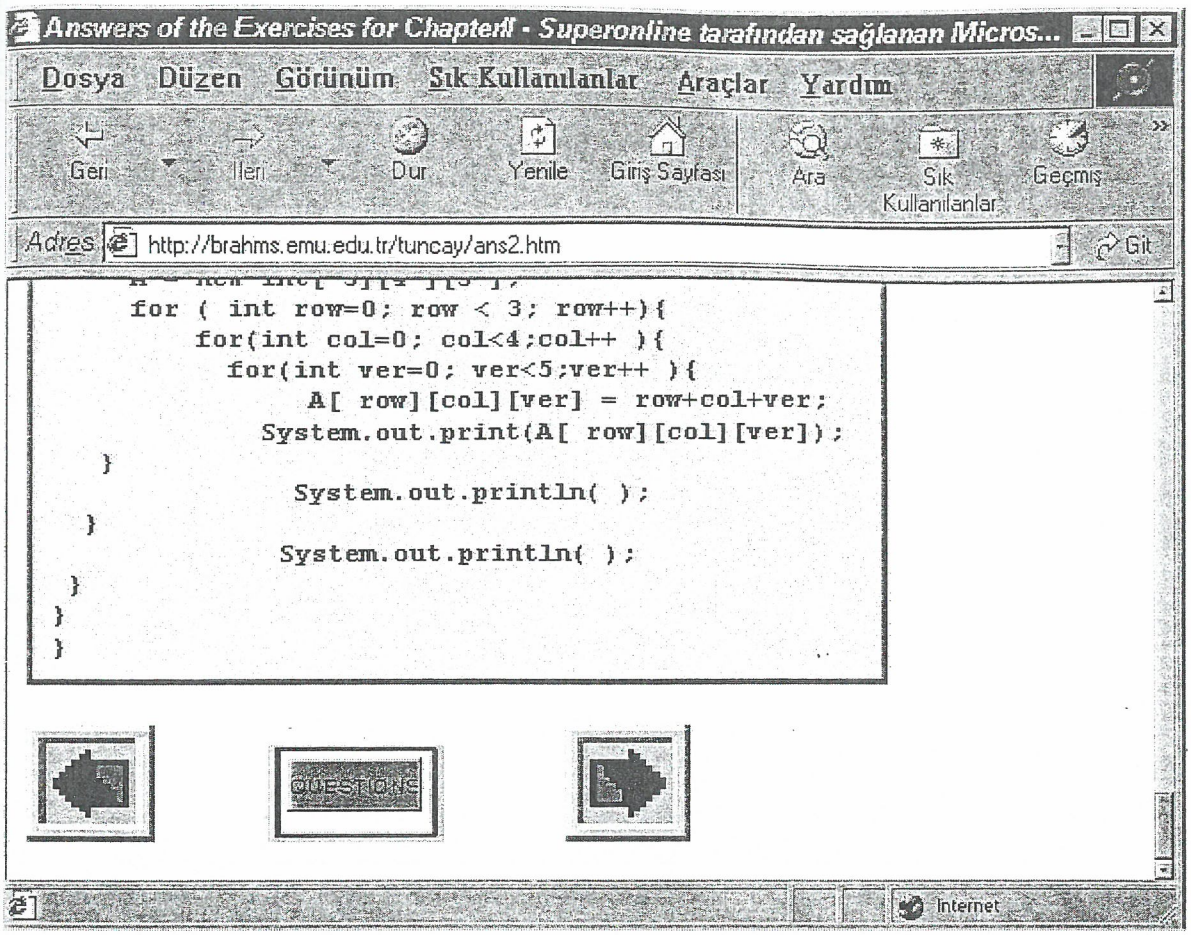

Figure 5.27 End of the answers of the exercises for chapter 2

5.2.7 The Preparation and Evaluation of Quizzes

Hot Patotoes v4.0 by Half-Baked Software is a suite of six programs for creating interactive educational exercises for the World Wide Web. It contains the *JBC* program which creates multiple-choice or true-false quizzes and *JQuiz* program which creates text-entry or short-answer quizzes along with many other programs. These programs use JavaScript and HTML for their functionality. One, does not need to know JavaScript or HTML in order to use these programs. By entering data (questions, answers etc.) and pressing a button, the programs will create web pages automatically.

When a Hot-Potatoes program creates Web pages, it does so by combining 3 resources: The **data** you entered, the **configuration** information and a set of **"source files"**, or templates, containing the page structure. The configuration information is a

collection of pieces of text, including instructions for doing the exercise, button captions, and link URLs, which are unlikely to change much from one exercise to another. One can get access to all the configuration information by choosing "**Configure Output**" from the **Options menu.**

JBC and JQuiz programs are used for creating exercises at the end of each chapter on the online course. Fig. 5.28 shows an example of JBC program which is used for preparing a multiple choice exam. Here the questions, the choices and the sentences for feedback are written.

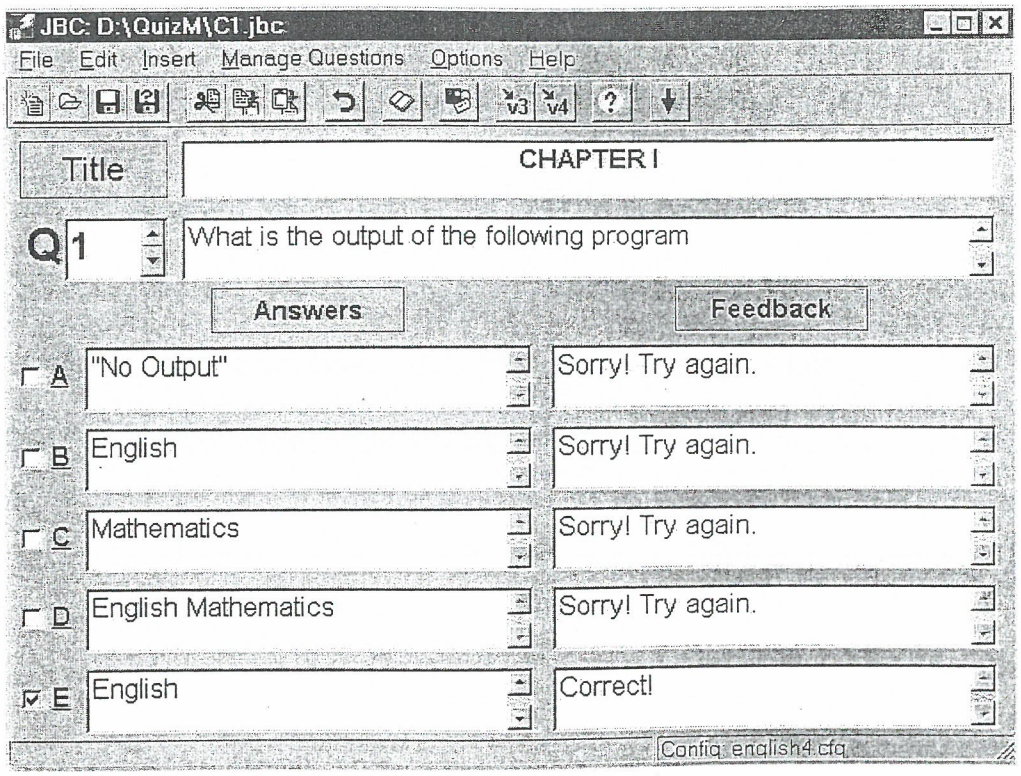

Figure 5.28 Preperation of a quiz with Hot_Patatoes.

After all of the questions are prepared for a quiz, they are exported to HTML as shown in Fig. 5.29.

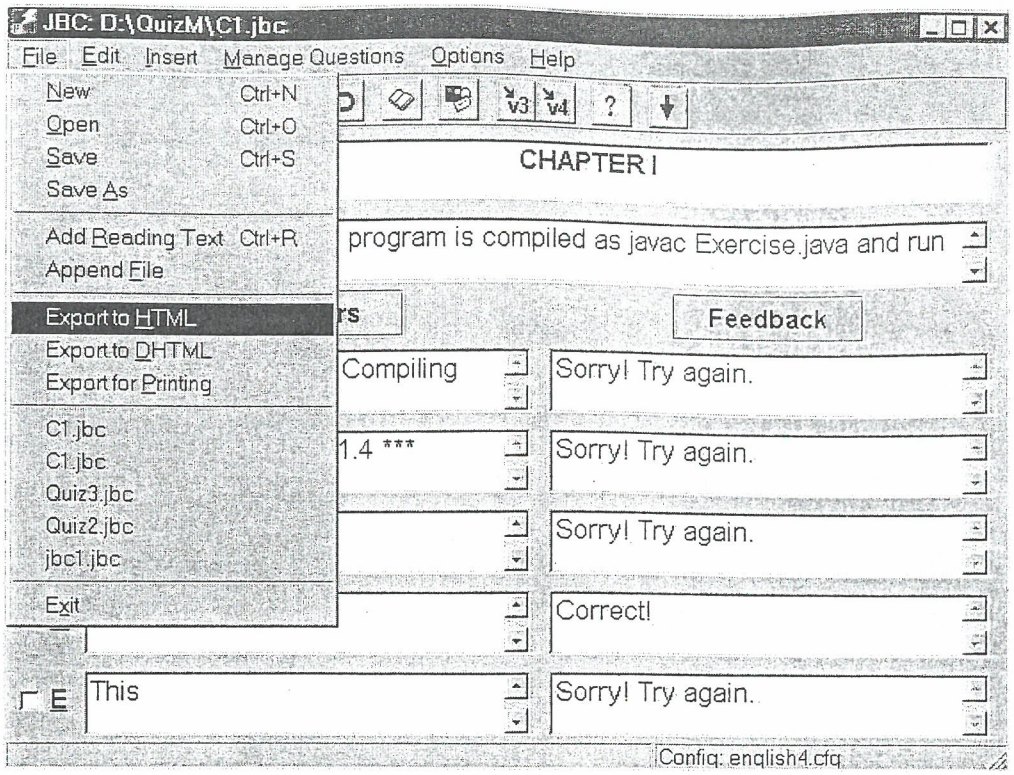

Figure 5.29 Exporting the questions prepared to HTML

If the questions are "Exported to DHTML", the questions appears on the page one at a time. Also the quizzes "Exported to DHTML" can have the proporties that the order of the questions changes each time that the page loads and r the order of the correct answer between the other choices changes.

After exporting to HTML, a menu appears as shown in Fig. 5.30 asking where to save the files that JBC has created.

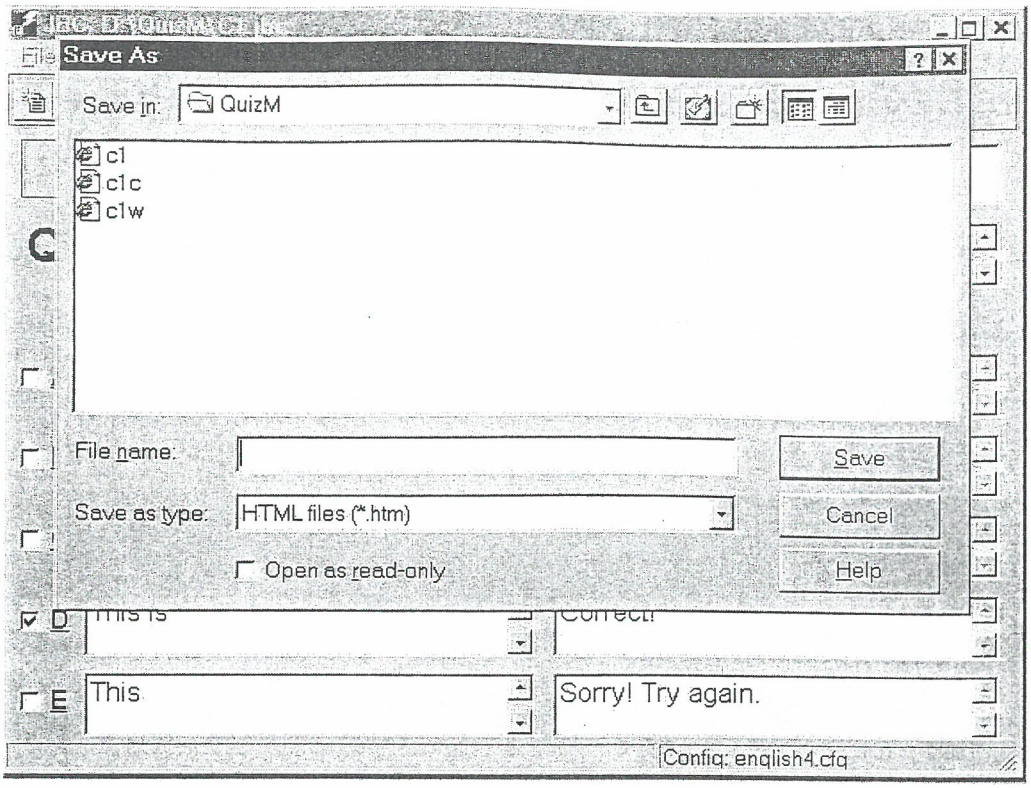

Figure 5.30 Saving the files created at the end of the process of exporting to HTML.

The quiz is ready and a message telling that it is possible to view the quiz, as shown in the Fig. 5.31, appears.

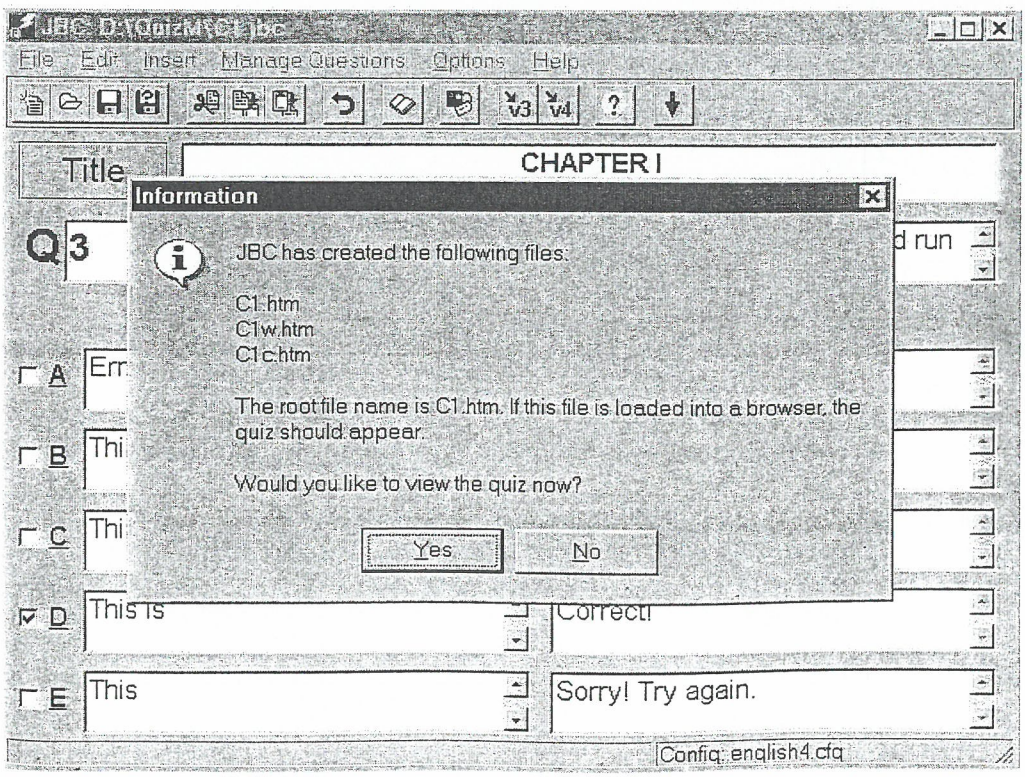

Figure 5.31 Message that appears after saving the files that JBC created.

Fig. 5.32 is the third question of the multiple-choice quiz at the end of the chapter1. There are three buttons at the top with the labels: BACK, CHAPTER1 and NEXT. When the button with the label BACK is clicked, the web page, which is visited just before coming to the quiz, appears. When the button with the label CHAPTER1 is clicked, notes for chapter 1 appears. When the button with the label NEXT is clicked, short-answer quiz questions appears.

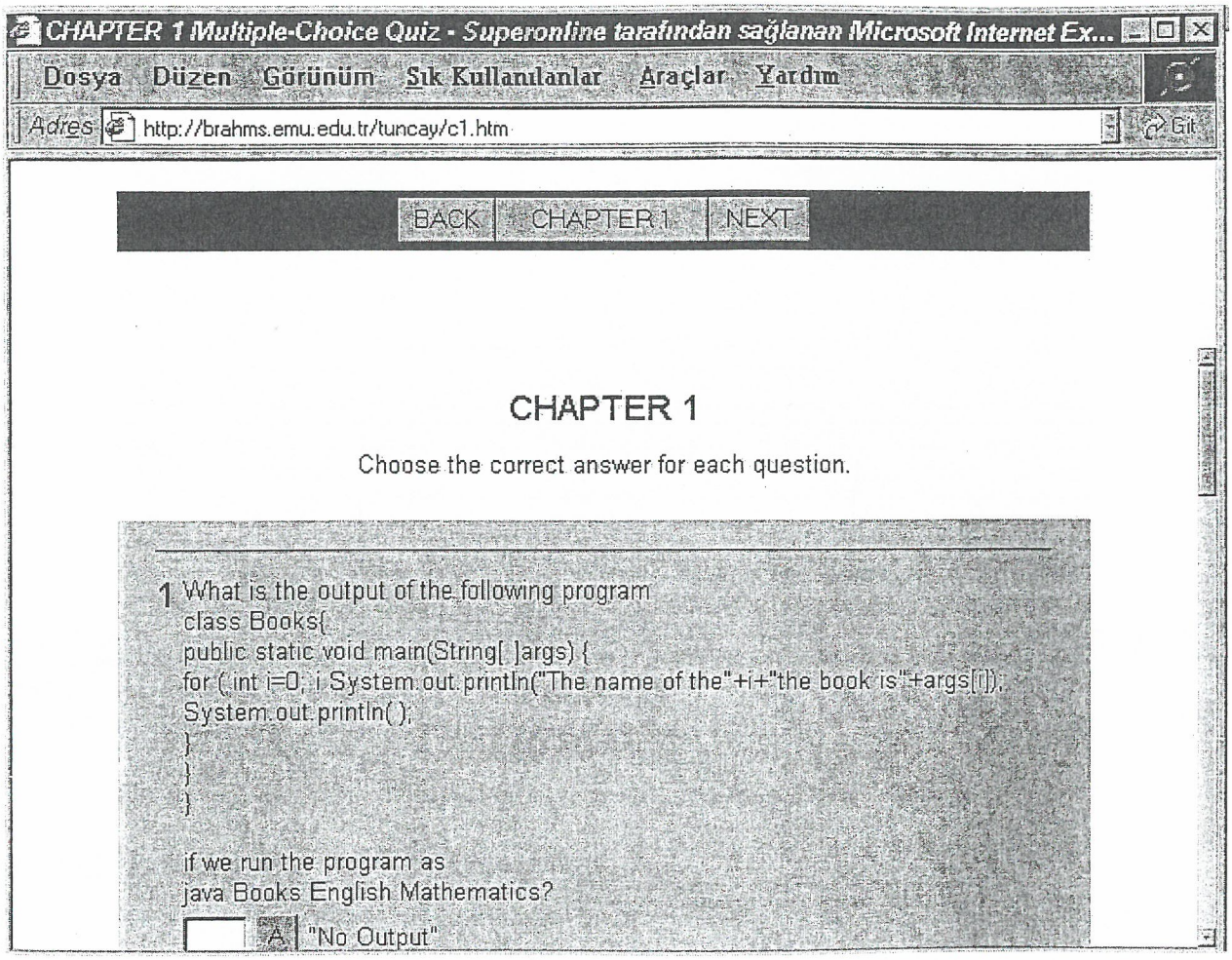

Figure 5.32 First question of the multiple-choice quiz from the chapter1

There are five buttons of choices in each multiple-choice question in the course. These have labels: A, B, C, D and E. When a wrong answer is clicked ' X ' appears in the area in front of the buttons. And a message like "Sorry! Try again." appears in the frame at the top.

Fig. 5.33 shows the case that a wrong answer is given to the second question of the multiple-choice quiz in the chapter 3.

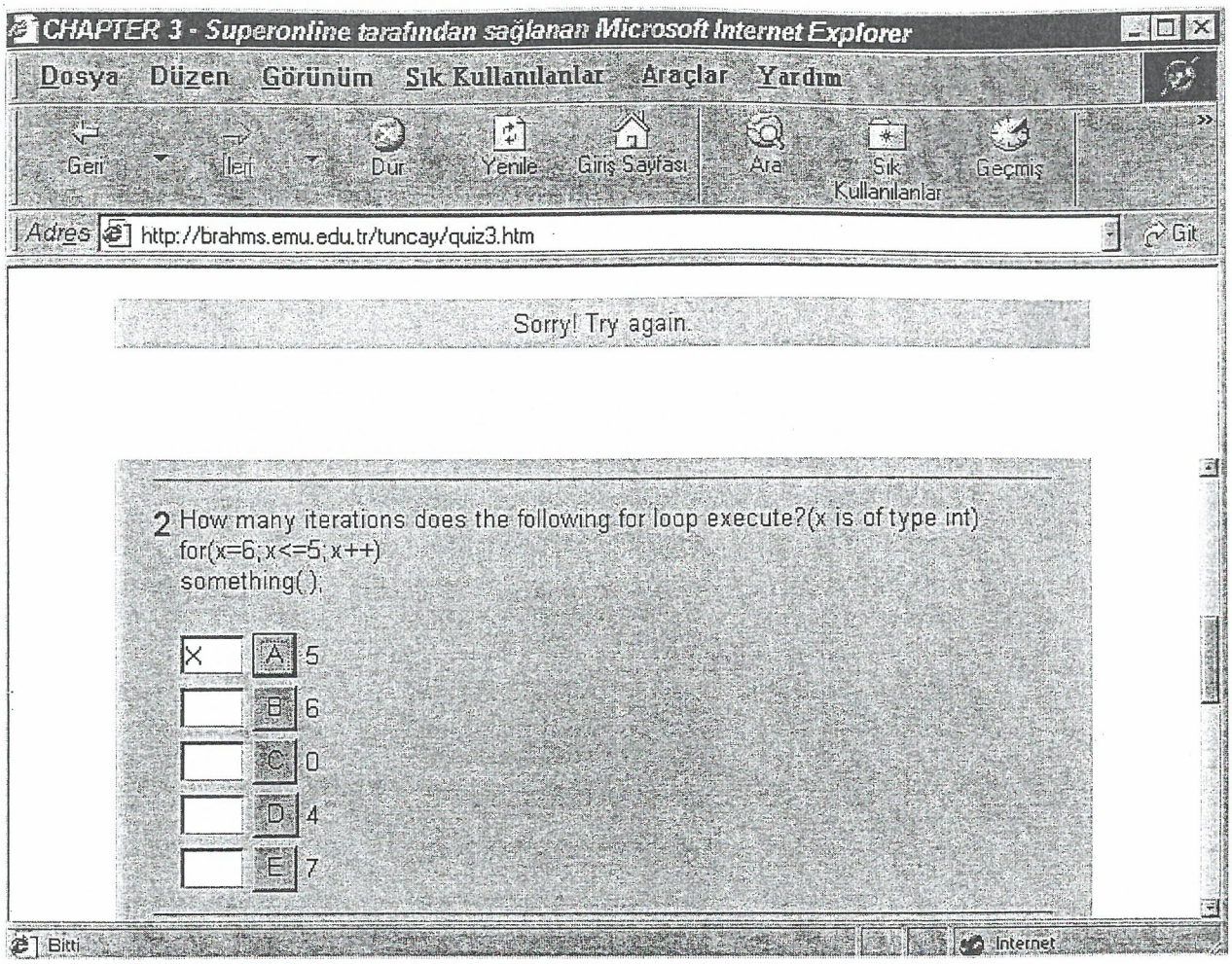

Figure 5.33 Response given to a wrong answer

On the other hand when one of the choices is correct ":-)" appears in the area in front of the buttons.

Fig. 5.34 shows the response given to a correct answer, in chapter1. Here when it is found that the correct answer to the question is "E", "Correct!" message and " Your score is %100" appears in the frame at the top. This means that the answer is correct and it is found at first guess.

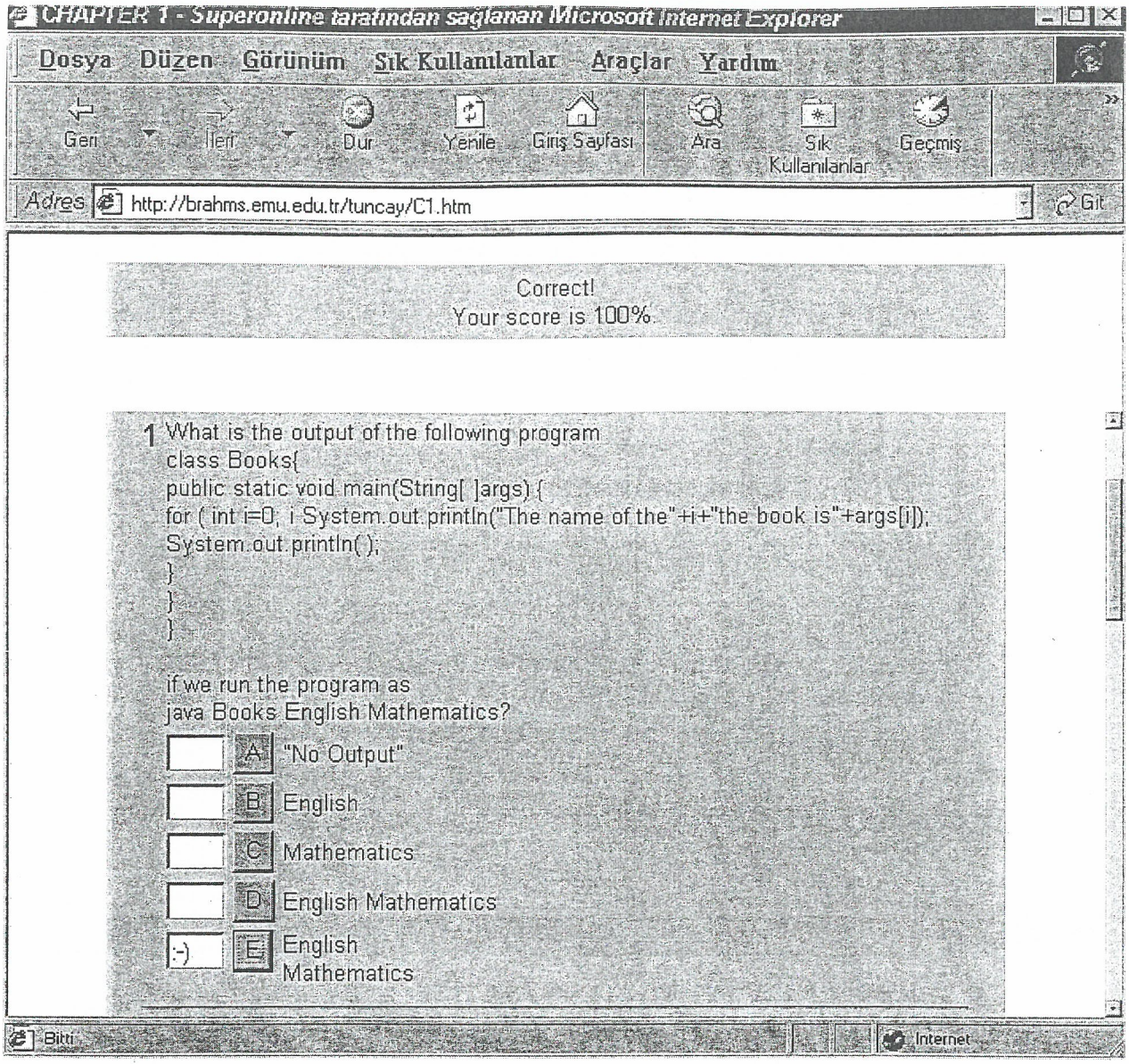

Figure 5.34 Response given to a correct answer

Sometimes, if there is a rule explaining why the answer cannot be correct, a reasoning explaining this appears in the frame at the top of the page.

The following figure, Fig. 5.35, shows the first question in the second chapter. In this question, the choice "**A**" is chosen which is wrong. So, a reasoning why it is wrong is given.

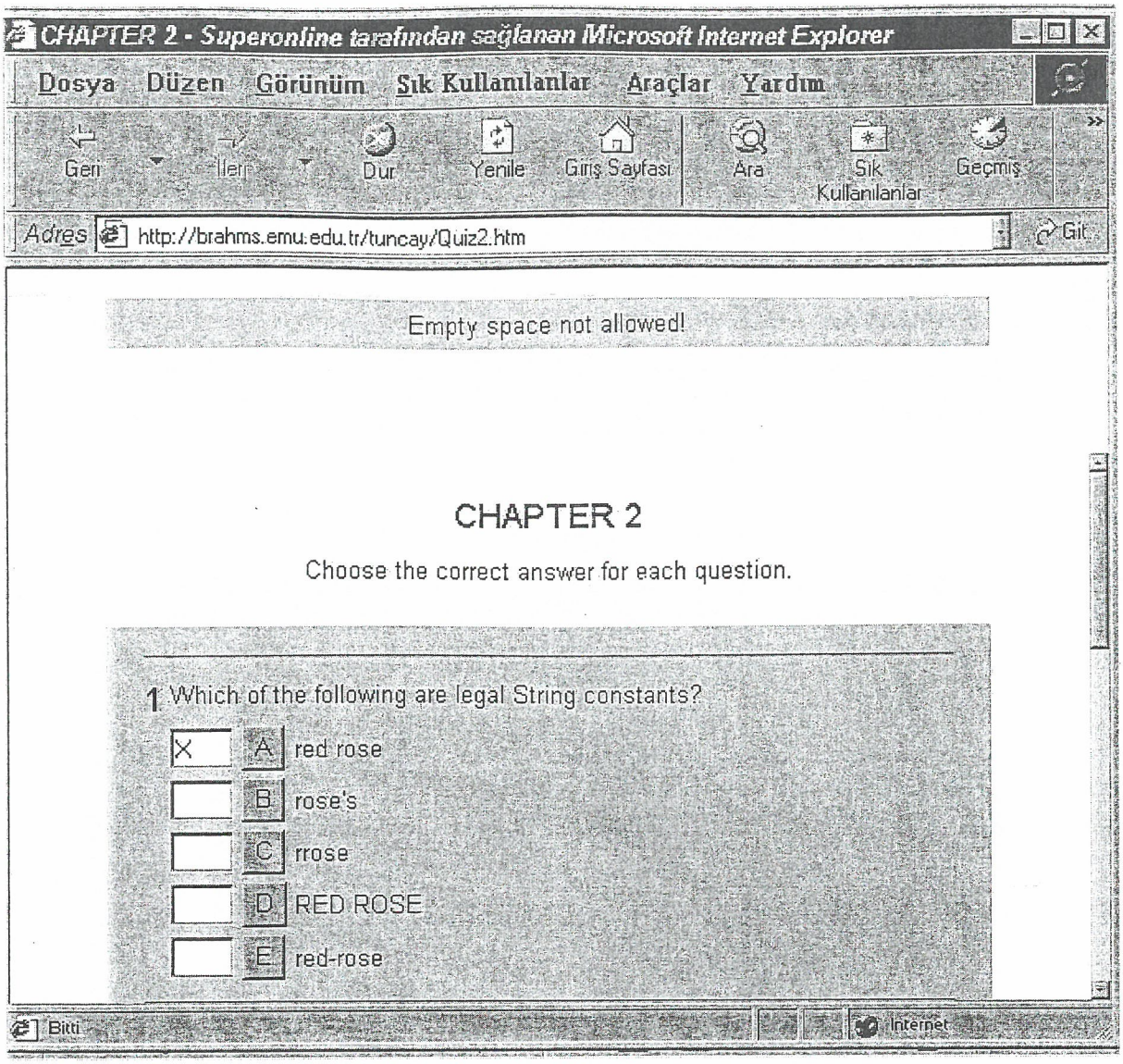

Figure 5.35 Feedback given to a wrong answer

If the user gets the correct answer after second trial then after the "Correct!" message a scoring such as "%50" appears in the frame at the top of the page, showing that the student got the correct answer after second trial.

Fig. 5.36 shows the first question in chapter 3, where the correct answer is found after the second trial.

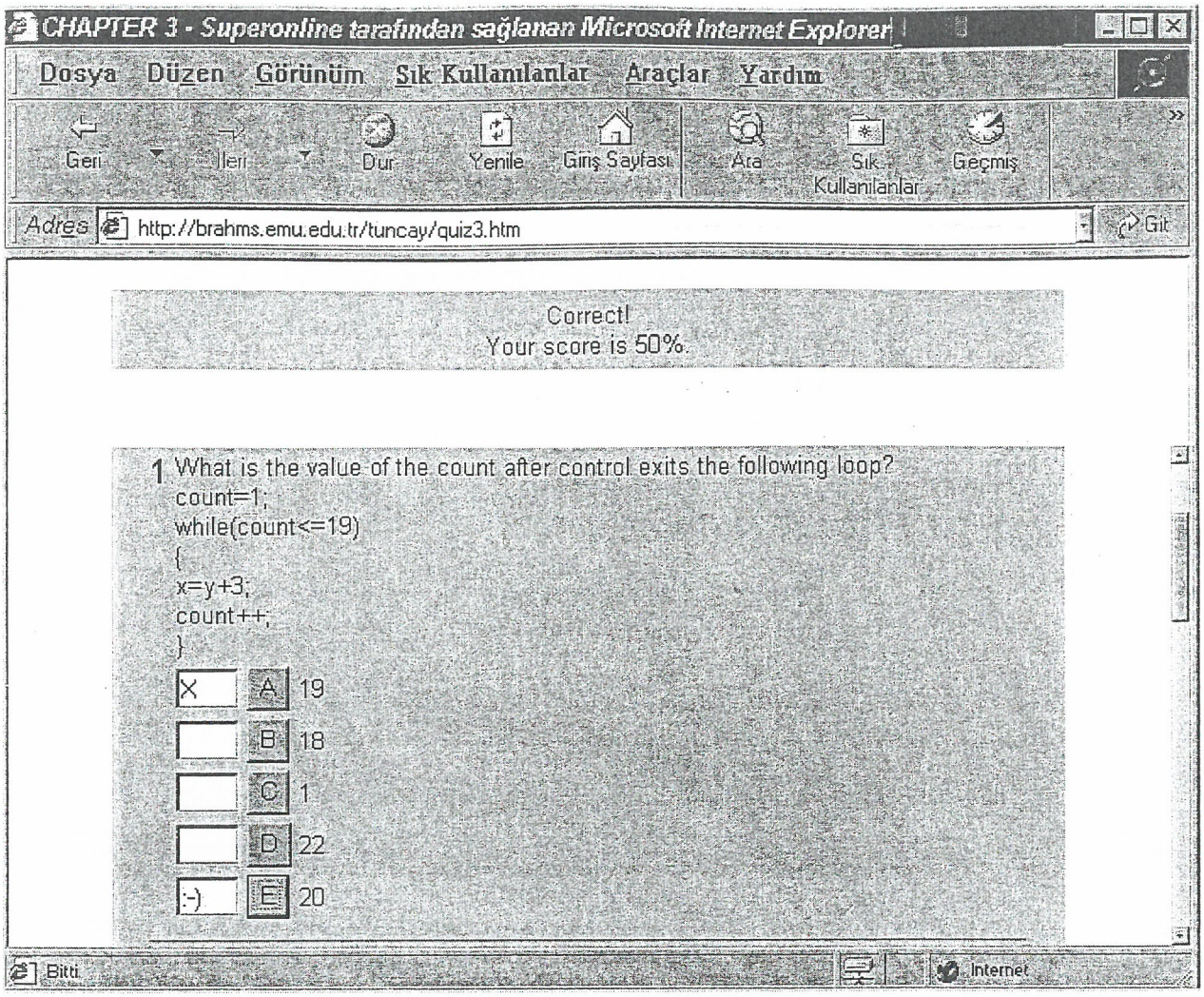

Figure 5.36 Finding the correct answer after second trial

If the user gets the correct answer after fourth trial, then the "Correct!" message and a scoring such as "%25" appears at the top of the page, showing that the student got the correct answer after fourth trial.

Fig. 5.37 shows the second question in chapter 4, where the correct answer is found after fourth trial. It is possible to see in the figure also the "X" that appears after each wrong answer and ":-)" appears when the correct answer is found, with the message "Correct" and scoring "%25".

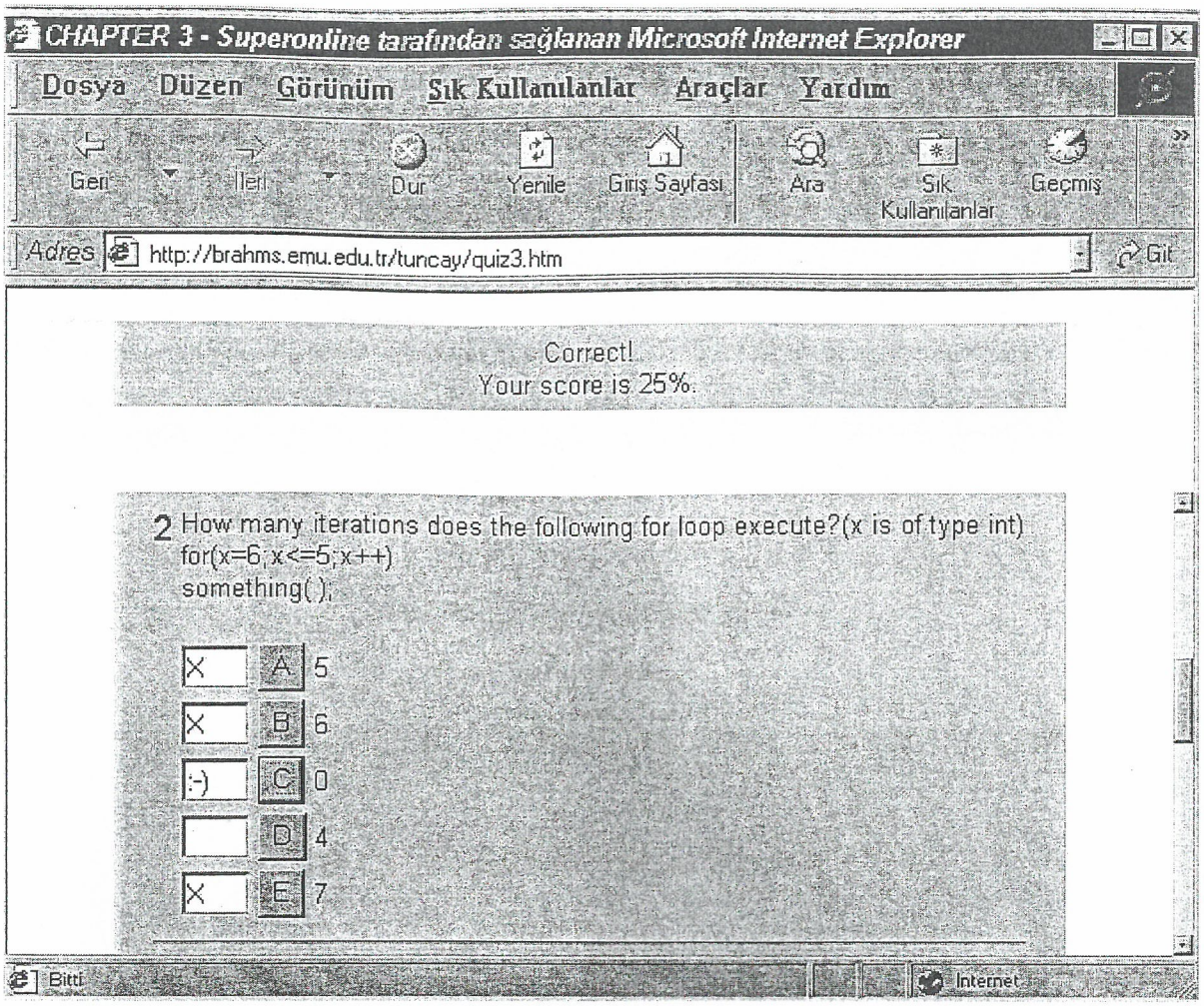

Figure 5.37 Finding the correct answer after fourth trial

By using **JQuiz** programs, **short-answer quizzes** are prepared and they are exported to HTML These quizzes can be reached from the "NEXT" button of the multiple choice questions, in each chapter. These quizzes, contains an upper frame like the multiple choice questions which have 3 buttons "BACK", "CHAPTER1" and "CHAPTER2". There is a frame at the bottom of each question, containg "Check" and "Show Answer" buttons. These buttons are used for checking the correctness of the answer given and for seeing the correct answer, respectively.

Fig. 5.38 shows the first question of the chapter1 short-answer quiz. If the "BACK" button is links to the multiple-choice questions.

"CHAPTER1" and "CHAPTER2" buttons links to the chapter 1 and chapter 2 "course outlines", respectively.

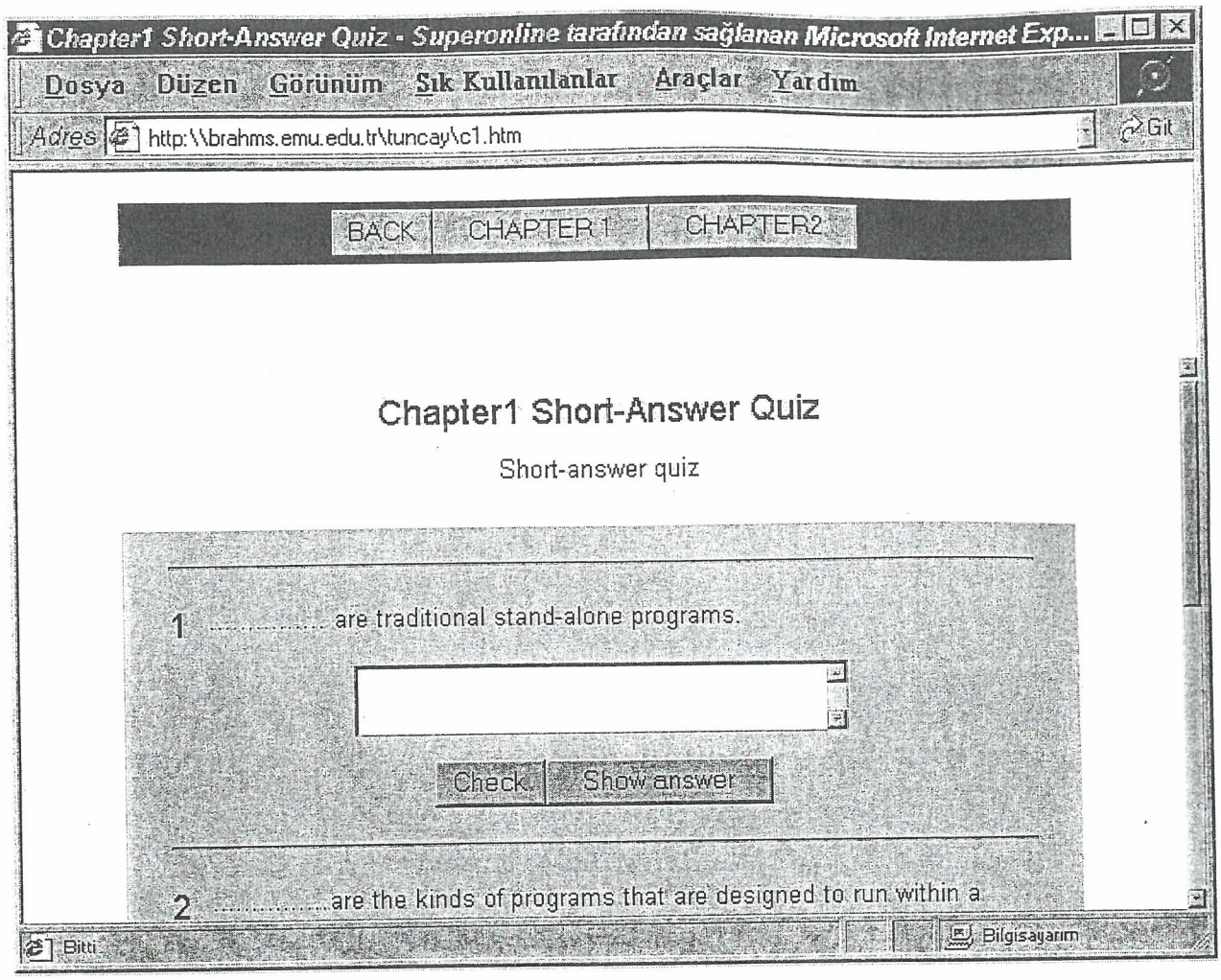

Figure 5.38 Chapter 1 short-answer quiz

The correct answers in short-answer quizzes are expected to be writen to the "white textarea" just after the question. Then by pressing the "Check" button one can check whether the answer is correct.

Fig. 5.39 shows the first question in the short-answer quiz for chapter 3. The answer "28" is writen to the "white textarea" and then "Check" button is pressed to observe that the answer is correct. Thus, "Correct!" message appears in the frame at the top of the page.

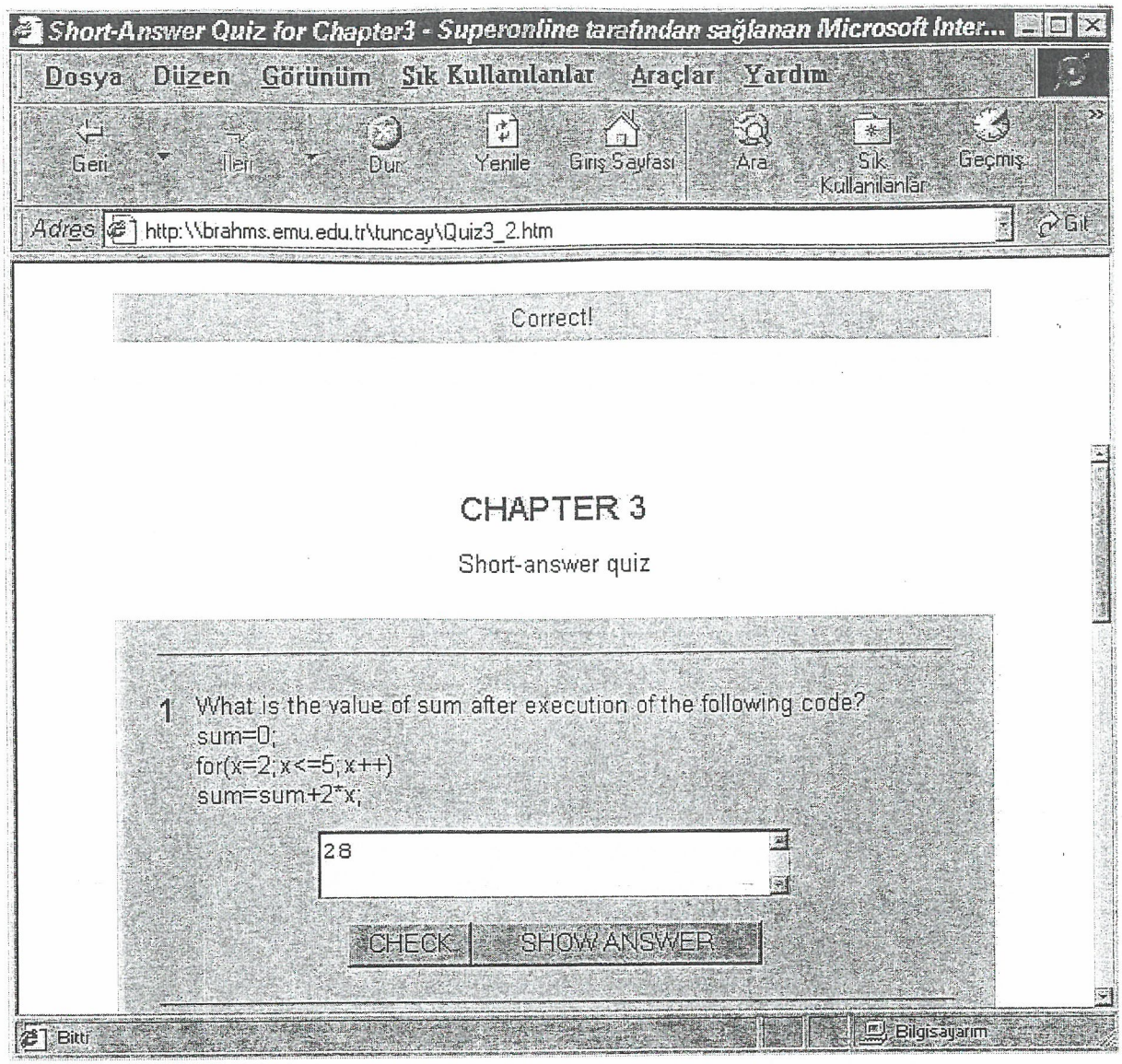

Figure 5.39 Response to a correct answer in the short-answered quiz

On the other hand, if the one cannot find the correct answer, by clicking the "SHOW ANSWER" button, the correct answer can be seen.

Fig. 5.40 shows the action of the "Show Answer" button for the first question of the short-answer quiz from the chapter 3. When the button is clicked, "Correct answer is :28" appears in the frame, at the top of the page.

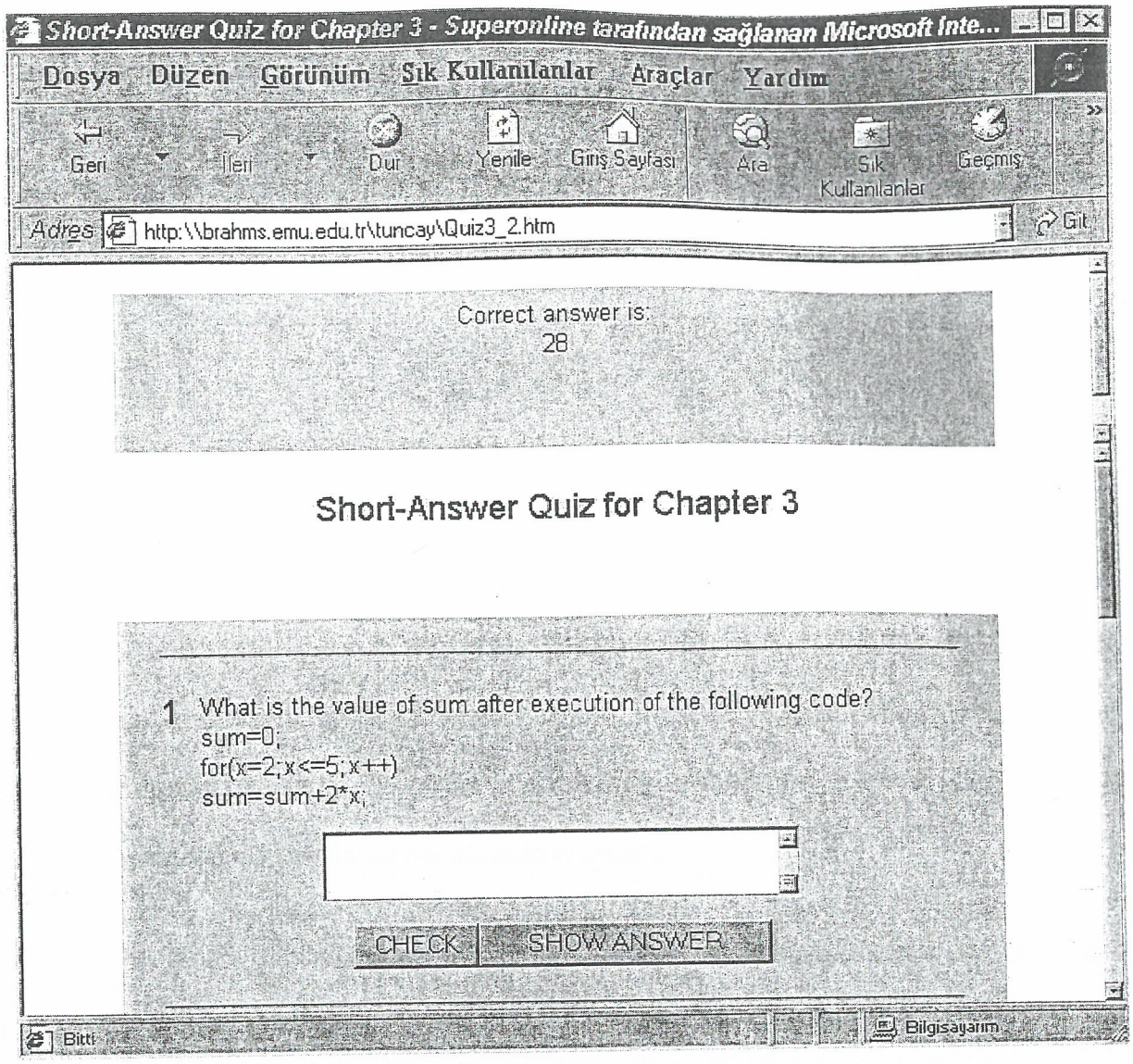

Figure 5.40 The action of the "SHOW ANSWER" button

Thus, the quizzes are arranged in a way that students can stop the quiz any time, and pass to the course notes. Also, new quiz can be started anytime by the help of the buttons, throughout the text.

Another comment here on these quizzes is that: if they had been "exported to the DHTML", since the order of the questions and the correct answer will be suffled each time, the probability of the students memorising the questions and their answers will be less. This is especially important for the students taking the same quiz more than once.

6 CONCLUSION

As educational institutions extend their campuses, as the population ages, and as the advance of technology requires a greater need for worker retraining, colleges increasingly have to cater to a different type of student in order to stay ahead and serve their communities. And at this stage, Distance Education and Distance Educators becomes two important isues. Although there are many other ways of developing distance education courses, developing a distance education course using Java and JavaScript languages constitues an excellent choice.

In this thesis, a distance education course is prepared that contain "Java Language" course materials by using Java, JavaScript, Microsoft FrontPage, HTML and Paint Shop Pro 7.0 and Hot-Potatoes facilities.

There are many other features of Java that as one increases his/her knowledge of Java language, sees that lots of excellent things can be done. In other words, better Distance Education courses can be designed by somebody who is an expert in Java language, J Builder, Microsoft Visual J++ and Symantec Visual Café. However, for the beginning, the course prepared for this thesis is sufficient for the purpose of teaching the Java Language as a distance education course.

APPENDIX Sample HTML Codes

This appendix contains the html notes of the online Java course web pages, available on the http://brahms.emu.edu.tr/tuncay

Figs A.1- A.3 shows the HTML notes of the **"Javapage.htm"** which containts links to online Java course, javagames and java software. There are 3 Java applets on this web page, written in bold.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="NAZIME" content="Java Page">
<title>Nazime Tuncay's Page</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCFFFF">
<span style="left: 326; top: 6; position: absolute">
<applet code="Clock2.class" width=150 height=110> </applet>
</span>
<p><span style="left: 26; top: 21; position: absolute"><img border="0" src="javalogo52x88.gif"
width="52" height="88"></span> ; </p>
<p> </p>
<p> </p>
<hr>
<span style="left: 47; top: 193; position: absolute"><applet code="Welcome.class" width=423
 height=50> <param name="credits" value="Nazime Tuncay">
<param name=text value="Welcome to the Java WEBPAGE">
</applet>
</span>
<p> </p>
<div style="left: 8; top: 164; width: 509; height: 19; position: absolute">
<div style="left: 2; top: 38; width: 509; height: 19; position: absolute">
</div>
</div>
<p> <i><font size="2" color="#800000"> </font></i></p>
<p><i><b><font size="2">Click tothe Java  applet  to stop animation! 
Double click to  start  again </font></b></i><b> </b></p>
<hr>
<p>
<span style="left: 102; top: 346; position: absolute">
<applet code="WhatIs.java" width=300  height=175>
<param name="credits" value="Nazime Tuncay">
</applet>
```

Figure 1 First page of the *Javapage*

```
</span>
</p>
<p> </p>
<div style="left: 12; top: 409; width: 509; height: 19; position: absolute">
</div>
<div style="left: 26; top: 627; width: 476; height: 258; position: absolute">
<table width=476 height=1>
<tr>
<td align=center valign=top height="24" width="468">
<p align="left"><a href="http://java.sun.com"><img border="0" src="aniduke.gif"
 width="55"   height="68"></a>Java is a project of Sun Microsystems.</p>
<p align="left">Do you want to <font size="6"> install Java ? </font> Click
 <font color="#800000"> ME</font> to go to the Sun
Microsystems Web Page.   </p>
<p align="left"> </p>
<p align="left"> </p>
</table>
</div>
<hr>
<p align="left"> </p>
<p align="left"> </p>
<p align="left"> </p>
<p align="left"> </p>
<p align="left"> </p>
<p align="left"> </p>
<hr>
<p align="left"><a href="contents.html"><img border="0" src="cpu_8_4kb.gif"
 width="72"   height="70"></a>Want to<font size="6"> learn
Jav</font><font size="6">a</font> Language????
Click <font color="#800000"> ME</font> to go to Nazime Tuncay's Online Java course.</p>
<p align="left"> </p>
<p align="left"> </p>
<hr>
<p align="left"> </p>
<p align="left"><img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14"   height=”14”> 
<img border="0" src="ball.gif" width="14" height=”4”> 
<img border="0" src="ball.gif" width="14" height=”4”> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14">  </p>
<p align="left">   <b><i><font size="6" color="#800000">
<a href="http://thinks.com/games/">Click
Here To play Java  GAMES</a></font></i></b></p>
<p align="left"> <img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height=”4”> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
```

Figure 2 Second page of the *Javapage*

```
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"> 
<img border="0" src="ball.gif" width="14" height="14"></p>
<p align="left">   
</td>
</tr>
</table>
<p> </p>
<span style="left: 68; top: 219; position: absolute">
<param name="credits" value="Nazime Tuncay">
<span style="left: 42; top: 20; position: absolute">
<span style="left: 7; top: 31; position: absolute">
 
</span>
</span>
</span>
</body>
</html>
```

Figure 3 Third page of the *Javapage*

Figs.A.4 shows the first page html notes of **"Contents.html** "of online Java course. (It is lengthy, so the rest is not included in the appendix.)

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="Nazime Tuncay's Page" content="">
<title>ONLINE JAVA COURSE</title>
<script language="JavaScript" fptype="dynamicanimation">
<!--
function dynAnimation() {}
function clickSwapImg() {}
//-->
</script>
<script language="JavaScript1.2" fptype="dynamicanimation" src="animate.js">
</script>
<meta http-equiv="Page-Enter" content="revealTrans(Duration=1.0,Transition=5)">
</head>
<body background="a9.jpg" onload="dynAnimation()">
<h1 align="center" dynamicanimation="fpAnimdropWordFP1" id="fpAnimdropWordFP1"
style="position: relative !important; visibility: hidden" language="Javascript1.2">ONLINE JAVA
COURSE</h1><></h1>
<table width=531 height=90%>
<tr>
<td align=CENTER valign=CENTER width="523">
<h3 style="text-align:justify" dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="border-
style: solid" onmouseover="rollIn(this)" onmouseout="rollOut(this)" language="Javascript1.2"><font
size="4"><span style="mso-bidi-font-weight: normal; font-family: Times New Roman; color: maroon" lang="EN-
AU">CHAPTER
1 . I<span lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;;color:maroon"><b style="mso-bidi-
font-weight:normal">INTRODUCTION<o:p>
</b></span></span></font><span lang="EN-AU" style="font-family:&amp;quot;Times New
Roman&amp;quot;;color:maroon"><b style="mso-bidi-font-weight:
normal"><font size="4"></o:p>
</font></b>
</span></h3>
<p class="MsoNormal" style="text-align: justify"><font size="4"><span style="font-family:&amp;quot;Times New
Roman&amp;quot;;
color:maroon">1.1</span><span style="font-family:&amp;quot;Times New Roman&amp;quot;"> <a href="1_1.htm"
dynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="color: #800000; font-weight: bold"
onmouseover="rollIn(this)" onmouseout="rollOut(this)" language="Javascript1.2">
Object Oriented Pro<o:p>gramming </a></o:p>
</span></font></p>
<p class="MsoNormal" style="margin-left:21.0pt;text-align:justify;text-indent:
-21.0pt;mso-list:l119 level2 lfo4;tab-stops:list 21.0pt"><font size="4"><span style="font-family:&amp;quot;Times New
Roman&amp;quot;;color:maroon">1.2<span style="font-style: normal; font-variant: norm l; font-weight: normal; font-
family: Times New Roman">&amp;nbsp;&amp;nbsp;&amp;nbsp;
</span></span><span style="font-family:&amp;quot;Times New Roman&amp;quot;"><a href="1_2.HTM"
adynamicanimation="fpAnimformatRolloverFP1" fprolloverstyle="color: #800000; font-weight: bold"
onmouseover="rollIn(this)" onmouseout="rollOut(this)" language="Javascript1.2">What is Java?<o:p>
```

Figure 4 First page of the *Contents* of the online Java course

The following figures Fig.A.17-A.21 are the html notes of the **"Exercises1.htm"** of chapter 1.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>EXERCISES</title>
</head>
<body>
<h2 align="left"><span lang="EN-AU" style="font-family:&quot;Times New Roman&quot;;color:maroon;font-
style:normal"><font size="4">EXERCISES<o:p>:
</font><u>
0</o:p>
</u>
</span></h2>
<p class="MsoNormal" style="margin-left:18.0pt;text-align:justify;text-indent:
-18.0pt;mso-list:l229 level2 lfo191;tab-stops:list 18.0pt"><b style="mso-bidi-font-weight:normal"><span
style="font-family:&quot;Times New Roman&quot;;
color:maroon">1.1<span style="font:7.0pt &quot;Times New Roman&quot;">  </span></span></b><span
style="font-family:&quot;Times New Roman&quot;">Describe
the meanings of the following terms:<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><b style="mso-bidi-font-weight:
normal"><span style="font-family:&quot;Times New Roman&quot;;color:maroon">a)</span></b><span
style="font-family:&quot;Times New Roman&quot;">
Java<span style="mso-spacerun: yes">     </span><b style="mso-bidi-font-
weight:normal"><span style="color:maroon">b)</span></b>
Applet<span style="mso-spacerun: yes">      </span><b style="mso-bidi-font-
weight:
normal"><span style="color:maroon">c)</span></b>Application<span style="mso-spacerun:
yes">       
</span><b style="mso-bidi-font-weight:normal"><span style="color:maroon">d)</span></b>Methods<span
style="mso-spacerun: yes">    
</span><b style="mso-bidi-font-weight:normal"><span style="color:maroon">e)</span></b>Fields<span
style="mso-spacerun: yes">      
</span><b style="mso-bidi-font-weight:normal"><span style="color:maroon">f )</span></b><span
style="color:maroon">
</span>HTML<b style="mso-bidi-font-weight:normal"><span style="color:maroon"><o:p>
</o:p>
</span></b></span></p>
<p class="MsoNormal" style="text-align:justify"><b style="mso-bidi-font-weight:
normal"><span style="font-family:&quot;Times New Roman&quot;;color:maroon">1.2 </span></b><span
style="font-family:&quot;Times New Roman&quot;">The<span style="color:maroon">
</span>following program is saved as Application.java. Find the errors in this
program and correct them.<o:p></o:p>
```

Figure 5 First page of the *exercises* of the chapter 1

```
</span></p>
<p class="MsoNormal" style="text-align: justify"><img border="0" src="CH1Exe27.gif" width="376"
height="136"></p>
<h2 style="text-align:justify"><font size="3"><span style="font-family: Times New Roman; font-style: normal"
lang="EN-AU">1.3<span style="font-family:
&amp;quot;Times New Roman&amp;quot;;font-weight:normal;font-style:normal"> The following
program is saved as Program.java. Find the errors in this program and correct
them</span></span></font></h2>
<h2 style="text-align:justify"><img border="0" src="CH1Exe28.gif" width="386" height="144"></h2>
<h2 style="text-align:justify"><span lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;;
color:maroon;font-style:normal"><font size="3">1.4 </font> </span><span lang="EN-AU" style="font-family:
&amp;quot;Times New Roman&amp;quot;;font-weight:normal;font-style:normal"><font size="3">What is the
output of the following program if<o:p> </font>
</o:p>
</span></h2>
<p class="MsoNormal" style="text-align:justify"><b style="mso-bidi-font-weight:
normal"><span style="font-family:&amp;quot;Times New Roman&amp;quot;;color:maroon">a) </span></b><span
style="font-family:&amp;quot;Times New Roman&amp;quot;">We
run the program as<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><i style="mso-bidi-font-style:
normal"><span style="font-family:&amp;quot;Times New Roman&amp;quot;">java Books</span></i><span style="font-
family:&amp;quot;Times New Roman&amp;quot;;color:maroon"><o:p>
</o:p>
</span></p>
<p class="MsoFooter" style="text-align:justify;tab-stops:36.0pt"><b style="mso-bidi-font-weight:normal"><span
lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;;
color:maroon">b)</span></b><span lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;">
We run the program as<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><i style="mso-bidi-font-style:
normal"><span style="font-family:&amp;quot;Times New Roman&amp;quot;">java Books<span style="mso-spacerun:
yes"> 
</span>English</span></i><span style="font-family:
&amp;quot;Times New Roman&amp;quot;;color:maroon"><o:p>
</o:p>
</span></p>
<p class="MsoFooter" style="text-align:justify;tab-stops:36.0pt"><b style="mso-bidi-font-weight:normal"><span
lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;;
color:maroon">c)</span></b><span lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;">
We run the program as<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><i style="mso-bidi-font-style:
normal"><span style="font-family:&amp;quot;Times New Roman&amp;quot;">java Books English
Mathematics<span style="color:maroon"><o:p>
</o:p>
</span></span></i></p>
<p class="MsoFooter" style="text-align:justify;tab-stops:36.0pt"><b style="mso-bidi-font-weight:normal"><span
lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;;
color:maroon">d)</span></b><span lang="EN-AU" style="font-family:&amp;quot;Times New Roman&amp;quot;">
We run the program as<o:p>
</o:p>
```

Figure 6 Second page of the *exercises* of the chapter 1

```
</span></p>
<i style="mso-bidi-font-style:normal"><span style="font-size:12.0pt;mso-bidi-font-size:
10.0pt;font-family:&quot;Times New Roman&quot;;mso-fareast-font-family:&quot;Times New Roman&quot;;
mso-ansi-language:EN-US;mso-fareast-language:EN-US;mso-bidi-language:AR-SA">java
Books English<span style="mso-spacerun: yes">  </span>Computer Mathematics</span></i>
<h2 style="text-align:justify"><span lang="EN-AU" style="font-family:
&quot;Times New Roman&quot;;font-weight:normal;font-style:normal"><o:p>
</o:p>
<img border="0" src="CH1Exe29.gif" width="540" height="136"></span></h2>
<h2 style="text-align:justify"><span style="font-family: Times New Roman; font-style: normal"><font
size="3"><span lang="EN-AU">1.5</span><span lang="TR" style="font-family:
&quot;Times New Roman&quot;;mso-ansi-language:TR;font-weight:normal;font-style:normal">Write
a Java application in order to output the following: </span><span lang="EN-AU" style="font-family:&quot;Times
New Roman&quot;;font-weight:normal;font-style:normal">(Hint
:Use commandline arguments</span><span lang="TR" style="font-family:&quot;Times New Roman&quot;;
mso-ansi-language:TR;font-weight:normal;font-style:normal">)</span></font></span></h2>
<h2 style="text-align:justify"><img border="0" src="CH1Exe30.gif" width="396" height="194"></h2>
<h2 style="text-align:justify"><font size="3"><span style="font-family: Times New Roman; font-style: normal"
lang="EN-AU">1.6
<span style="font-family:
&quot;Times New Roman&quot;;font-weight:normal;font-style:normal">The following
program is compiled as javac Exercise.java and run as java Exercise. What is the
output of the following program?</span></span></font><span lang="EN-AU" style="font-family:
&quot;Times New Roman&quot;;font-weight:normal;font-style:normal"><o:p>
</o:p>
</span></h2>
<h2 style="text-align:justify"><span lang="TR" style="font-family:&quot;Times New Roman&quot;;
mso-ansi-language:TR;font-weight:normal;font-style:normal"><o:p>
</o:p>
<img border="0" src="CH1Exe31.gif" width="444" height="151"></span></h2>
<h2 style="text-align:justify"><font size="3"><span style="font-family: Times New Roman"><span lang="EN-
AU"><font color="#800000">1.7</font> </span><span lang="EN-AU" style="font-family:
&quot;Times New Roman&quot;;font-weight:normal;font-style:normal">Write a Java
application and an applet to output<o:p>
</o:p>
</span></span></font></h2>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Times New
Roman&quot;"><font size="3">Hello<o:p>
 </font>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Times New
Roman&quot;">Java<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><b style="mso-bidi-font-weight:
normal"><span style="font-family:&quot;Times New Roman&quot;;color:maroon">1.8</span></b><span
style="font-family:&quot;Times New Roman&quot;">
Write a Java application to output on the screen<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Times New
Roman&quot;">Hello
NAZIME<o:p>
</o:p>
</span></p>
```

Figure 7 Third page of the *exercises* of the chapter 1

```
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Times New
Roman&quot;"> <o:p>
</o:p>
</span></p>
<p class="MsoFooter" style="text-align:justify;tab-stops:36.0pt"><span lang="EN-AU" style="font-
family:&quot;Times New Roman&quot;">HI,
your name<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><b style="mso-bidi-font-weight:
normal"><span style="font-family:&quot;Times New Roman&quot;;color:maroon">1.9 </span></b><span
style="font-family:&quot;Times New Roman&quot;">Write
a Java application to print the picture<b style="mso-bidi-font-weight:normal"><i style="mso-bidi-font-style:
normal"> </i></b>using asterisks (*) as follows:<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="margin-left:18.0pt;text-align:justify"><span style="font-family:&quot;Courier
New&quot;;mso-bidi-font-family:&quot;Times New Roman&quot;">*********<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="margin-left:18.0pt;text-align:justify"><span style="font-family:&quot;Courier
New&quot;;mso-bidi-font-family:&quot;Times New Roman&quot;">*******<o:p>
</o:p></span></p>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Courier New&quot;;
mso-bidi-font-family:&quot;Times New Roman&quot;">******<o:p>
</o:p></span></p>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Courier New&quot;;
mso-bidi-font-family:&quot;Times New Roman&quot;">***<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><span style="font-family:&quot;Courier New&quot;;
mso-bidi-font-family:&quot;Times New Roman&quot;">*<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify;tab-stops:276.45pt"><b style="mso-bidi-font-
weight:normal"><span style="font-family:&quot;Times New Roman&quot;;
color:maroon">1.10</span></b><span style="font-family:&quot;Times New Roman&quot;;
color:maroon"> </span><span style="font-family:&quot;Times New Roman&quot;">Write
a Java applet to output the picture below. Execute the applet by using the
appletviewer provided in the JDK.<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><span lang="TR" style="font-family:
&quot;Courier New&quot;;mso-bidi-font-family:&quot;Times New Roman&quot;;mso-ansi-language:TR">*
* * *<span style="mso-spacerun: yes">        
</span>**<o:p>
</o:p>
</span></p>
<p class="MsoNormal" style="text-align:justify"><span lang="TR" style="font-family:
&quot;Courier New&quot;;mso-bidi-font-family:&quot;Times New Roman&quot;;mso-ansi-language:TR">*
<span style="mso-spacerun: yes">    
</span>*<span style="mso-spacerun: yes">       
</span>****<o:p>
</o:p>
</span></p>
```

Figure 8 Fourth page of the *exercises* of the chapter 1

```
<p class="MsoNormal" style="text-align:justify"><span lang="TR" style="font-family:
&amp;quot;Courier New&amp;quot;;mso-bidi-font-family:&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">*
* * *<span style="mso-spacerun: yes">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;
</span>**<o:p>
</o:p>
</span></p>
<p class="MsoNormal"><b style="mso-bidi-font-weight:normal"><span lang="TR" style="font-
family:&amp;quot;Times New Roman&amp;quot;;color:maroon;mso-ansi-language:TR">1.11
</span></b><span lang="TR" style="font-family:&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">Write
the<span style="mso-spacerun: yes">&amp;nbsp; </span>Java code and Html code<span style="mso-spacerun:
yes">&amp;nbsp;
</span>that can be used in order to display the following output in </span><b style="mso-bidi-font-
weight:normal"><span lang="TR" style="font-family:&amp;quot;Courier New&amp;quot;;mso-bidi-font-
family:&amp;quot;Times New Roman&amp;quot;;
mso-ansi-language:TR">F</span><span lang="TR" style="font-size:10.0pt;font-family:&amp;quot;Courier
New&amp;quot;;mso-bidi-font-family:
&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">igure11</span></b><span lang="TR" style="font-
family:&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">
in a web page.<o:p>
</o:p>
</span></p>
<p class="MsoNormal"><b style="mso-bidi-font-weight:normal"><span lang="TR" style="font-
family:&amp;quot;Times New Roman&amp;quot;;color:maroon;mso-ansi-language:TR">a)</span></b><span lang="TR"
style="font-family:&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">Let
Param tag be “See you next chapter”<b style="mso-bidi-font-weight:normal"><o:p>
</o:p>
</b></span></p>
<p class="MsoNormal"><b style="mso-bidi-font-weight:normal"><span lang="TR" style="font-
family:&amp;quot;Times New Roman&amp;quot;;color:maroon;mso-ansi-language:TR">b)</span></b><span lang="TR"
style="font-family:&amp;quot;Times New Roman&amp;quot;;mso-ansi-language:TR">Let
Param tag be “See you tomorrow”<o:p>
</o:p>
</span></p>
<p class="MsoNormal"><span lang="TR" style="font-family:&amp;quot;Times New Roman&amp;quot;;
mso-ansi-language:TR">&amp;nbsp;</span><img border="0" src="CH1Exe32.gif" width="493" height="315"></p>
<p
class="MsoNormal">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp
;&amp;nbsp;&amp;nbsp;<font size="4"><a href="chapter1.HTM"><img border="0" src="go_prev_.gif" width="62"
height="56"></a></font>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;
<font size="4"><a href="chapter2.htm"><img border="0" src="go_next_.gif" width="58" height="53"></a></font>
</p>
<p><applet code=”NazimeButtonA1.class width=80 height=30></applet></p>
</body>
</html>
```

Figure 9 Fifth page of the *exercises* of the chapter 1

“quiz2.htm” is the multiple choice quiz for chapter 2. It calls two other programs “quiz2c.htm” and “quiz2w.htm”. Figs. A.10-A.18 shows the HTML notes of “quiz2.htm”, “quiz2c.htm” and “quiz2w.htm”

```
<html>
<head><meta name="author" content="created with hot potatoes by half-baked software, registered to
unregistered.">
<title>
chapter 2
</title>
</head>
<frameset rows="100,*" frameborder=no framespacing=0 border=0>
        <frame src="quiz2w.htm" name="feedbackframe">
        <frame src="quiz2c.htm" name="codeframe">
</frameset>
</html>
```

Figure 10 Multiple-choice quiz at the end of the chapter 2

```
<HTML>
<HEAD><META NAME="Author" CONTENT="Created with Hot Potatoes by Half-Baked Software,
registered to UNREGISTERED.">
<TITLE>CHAPTER 2</TITLE>
<SCRIPT language="JavaScript">var min
var sec
var ShowReadingAgain = 'Show reading again'
var platform = 'win'
if (navigator.appVersion.indexOf('Mac') != -1) {platform = 'mac'}
macrightchar = ':-)'
macwrongchar = 'X'
winrightchar = ':-)'
winwrongchar = 'X'
DoneStatus = new Array();
DoneStatus[1] = '0';
DoneStatus[2] = '0';
DoneStatus[3] = '0';
Responses = new Array();
Responses[1]='Empty space not allowed!';
Responses[2]='Appostraphe NOT allowed';
Responses[3]='';
Responses[4]='';
Responses[5]='Hyphen is not an alphanumeric character!';
Responses[6]='';
Responses[7]='';
Responses[8]='';
Responses[9]='';
Responses[10]='';
Responses[11]='';
Responses[12]='';
Responses[13]='';
Responses[14]='';
Responses[15]='';
IndicatorName = new Array();
IndicatorName[1] = 'document.QuizForm.Q1A';
IndicatorName[2] = 'document.QuizForm.Q1B';
IndicatorName[3] = 'document.QuizForm.Q1C';
IndicatorName[4] = 'document.QuizForm.Q1D';
IndicatorName[5] = 'document.QuizForm.Q1E';
```

Figure11 First page of the *Quiz2c*

```
IndicatorName[6] = 'document.QuizForm.Q2A';
IndicatorName[7] = 'document.QuizForm.Q2B';
IndicatorName[8] = 'document.QuizForm.Q2C';
IndicatorName[9] = 'document.QuizForm.Q2D';
IndicatorName[10] = 'document.QuizForm.Q2E';
IndicatorName[11] = 'document.QuizForm.Q3A';
IndicatorName[12] = 'document.QuizForm.Q3B';
IndicatorName[13] = 'document.QuizForm.Q3C';
IndicatorName[14] = 'document.QuizForm.Q3D';
IndicatorName[15] = 'document.QuizForm.Q3E';
var Score = 0;
var Tries = 0;
var Percent = 0;
var YourScore = 'Your score is ';
var DefaultRight = 'Correct!';
var DefaultWrong = 'Sorry! Try again.';
function DownTime(mm,ss){
        min = mm;
        sec = ss;
        if (sec==0) {
                sec = 60;
                min--;
        }
        sec--;
        if (sec<10){
                sec="0" + sec;
        }
        parent.TimerFrame.document.TimerForm.face.value= min+":"+sec;
        if ((min==0) && (sec==0)){
                TimesUp()
        }
        else{
                setTimeout('DownTime(min,sec)',1000)
        }
}
function SetUp(){
        if (parent.TimerFrame != null) {
                Stuff='<INPUT TYPE="Button"';
                Stuff+=' NAME="StartIt" VALUE="Start reading"
onClick="parent.CodeFrame.StartReading()">';
                WriteToReading('Click on the button to see the reading.');
                WriteToTimer(Stuff);
        }
}
function StartReading(){
        Stuff='Remaining time:';
        Stuff+=' <FONT SIZE="3"><INPUT TYPE="text" NAME="face" SIZE="6"
VALUE="1:00"></FONT>';
        WriteToTimer(Stuff);
        parent.ReadingFrame.location.href='Quiz2r.htm';
        timeoutID=setTimeout('DownTime(1,0)',1000);
}
function TimesUp() {
        if (ShowReadingAgain.length > 0) {
                Stuff='<INPUT TYPE="Hidden" NAME="ThisAvoidsNavigator3Bug">';
                Stuff+='<INPUT TYPE="Button" NAME="StartIt" VALUE="Show reading again"
```

Figure 12 Second page of the *Quiz2c*

```
onClick="parent.CodeFrame.StartReading()">';
WriteToTimer(Stuff);
}
        WriteToReading('Your time is over!');
}
function WriteToReading(Stuff) {
        parent.ReadingFrame.document.clear();
        parent.ReadingFrame.document.open();
        parent.ReadingFrame.document.writeln ("<HTML>");
        parent.ReadingFrame.document.writeln ("<BODY  BACKGROUND=\"image.gif\"
BGCOLOR=\"#FFFFFF\" TEXT=\"#000033\">");
        parent.ReadingFrame.document.writeln ("<CENTER><TABLE BORDER=\"0\" CELLPADDING=\"5\"
CELLSPACING=\"1\" WIDTH=\"85%\">");
        parent.ReadingFrame.document.writeln ("<TR><TD BGCOLOR=\"#BBBBEE\"
ALIGN=\"CENTER\">")
        parent.ReadingFrame.document.writeln ("<FONT FACE=\"Geneva,Arial\" SIZE=\"2\">");
        parent.ReadingFrame.document.writeln (Stuff);
        parent.ReadingFrame.document.writeln
("</FONT></TD></TR></TABLE></CENTER></BODY></HTML>");
        parent.ReadingFrame.document.close();
}
function WriteToTimer(Stuff){
        parent.TimerFrame.document.clear();
        parent.TimerFrame.document.open();
        parent.TimerFrame.document.writeln ("<HTML>");
        parent.TimerFrame.document.writeln ("<BODY  BACKGROUND=\"image.gif\"
BGCOLOR=\"#FFFFFF\" TEXT=\"#000033\">");
        parent.TimerFrame.document.writeln ("<CENTER><TABLE BORDER=\"0\" CELLPADDING=\"5\"
CELLSPACING=\"1\" WIDTH=\"85%\">");
        parent.TimerFrame.document.writeln ("<FORM NAME=\"TimerForm\"><TR><TD
BGCOLOR=\"#BBBBEE\" ALIGN=\"CENTER\">")
        parent.TimerFrame.document.writeln ("<FONT FACE=\"Geneva,Arial\" SIZE=\"2\">");
        parent.TimerFrame.document.writeln (Stuff);
        parent.TimerFrame.document.writeln
("</FONT></TD></TR></FORM></TABLE></CENTER></BODY></HTML>");
        parent.TimerFrame.document.close();
}
if (platform == 'mac') {
        rightchar = unescape(macrightchar)
        wrongchar = unescape(macwrongchar)
}
else {
        rightchar = unescape(winrightchar)
        wrongchar = unescape(winwrongchar)
}
function CheckAnswer(QNum, AnsNum, Correct) {
whichfb = (5*(QNum-1) + AnsNum)
 if (Correct == 1) {
        if (DoneStatus[QNum] != '1'){
                Tries = Tries + 1;
                DoneStatus[QNum] = '1';
                Score = Score + 1;}
        if (eval(IndicatorName[whichfb]) != null){
                eval(IndicatorName[whichfb]).value = rightchar}
        Feedback = Responses[whichfb]
        if (Feedback == '') {Feedback = DefaultRight}
```

Figure 13 Third page of the*Quiz2c*

```
Percent = Math.floor(Score*100/Tries) + '%';
            Feedback = Feedback + '<BR>' + YourScore + ' ' + Percent + '.';
            var i = 0;
var AllDone = true;
for (i=1; i<DoneStatus.length; i++){
                        if (DoneStatus[i] == '0'){
                                    AllDone = false;
                        }
            }
            if (AllDone == true){
                        Feedback += '<BR><FORM><INPUT TYPE=\"button\" VALUE=\"NEXT\"
onClick=\"parent.location=\'Quiz2_2.htm\'\"></FORM>';
            }
            WriteFeedback(Feedback);
            }
 else {
            if (DoneStatus[QNum] != '1'){
                        Tries = Tries + 1;}
            if (eval(IndicatorName[whichfb]) != null){
                        eval(IndicatorName[whichfb]).value = wrongchar}
            Feedback = Responses[whichfb]
            if (Feedback == '') {Feedback = DefaultWrong}
            WriteFeedback(Feedback)
 }
}
function WriteFeedback(Feedback) {
            parent.FeedbackFrame.document.clear();
            parent.FeedbackFrame.document.open();
            parent.FeedbackFrame.document.writeln ("<HTML>");
            parent.FeedbackFrame.document.writeln ("<BODY  BACKGROUND=\"image.gif\"
BGCOLOR=\"#FFFFFF\" TEXT=\"#000033\">");
            parent.FeedbackFrame.document.writeln ("<CENTER><TABLE BORDER=\"0\"
CELLPADDING=\"5\" CELLSPACING=\"1\" WIDTH=\"85%\">");
            parent.FeedbackFrame.document.writeln ("<TR><TD BGCOLOR=\"#BBBBEE\"
ALIGN=\"CENTER\">")
            parent.FeedbackFrame.document.writeln ("<FONT FACE=\"Geneva,Arial\" SIZE=\"2\">");
            parent.FeedbackFrame.document.writeln (Feedback);
            parent.FeedbackFrame.document.writeln
("</FONT></TD></TR></TABLE></CENTER></BODY></HTML>");
            parent.FeedbackFrame.document.close();
}
</SCRIPT>
</HEAD>
<BODY BACKGROUND="image.gif" BGCOLOR="#FFFFFF" TEXT="#000033" onLoad="SetUp()">
<CENTER>
<TABLE BORDER="0" CELLPADDING="5" CELLSPACING="1" WIDTH="85%">
<TR><TD ALIGN="CENTER"><FONT SIZE="+1" FACE="Geneva,Arial" COLOR="#000033">CHAPTER
2</FONT></TD></TR>
<TR><TD ALIGN="CENTER"><FONT SIZE="-1" FACE="Geneva,Arial" COLOR="#000033">Choose the
correct answer for each question.<BR> </FONT></TD></TR>
<TR><TD BGCOLOR="#BBBBEE" ALIGN="CENTER">
<FORM NAME="QuizForm">
<TABLE BORDER="0" CELLPADDING="1" CELLSPACING="2" WIDTH="95%">
<TR>
<TD COLSPAN="4"><HR WIDTH="100%" HEIGHT="5">
```

Figure14 Fourth page of the *Quiz2c*

```
</TD>
</TR>
<TR>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE" ALIGN="RIGHT"><FONT SIZE="+1"
FACE="Geneva,Arial">1</FONT></TD><TD COLSPAN="3" VALIGN="TOP"
BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">Which of the following are legal String
constants?</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q1A"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" A "
OnClick="CheckAnswer(1,1,0)"></TD>
<TD WIDTH="100%" BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">red
rose</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q1B"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" B "
OnClick="CheckAnswer(1,2,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">rose's</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q1C"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" C "
OnClick="CheckAnswer(1,3,1)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">rrose</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q1D"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" D "
OnClick="CheckAnswer(1,4,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">RED ROSE</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q1E"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" E "
OnClick="CheckAnswer(1,5,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">red-rose</FONT></TD>
</TR>
<TR>
<TD COLSPAN="4"><HR WIDTH="100%" HEIGHT="5">
</TD>
</TR>
<TR>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE" ALIGN="RIGHT"><FONT SIZE="+1"
```

Figure15 Fifth page of the *Quiz2c*

```
FACE="Geneva,Arial">2</FONT></TD><TD COLSPAN="3" VALIGN="TOP"
BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">Assume a=2;b=3;c=4;d=0.What  does the
following statement print?<BR>
   System.out.println(b/a>d&&c*d<a);</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q2A"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" A "
OnClick="CheckAnswer(2,1,1)"></TD>
<TD WIDTH="100%" BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">true</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q2B"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" B "
OnClick="CheckAnswer(2,2,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">false</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q2C"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" C "
OnClick="CheckAnswer(2,3,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">yes</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q2D"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" D "
OnClick="CheckAnswer(2,4,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">no </FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q2E"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" E "
OnClick="CheckAnswer(2,5,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">An error message </FONT></TD>
</TR>
<TR>
<TD COLSPAN="4"><HR WIDTH="100%" HEIGHT="5">
</TD>
</TR>
<TR>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE" ALIGN="RIGHT"><FONT SIZE="+1"
FACE="Geneva,Arial">3</FONT></TD><TD COLSPAN="3" VALIGN="TOP"
BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">What is the error in  the following program
segment?<BR>
Array_Size=8;<BR>
```

Figure 16 Sixth page of the *Quiz2c*

```
int y[]=new int[8]<BR>
  for(int i=0;i<=y.length;i++)<BR>
    y[i]=2.0;<BR>
<BR>
</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q3A"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" A "
OnClick="CheckAnswer(3,1,0)"></TD>
<TD WIDTH="100%" BGCOLOR="#BBBBEE"><FONT SIZE="-1" FACE="Geneva,Arial">array out of
bounds</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q3B"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" B "
OnClick="CheckAnswer(3,2,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">no error</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q3C"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" C "
OnClick="CheckAnswer(3,3,1)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">type mismatch</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q3D"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" D "
OnClick="CheckAnswer(3,4,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">all of the above</FONT></TD>
</TR>
<TR>
<TD></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="TEXT" VALUE=" " SIZE="3"
NAME="Q3E"></TD>
<TD VALIGN="TOP" BGCOLOR="#BBBBEE"><INPUT TYPE="BUTTON" VALUE=" E "
OnClick="CheckAnswer(3,5,0)"></TD>
<TD WIDTH="100%"><FONT SIZE="-1" FACE="Geneva,Arial">none of the above</FONT></TD>
</TR>
<TR>
<TD COLSPAN="4"><HR WIDTH="100%" HEIGHT="5">
</TD>
</TR>
</TABLE>
</FORM>
</TD></TR>
<TR><FORM NAME="NavButtons2">
```

Figure 17 Seventh page of the *Quiz2c*

```
<TD BGCOLOR="#000033" ALIGN="CENTER">
<FONT FACE="Geneva,Arial" SIZE="2"><INPUT TYPE="button" VALUE="CHAPTER2 "
onClick="parent.location='chapter2.htm'"><INPUT TYPE="button" VALUE="NEXT"
onClick="parent.location='Quiz2_2.htm'"></FONT></TD>
</FORM></TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Figure 18 Eight page of the *Quiz2c*

```
<HTML>
<HEAD><META NAME="Author" CONTENT="Created with Hot Potatoes by Half-Baked Software, registered to
UNREGISTERED."><META NAME="Author" CONTENT="Created with Hot Potatoes by Half-Baked Software,
registered to UNREGISTERED.">
<TITLE>CHAPTER 2</TITLE>
</HEAD>
<BODY BACKGROUND="image.gif" BGCOLOR="#FFFFFF" TEXT="#000033">
<CENTER>
<TABLE BORDER="0" CELLPADDING="4" CELLSPACING="1" WIDTH="85%">
<TR>
<FORM NAME="NavButtons1">
<TD BGCOLOR="#000033" ALIGN="CENTER"><FONT FACE="Geneva,Arial" SIZE="2"><INPUT
TYPE="button" VALUE="CHAPTER2 " onClick="parent.location='chapter2.htm'"><INPUT TYPE="button"
VALUE="NEXT" onClick="parent.location='Quiz2_2.htm'"></FONT></TD>
</FORM>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Figure 19 HTML notes of Quiz2w

REFERENCES

[1] Arnold, K. and Gosling, J. , The Java Programming Language. California: Addison Wesley Longman, Inc.,1996, 373p.

[2] Cooper, J.W., Principles Of Object Oriented Programming in JAVA 1.1. The Practical Guide to Efficient Program Design. Ventana Communication Group, NC, USA.,1997, 372p.

[3] Decker, R. and Hirshfield S., programming.java. An Introduction To Programming Using Java. PWS Publishing Company, Boston, Albany, Bonn, etc., 1998, 574p.

[4] Deitel, H.M. and Deitel, P.J., Java How To Program.Second Edition, Prentice Hall, London, Sydney,etc., USA, 1997, 1063p.

[5] Grand, M., Java Language Reference.United States Of America: O'Relly & Associates, U.S.A, 1997, 448p.

[6] Grup Java,JAVA.BETA Basim Yayim Dagitim A.s. Istanbul, Türkiye,200,824p.

[7] Harold, R."*Animation". http://www.ibiblo.arg/javafag/course*

[8] Horton, I., Beginning Java.Wrox Press Ltd., Canada, 1997, 1038p.

[9] Leestma S. and Larry N., Turbo Pascal Programming and Problem Solving. Second Edition.Macmillan Publishing Company, New York, Oxford, Singapore, etc.,1993, 997p.

[10] Lemay L.,Cadenhead R., S^MS Teach Yourself Java 2 in 21 Days. A Division of Macmillian Computer Publishing, U.S.A, 1999,680p.

[11] Lewis, J. and Loftus, W., Java Software Solutions. Foundations Of Program Design. Addison Wesley Longman, Inc.,Massachussetts, Harlow, England, etc.,1998, 857p.

[12] Reilly,D.*"SimpleEvent HanDistance Educationing": http://www.javacoffeebreak.com*

[13] Weiss, M.I., Data Structures and Problem Solving Using Java. Addison Wesley Longman, Inc., U.S.A., 1997, 1038p.

[14] http://master.math.unibuc.ro/documentatie/books/javascript/1562764233/ch1.htm

[15] "*Brewing Java: A Tutorial": http://metalab.unc.edu/javafag/javatutorial.html*

[16] *http://www.cs.demson.edu/~cs428/resources/java/tutorial*

[17] *http://developer.irt.org/script/478.htm*

[18] *http://java.sun.com/docs/books/tutorial*