**CHAPTER 1**

**INTRODUCTION**

# **COMPUTER BASED TEST**

Computer based test (CBT) is alternate used across the globe by educational institutes and companies for conducting entrance test for their institute as it requires conducting the exam on a large scale and after the exam checking them and compiling the result could turn to extremely hectic task so this type of modern technology based alternative is made to come into play.

* + 1. Meaning and nature of CBT(Computer Based Test)

Computer based Test simply refers to tests and assessments conducted through the use of the organized systems on computers. Computer Based tests have the ability to automate a very time consuming task, marking and monitoring progress. Chalmers (2011) sees Computer Based test as a test that can be used in a supervised or non-supervised environment, and can allow students to check their own progress through self-assessment. It can also be used for testing lower-order skills (such as knowledge, understanding and application); it can also be used for testing higher-order skills to improve the students' analysis, synthesis and evaluation skills with more complex application software.

* + 1. Benefits of CBT

### Instant score report

Upon finishing the exam, candidates are immediately given a Score Report showing their exam results and passing status. Candidates ,after the exam is over,are provided with a breakdown of how they performed in all areas of the examination.

* Easy Instructions

A CBT application is made to be extremely user-friendly and responsive as well as containing attractive interface. The CBT format is a point and click format. Before taking the exam, one can go through instructions or tutorial on using a specific CBT application used for a particular examination before giving the final exam.

* Reduces or removes checker’s role

CBT application internally cross checks the candidate’s responses with the correct ones given to the application’s code beforehand. And in the produces the result alongside showing the performance of candidate in answering a particular question and the correct choice for the same if candidate had choosen the wrong choice.

* + 1. Project work(Brief description)

In this project work I coded to develop a small **offline computer based test desktop application** using the popular and versatile JAVA programming language.

* + - 1. *Reason for choosing this as project*

JAVA from the year of it’s development has been constantly used to write code for developing various desktop applications or software’s. A major reason has been it’s **WRITE ONCE REUSE/RUN ANYWHERE** feature which enables one to do the bank-end coding on any system and reuse the same code on any other platform/architecture very easily.

A computer based exam system proved to be a revolutionary step in conducting major exams like entrance exams for various universities and organisations/companies. Major reasons for this being the advantage provided to conduct exam on a **large scale**, ease in preparing the result and many more.

After facing the online examinations like JEE-MAIN, VITEEE, BITSAT conducted on a similar environment . I thought of always out of curiosity wanted to know the logic behind the exam, although it was an online exam build on a similar concept i.e. using JAVA language but as I learned JAVA CORE only in this summer training because course on ADVANCED JAVA was unavailable there. So using CORE JAVA I thought of developing a similar environment and was possible using the concepts of swings and developing a desktop application for the same.

Another reason was the important role of swings and GUI in JAVA explored by developers over the globe to build various desktop applications /software’s. Major example of it being the ANDROID STUDIO, various pdf readers and media players like FOXIT and VLC ,CLEMENTINE and many more(most of whom are developed by open community developers.)

* + - 1. *Various parts of the* ***quiz application***

The computer based exam developed named as **Quiz Application** is broadly divided into three parts on bases of the windows appearing during the course of it’s run, viz. welcome window, the main exam environment window, result window.

* + - * 1. Welcome window

A small popup appearing to ask for name of the candidate and if name is not provided it’s it be default which is defined in code as Anonymous.

* + - * 1. Exam environment

The consisting of main exam environment containing the instructions to be followed by the candidate during the exam. Various buttons at the bottom to control various operations like start, next , stop etc. for the candidate’s use. The most important part being the panel containing the questions and choices for the same and timer displaying the time left to complete the exam, after the candidate clicks the start button.

* + - * 1. Result window

This window displays the candidate’s result consisting of the marks obtained and whether candidate passed the exam or not. Also the performance in answering each question is displayed.

**CHAPTER 2**

**THE DESKTOP APPLICATION DEVELOPED**

At the end of the training we made a small working project. Following chapter briefs about various aspects and parts of the same.

* 1. **COMPONENTS OF DESKTOP APPLICATION**
     1. Welcome popup box

It is predefined function of JFrame class asking from from a input in the text field provided by it. The message could be defined in the code as following:

Line 85: studentname = JOptionPane.showInputDialog("Enter your name: ");

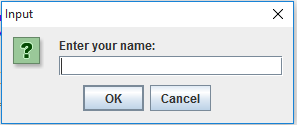


Figure 3(a):Welcome popup box asking for name of student

* + 1. Exam environment

It contains three windows north, south and central which are then added to main panel or window.

The central window is internally divided into two windows viz. east and west .

West window before clicking start button contains instructions for candidate and after clicking start button contain the questions to be answered shown in sequential order one at a time.



Figure 3(b):North window part after clicking start button

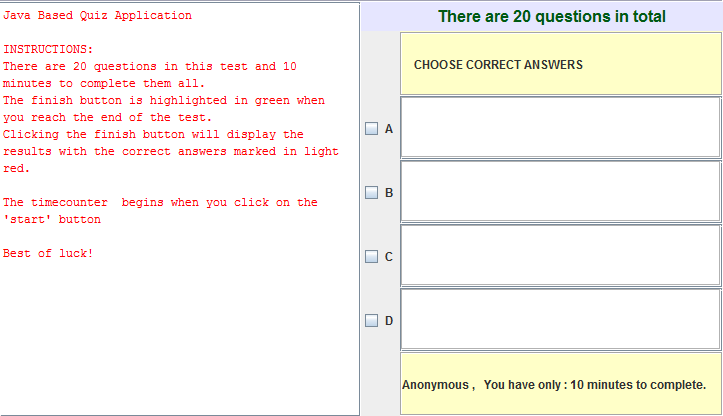


Figure 3(c):Central window before clicking start

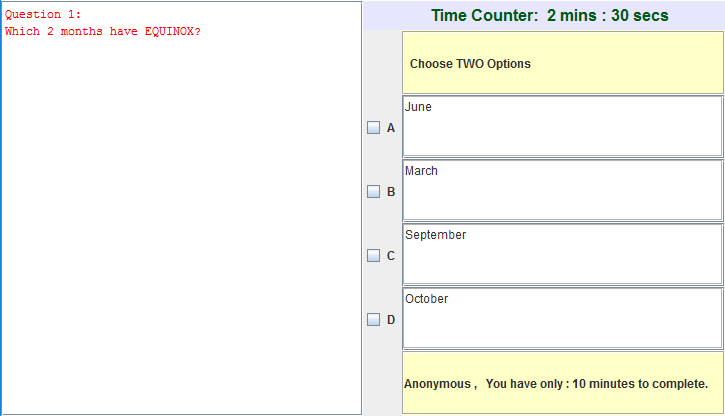


Figure 3(d):Central window after clicking start



Figure 3(e):South window before start

* + 1. Result window

This window displays the candidate’s performance in the test conducted.

Marks obtained and result of response of each question asked is displayed in the window. At the same time the correct option and the option the candidate chose and whether the same was correct one or not is showcased in the exam environment window.

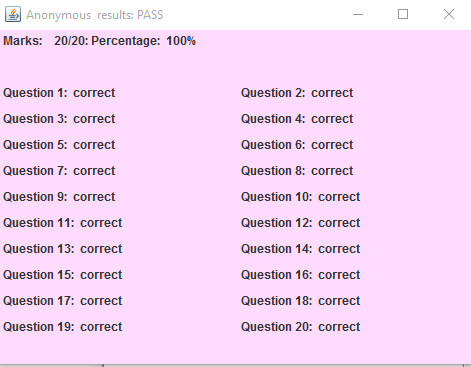


Figure 3(f).Result window

After discussing about the major sections displayed in the run of the application, the next section discusses the the code written for same, in brief.

* 1. **EXPLANATION OF CODE WRITTEN**

The code written for development of this application used various concepts of core Java like swing environment, exception handling, multithreading, data hiding and abstraction etc.

This section discusses the functionality of code by breaking down into smaller sections.

import java.awt. \*;

import java.awt. event.\*;

import javax.swing.\*;

Above lines import the required classes containg various functions, interfaces used in the code written.

class QuestionSeries

{static String info =

"Java Based Quiz Application \n \nINSTRUCTIONS:\nThere are 20 questions in this test "+ "and 10 minutes to complete them all.\nThe finish button is highlighted in green when you reach the end of the test. "+ "\nClicking the finish button will display the results with the correct answers marked in light red. "+ " \n \nThe timecounter begins when you click on the 'start' button \n \nBest of luck!\n";

static String []question =

{"Question 1:\nWhich 2 months have EQUINOX?",

"Question 2:\nCONDENSATION is the reverse process of?",

"Question 3: \nWhich country ranks second in terms of largest land area?",

"Question 4:\nWhich river carries maximum amount of water in the sea?",

"Question 5:\nWhere is the permanent secretariat of SAARC located?",

"Question 6:\nWhen is the WORLD ENVIRONMENT DAY celebrated?",

"Question 7:\nIn which year was UNO formed?",

"Question 8:\nWho is the author of Harry Potter?",

"Question 9:\nIn which ocean is Hawaii situated?",

"Question 10:\nHow many legs does a butterfly has?",

"Question 11:\nWhich is the most common element on earth?",

"Question 12:\nHow many tentacles does an octopus have?",

"Question 13:\nWhich planet has rings around it?",

"Question 14:\nWhat is the main use of Quinine?",

"Question 15:\nHow many is a Baker's dozen?",

"Question 16:\nWhich one of the following is not a primary colour?",

"Question 17:\nIn which country did Sherpa's originate?",

"Question 18:\nWhich is the largest gland in human body?",

"Question 19:\nWhich is the fastest mammal on earth?",

"Question 20:\nWhich sport is called the Sport of Kings?"

};

static String [][] answers =

{{"June\n","March\n","September\n","October\n"},

{"Melting\n"," \nEvaporation","Freezing\n"," \nSolidifying\n"},

{" \nChina\n","India\n","Canada\n","Russia\n"},

{" \nAmazon","Nile\n"," \nGanga\n"," \nThames\n"},

{" \nKathmandu\n"," \nWashington\n"," \nToronto\n"," \nLondon\n"},

{" \n5 july\n"," \n5 may\n"," \n4 may\n","5 june\n"},

{" \n1944\n"," \n1946\n"," \n1947\n"," \n1945\n"},

{" \nCharles Dickens\n"," \nEnid Blyton\n"," \nJ.K Rowling\n"," \nJane Austen\n"},

{" \nIndian\n"," \nArctic\n"," \nAntarctic\n"," \nPacific\n"},

{" \n8\n"," \n10\n"," \n6\n"," \n4\n"},

{" \nOxygen\n"," \nMagnesium\n"," \nSilicon\n"," \nHydrogen\n"},

{" \n8\n"," \n10\n"," \n12\n"," \n14\n"},

{" \nSaturn\n"," \nNeptune\n"," \nUranus\n"," \nJupiter\n"},

{"prevent fever\n"," \nprevent small pox\n"," \nprevent malaria\n"," \nprevent chicken pox\n"},

{" \n11\n"," \n12\n"," \n10\n"," \n13\n"},

{" \nred\n"," \nblue\n"," \nyellow\n"," \ngreen\n"},

{" \nchina\n"," \nnepal\n"," \nbhutan\n"," \nIndia\n"},

{" \nThyroid Gland\n"," \nKidney\n"," \nPituitary\n"," \nLiver\n"},

{" \nElephant\n"," \nTiger\n"," \nCheetah\n"," \nHumans\n"},

{" \nPolo\n"," \nHorse Racing\n"," \nCricket\n"," \nIce Hockey\n"},

{" \n50\n"," \n60\n"," \n75\n"," \n100\n"},

{"Argentina\n"," \nThailand\n"," \nJapan\n"," \nChina\n"},

{" Iron\n"," Zinc\n"," Lead\n"," Potassium\n"},

{" \n100\n"," \n1000\n"," \n10,000\n"," \n10\n"},

{"Brazil\n","Sri Lanka\n","Zambia\n"," \nIndia\n"} };

static int []n = {2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

static String []choice= {"23","2","3","1","1","4","4","3","4","3","4","1","1","3","4","4","2","4","3","2"};

static int tally = choice.length;

static String testtitle="Java Based Quiz Application";

static int timeLimit =10;

static int passMark = 33;

}

Class ‘QuestionSeries’ contain a string info which provides the candidate the required instructions regarding the exam. Another string ‘question’ contains the various questions asked to the candidate. String ‘answers’ contains the options available to the candidate from he/she has to choose most appropriate one as his/her choice. An array ‘n’ consists of the number of correct options for a particular question. String ‘choice’ contains the correct options for a question. Variable ‘tally’ of integer type stores the length of string ‘choice’. String testtitle sets the title being displayed in title bar of window. Integers timeLimit and passMark are set to 10(minutes) and 33 respectively.

public class Quiz extends JFrame{

static String studentname ="";

static int TOTAL=0;

static {

try{

TOTAL = QuestionSeries.tally;

studentname = JOptionPane.showInputDialog("Enter your name: ");

if(studentname.length() < 1) studentname = "Anonymous ";

else studentname = studentname.trim() + " ";

}

catch(NullPointerException e){ System.exit(0); }

}

int seconds, minutes;

int quesnum, itemCheck, mark;

final String TESTTITLE = QuestionSeries.testtitle;

final int TIMELIMIT = QuestionSeries.timeLimit;

final int PASS = QuestionSeries.passMark;

String []answers = new String[TOTAL];

boolean start\_test, check\_answer, allowRestart, finishtest;

JButton []choice\_button = new JButton[6];

JTextArea answerboxes[] = new JTextArea[4];

JCheckBox []boxes = new JCheckBox[4];

JTextPane pane = new JTextPane();

JLabel student, choose, message, timecounter, testresult;

Northwindow panelNorth = new Northwindow();

Southwindow panelSouth = new Southwindow();

Centerwindow panelCenter = new Centerwindow();

protected Quiz(){

for (int i=0; i<TOTAL; i++)

answers[i] ="";

getContentPane().setLayout(new BorderLayout() );

getContentPane().add("North", panelNorth);

getContentPane().add("South", panelSouth);

getContentPane().add("Center", panelCenter);

int width=740,height=540;

setSize(width,height);

Dimension dim = java.awt.Toolkit.getDefaultToolkit().getScreenSize();

setLocation((dim.width-width)/2, (dim.height-height)/2);

}

Class Quiz is the class which includes all the working code of the application. It extends the JFrame class for inheriting the methods and interfaces which are included in this class. Variables studentname and TOTAL are initialized as ‘null’ and ‘0’ respectively. Static block contains the try-catch set of statements which is used to manage the input received from user as the welcome dialog pops up. Other declarations of variables and various components are made, which would be used in the further course of the code. Three panels are declared, which at the end would be added to main frame of class quiz. Quiz constructor initializes all the answers as null and further set the layout as BorderLayout and three panels viz. North, South, Center are added in the window. Further enhancement like setting size of window for proper viewing of all components of the window and putting the window in middle of the screen of the pc are made.

class Northwindow extends JPanel{

public Northwindow(){

setLayout(new GridLayout(2,2));

setBackground(new Color(230, 230, 255));

student = new JLabel("\t Welcome :"+studentname+"to the quiz ");

student.setFont(new Font("",Font.BOLD,16) );

message = new JLabel();

message.setForeground(Color.BLUE);

add(student);

add(message);

add(new JLabel(" ") );

add(new JLabel(" ") );

setBorder(BorderFactory.createEtchedBorder() );

}

}

class Southwindow extends JPanel{

public Southwindow(){

String []key = {"","start:","next:","finish:","check next:","check previous:"};

for(int i=0; i<choice\_button.length; i++)

{choice\_button[i] = new JButton(key[i]);

choice\_button[i].addActionListener(new ActionHandler() );

if(i !=0)add(choice\_button[i]);

}

setBorder(BorderFactory.createEtchedBorder() );

}

}

Class Northwindow is created as a panel to be added to the main panel. Under the constructor of this class layout is set to grid creating four blocks for elements to place. Background is set, a label student is created another label in blue is declared which would indicate the question traversed by user. In the end all is added to the panel created and also border is created to separate out the panel from that below it.

Another class Southwindow created whose constructor contains array of string type with different names of buttons which are assigned to them (these buttons were created before the Quiz constructor) using a loop and also action listener is attached with these buttons. This panel is also surrounded by a border.

class Centerwindow extends JPanel{

public Centerwindow(){

setLayout(new GridLayout(1,2) );

JScrollPane west = new JScrollPane(pane);

pane.setForeground(Color.red);

pane.setFont(new Font ("monospaced",0,12) );

pane.setText(QuestionSeries.info);

pane.setEditable(false);

JPanel east = new JPanel();

east.setLayout(new BorderLayout() );

JPanel northEast = new JPanel();

northEast.setBackground(new Color(230, 230, 255) );

east.add("North", northEast);

JPanel westEast = new JPanel();

westEast.setLayout(new GridLayout(6,1) );

east.add("West", westEast);

JPanel centerEast = new JPanel();

centerEast.setLayout(new GridLayout(6,1) );

centerEast.setBackground(new Color(255,255,200));

east.add("Center", centerEast);

timecounter = new JLabel(" There are "+TOTAL+" questions in total");

timecounter.setFont(new Font ("Arial",Font.BOLD,16) );

timecounter.setForeground(new Color(0,90,20) );

northEast.add(timecounter);

westEast.add(new JLabel(" "));

String []boxs = {" A ", " B ", " C ", " D "};

for(int i=0; i<boxes.length; i++)

{

boxes[i] = new JCheckBox(boxs[i]);

boxes[i].addItemListener(new ItemHandler() );

westEast.add(boxes[i]);

}

westEast.add(new JLabel() );

choose = new JLabel(" CHOOSE CORRECT ANSWERS");

choose.setBorder(BorderFactory.createEtchedBorder() );

centerEast.add(choose);

JScrollPane panes[] = new JScrollPane[4];

for(int i=0; i<answerboxes.length; i++)

{

answerboxes[i] = new JTextArea();

answerboxes[i].setBorder(BorderFactory.createEtchedBorder() );

answerboxes[i].setEditable(false);

answerboxes[i].setBackground(Color.white);

answerboxes[i].setFont(new Font("",0,12) );

answerboxes[i].setLineWrap(true);

answerboxes[i].setWrapStyleWord(true);

panes[i] = new JScrollPane(answerboxes[i])

centerEast.add(panes[i]);

}

if(TIMELIMIT >0)

testresult = new JLabel(studentname+", You have only : "+TIMELIMIT+" minutes to complete.");

else

testresult = new JLabel(" There is no time limit for this test");

testresult.setBorder(BorderFactory.createEtchedBorder() );

centerEast.add(testresult);

add(west);

add(east);

}

}

Another class Centerwindow created as a sub-panel whose constructor contains the code for setting the gridlayout for the panel with one row and two columns. The instructions for candidate are showed up in west in red. In east of this panel questions will be shown alongwith choices among which candidate has to choose his/her response. Conditional flow statements monitors the time limit once the test has started.

Then the testresult and west & east windows are added to the Centerwindow.

class ActionHandler implements ActionListener{

public void actionPerformed(ActionEvent evt){

String source = evt.getActionCommand();

if(source.equals("start:"))

{

choice\_button[1].setVisible(false);

start\_test=true;

allowRestart=true;

if(TIMELIMIT >0)

new Timer();

displayquestion();

}

if(start\_test){

if(source.equals("previous:"))

{

recordanswer();

quesnum--;

if(quesnum == -1)

quesnum=TOTAL-1;

checkteststatus();

displayquestion();

}

if(source.equals("next:"))

{

recordanswer();

quesnum++;

if(quesnum == TOTAL-1)

finishtest=true;

if(quesnum == TOTAL)

quesnum=0;

checkteststatus();

displayquestion();

}

if(source.equals("finish:")) {

if (finishtest){

recordanswer();

quesnum = 0;

choice\_button[4].setBackground(Color.lightGray);

timecounter.setForeground(Color.blue);

timecounter.setFont(new Font ("Arial",0,14) );

start\_test=false;

check\_answer=true;

panelSouth.add(choice\_button[0]);

mark\_ques();

displayquestion();

checkteststatus();

calculateResult();

}

else

JOptionPane.showMessageDialog(null,"Cycle through all questions before pressing finish","User Message",JOptionPane.INFORMATION\_MESSAGE);

}

}

if (check\_answer)

{

if(source.equals("check next:"))

{

quesnum++;

if(quesnum == TOTAL)

quesnum=0;

mark\_ques();

displayquestion();

checkteststatus();

}

if(source.equals("check previous:"))

{

quesnum--;

if(quesnum == -1)

quesnum=TOTAL-1;

mark\_ques();

displayquestion();

checkteststatus();

}

}

}

ActionHandler class to handle all the action events from the buttons is created. In it actionPerformed method is used and parameter evt stored in variable source. If start pressed: make start button disappeared , start the test, allow restart, start the timer, make the questions visible. Now if start\_test variable is true then the variable (source) containing the info. of action performed is checked for being equal to be previous or next and the action is handled accordingly. When finish button is pressed and variable finishtest is made true then answers are recorded, checknext button’s background made grey, timer’s background made blue, and start\_test & check\_answer variables made false. And mark\_ques() , displayquestion() , checkteststatus() , calculateResult() constructors are called. And if user pressed finish button before cycling through all the questions a warning is displayed.

Then result is displayed and user will be able to cycle through the questions again and now he/she will be able to see the correct response and his/her own response.

class Timer extends Thread implements Runnable{

public Timer(){

new Thread(this).start();

}

public void run() {

while(start\_test){

try {

Thread.sleep(1000);

seconds++;

if(seconds % 60 == 0 && seconds != 0){

seconds -= 60;

minutes++;

}

timecounter.setText(" Time Counter: "+minutes+" mins : "+seconds+" secs ");

if(minutes==TIMELIMIT){

start\_test=false;

endTest();

}

}

catch(InterruptedException ex) { System.out.print(ex); }

}

}

}

In above code snippet thread is created using the runnable interface and sleep time is manipulated to count the seconds parsed during the period i.e. during the course of the test. In the end if timelimit exceeded then the test is automatically suspended by the system. This is done under the try, catch statements to avoid system crash.

public void checkteststatus(){

if((quesnum == TOTAL-1)&&(start\_test))

choice\_button[3].setBackground(Color.green);

else

choice\_button[4].setBackground(Color.lightGray);

if(answers[quesnum].length() >0)

{

for(int i=0; i<answers[quesnum].length(); i++)

boxes[Integer.parseInt(answers[quesnum].substring(i,i+1) )-1].setSelected(true);

}

else

for(int i=0; i<boxes.length; i++)

boxes[i].setSelected(false);

}

public void displayquestion(){

int j = quesnum+1;

pane.setText(QuestionSeries.question[quesnum]);

if(start\_test)

message.setText("Question "+j+" out of "+TOTAL);

for (int i=0; i<4; i++)

answerboxes[i].setText(QuestionSeries.answers[quesnum][i]);

if(start\_test){

String temp="";

if(QuestionSeries.n[quesnum]==1)

temp="<html>&nbsp;&nbsp;&nbsp;Choose only <b>ONE</b> Option</html>";

else if(QuestionSeries.n[quesnum]==2)

temp="<html>&nbsp;&nbsp;Choose <b>TWO </b> Options</html>";

else if(QuestionSeries.n[quesnum]==3)

temp="<html>&nbsp;&nbsp;Choose <b>THREE</b> Options</html>";

else

temp="<html>&nbsp;&nbsp;<b>ALL are true</b> true</html>";

choose.setText(temp);

}

else {

timecounter.setText(" Your choices are shown in the boxes");

choose.setText(" Correct answers are marked in light red.");

}

}

Conditions are checked for the commencement of test and questions not exceeding the range and initially all the answers to true .Now in the displayquestion method the questions are displayed and also the provision is there for the number of correct answers to the particular question. And after completion of test the same function is used to display the correct as well as user selected answers.

public void recordanswer(){

String tmp = "";

for(int i=0; i<boxes.length; i++)

if(boxes[i].isSelected() )

tmp +=i+1;

answers[quesnum] = tmp;

}

public void endTest(){

message.setText("TIME OVER: please press 'finish'");

choice\_button[2].setEnabled(false);

choice\_button[3].setEnabled(false);

choice\_button[4].setEnabled(true);

}

The record answer method is used to record the answers selected by the user. And endTest method displays that time is over and changes the display of buttons accordingly.

public void mark\_ques(){

for(int i=0; i<answerboxes.length; i++) answerboxes[i].setBackground(Color.white);

for(int i=0; i<QuestionSeries.choice[quesnum].length(); i++)

answerboxes[Integer.parseInt(QuestionSeries.choice[quesnum].substring(i,i+1))-1].setBackground(Color.red);

if(QuestionSeries.choice[quesnum].equals(answers[quesnum])) message.setText("Answer correct, well done!");

else message.setText("Sorry, you got this one wrong.");

}

public void calculateResult(){

mark=0;

double temp=0.0;

java.text.DecimalFormat df = new java.text.DecimalFormat("#0.#");

for(int i=0; i<TOTAL; i++)

if(QuestionSeries.choice[i].equals(answers[i]))mark++;

temp=(double)mark;

if(temp/TOTAL\*100 >=PASS)

testresult.setText(" Well done "+studentname.substring(0,studentname.indexOf(' ') )+", you passed");

else

testresult.setText(" Better luck next time "+studentname.substring(0,studentname.indexOf(' ') ) );

student.setText(" Final score for "+studentname+": "+mark+" out of "+TOTAL+": "+df.format(temp/TOTAL\*100)+"%");

new Resultwindow().show();

}

}

class Resultwindow extends JFrame{

Resultwindow() {

super( studentname+" results: " +(mark\*100/TOTAL >=PASS?"PASS":"FAIL") );

Container cont = getContentPane();

cont.setLayout(new GridLayout(TOTAL/2+3,5,2,5) );

cont.setBackground(new Color(255,220,255) );

cont.add(new JLabel(" "+"Marks: "+mark+"/"+TOTAL+": "+"Percentage: "+(mark\*100/TOTAL)+"%") );

for(int i=0; i<3; i++)cont.add(new JLabel() );

String temp[] = new String[TOTAL];

for(int i=0; i<TOTAL; i++){

if(QuestionSeries.choice[i].equals(answers[i]))

temp[i]="correct";

else

temp[i]="wrong";

}

for(int i=0; i<TOTAL; i++)

cont.add(new JLabel(" Question "+(i+1)+": "+temp [i]) );

pack();

setLocation(200,200);

}

}

The mark\_ques() method is used at the time of displaying the result of the user. Correct answers are highlighted in red color and if the selected answer is same as that as the correct option fed to program then display the message associated. Then calculateResult() calculates the result for the user. In succeeding class Resultwindow if student marks are more than 33 than display result as PASS otherwise FAIL and also display the marks and percentage obtained by candidate.

class ItemHandler implements ItemListener{

public void itemStateChanged(ItemEvent evt){

if(start\_test){

for(int i=0; i<boxes.length; i++)

if(boxes[i].isSelected() ) itemCheck++;

if(itemCheck > QuestionSeries.n[quesnum]){

java.awt.Toolkit.getDefaultToolkit().beep();

if(QuestionSeries.n[quesnum]==1)

JOptionPane.showMessageDialog(null,"<html><font size='4' color='00308a'><center>"+

"There is only "+QuestionSeries.n[quesnum]+" possible<br> answer to question "+(quesnum+1)+

"<html>","User Information Message",JOptionPane.INFORMATION\_MESSAGE);

else

JOptionPane.showMessageDialog(null,"<html><font size='4' color='00308a'><center>"+

"There are only "+QuestionSeries.n[quesnum]+" possible<br> answers to question "+(quesnum+1)+

"<html>","User Information Message",JOptionPane.INFORMATION\_MESSAGE);

}

itemCheck=0;

}

}

}

public static void main(String [] args){

//Object class Quiz created

Quiz frame = new Quiz();

frame.setTitle(" "+QuestionSeries.testtitle);

frame.setDefaultCloseOperation( EXIT\_ON\_CLOSE );

frame.setVisible(true);

}

}

ItemHandler class is used to handle the no. of correct answers to a question and if the test has started and a box is selected further check for the particular question with predifined no. of correct options. Here applet is used to display the message in a dialog box. In the last main method object of Quiz class is used to set the title, default close operation and pass true as argument in the setVisible method.

* 1. **CONCLUSION**

In this project work I was able to put together all the concepts I learned across the course of my training program on CORE JAVA and make a interactive and useful desktop application from it. During the development of the application I was able to understand and feel the amount of concentration, dedication and precision it demands from a developer to make even a small piece of application or software or game that we so easily use without bothering about the complexities of it’s operation. In the field of Computer Science new technology overtakes the older one within a matter of few days or months but the basic architecture and concept behind every technology is the thing we as a undergraduate are expected to understand and master through practice . So by developing small interactive applications in one of the most versatile and “machine independent” language in the world JAVA one puts the right step ahead on the road leading to a brighter future as a better developer who works towards the betterment of the society.