Source Code

//Final Java OnlineTest

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

//creates the questions frame.

//takes answers as input.

//provides controls for switching between questions

public class OnlineTest extends JFrame implements ActionListener

{

int qno=0,cor1=0,cor2=0,atm1=0,sc=0,max=20;

int vis[]=new int[max];

int ans[]=new int[max];

Timer timer;

static int secs=600;

int mins=secs/60;

JLabel l; //stores questions

JRadioButton jb[]=new JRadioButton[5];

JButton nxt,prv,sub,gobut,clropt;

ButtonGroup bg;

String[] msg= {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20"};

JComboBox cmb= new JComboBox(msg);

JLabel lbl = new JLabel("Select Question :");

JLabel atem = new JLabel("Attempted - "+atm1);

JLabel unatem = new JLabel("Unattempted - "+(max-atm1));

JLabel timlef = new JLabel("Time left - 10:0 mins");

OnlineTest() //constructs frame containing the Q&A's.

{

super("Quiz System");

l=new JLabel();

add(l);

bg=new ButtonGroup();

for(int i=0;i<5;i++)

{

jb[i]=new JRadioButton();

add(jb[i]);

bg.add(jb[i]);

}

nxt=new JButton("Next");

prv=new JButton("Previous");

sub=new JButton("Submit");

gobut= new JButton("Go to Question");

clropt= new JButton("Clear");

nxt.addActionListener(this);

prv.addActionListener(this);

sub.addActionListener(this);

gobut.addActionListener(this);

clropt.addActionListener(this);

timer = new Timer(1000, this);

add(nxt);add(prv);add(sub);

prv.setEnabled(false);

set();

cmb.setBounds(640,170,140,30);

lbl.setBounds(640,145,200,20);

atem.setBounds(20,20,200,20);

unatem.setBounds(220,20,200,20);

timlef.setBounds(620,20,200,20);

gobut.setBounds(640,210,140,30);

add(cmb);

add(gobut);

add(clropt);

add(lbl);

add(atem);

add(unatem);

add(timlef);

l.setBounds(20,50,750,20);

jb[0].setBounds(20,80,610,20);

jb[1].setBounds(20,110,610,20);

jb[2].setBounds(20,140,610,20);

jb[3].setBounds(20,170,610,20);

nxt.setBounds(20,210,100,30);

prv.setBounds(140,210,100,30);

clropt.setBounds(260,210,100,30);

sub.setBounds(380,210,110,30);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(600,400);

setVisible(true);

setSize(800,260);

timer.start();

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

{ans[i]=4;}

}

//assigns tasks to each of the buttons and timer

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==nxt) //next buttton

{

ans[qno]=getAns();

if(isAtempted()&&vis[qno]==0)

{atm1++;vis[qno]=1;}

else if(!isAtempted()&&vis[qno]==1)

{atm1--;vis[qno]=0;}

qno++;

set();

jb[ans[qno]].setSelected(true);

if(qno==max-1)

nxt.setEnabled(false);

else

prv.setEnabled(true);

}

if(e.getSource()==prv) //previous button

{

ans[qno]=getAns();

if(isAtempted()&&vis[qno]==0)

{atm1++;vis[qno]=1;}

else if(!isAtempted()&&vis[qno]==1)

{atm1--;vis[qno]=0;}

qno--;

set();

jb[ans[qno]].setSelected(true);

if(qno==0)

prv.setEnabled(false);

else

nxt.setEnabled(true);

}

if(e.getSource()==gobut) //go to question button

{

ans[qno]=getAns();

if(isAtempted()&&vis[qno]==0)

{atm1++;vis[qno]=1;}

else if(!isAtempted()&&vis[qno]==1)

{atm1--;vis[qno]=0;}

qno=Integer.parseInt(cmb.getSelectedItem().toString());

qno--;

set();

jb[ans[qno]].setSelected(true);

if(qno==0)

prv.setEnabled(false);

else

nxt.setEnabled(true);

if(qno==max-1)

nxt.setEnabled(false);

else

prv.setEnabled(true);

}

if(e.getSource()==clropt) //clear selected option

{

bg.clearSelection();

}

if(e.getSource()==sub) //submit.

{

timer.stop();

ans[qno]=getAns();

if(isAtempted()&&vis[qno]==0)

{atm1++;vis[qno]=1;}

else if(!isAtempted()&&vis[qno]==1)

{atm1--;vis[qno]=0;}

qno++;

getScore();

sc=cor1\*4-cor2;

{new endsequence(max,sc,cor1,atm1);}

//JOptionPane.showMessageDialog(this,"correct ans="+cor1);

//System.exit(0);

setVisible(false);

}

if(e.getSource()==timer) //timer

{

timlef.setText("Time left - "+(secs/60)+":"+(secs%60)+" mins");

secs--;

if (secs < 0)

{

if(secs==-5)

{timer.stop();sub.doClick(0);}

timlef.setText("Time up!");

}

}

}

public void set() //sets the label of Q&A's.

{

jb[4].setSelected(true);

if(qno==0)

{

l.setText("1: What is the return type of a method that does not returns any value? ");

jb[0].setText("int");

jb[1].setText("float");

jb[2].setText("void");

jb[3].setText("double");

}

if(qno==1)

{

l.setText("2: Which method can be defined only once in a program?");

jb[0].setText("main method");

jb[1].setText("finalize method");

jb[2].setText("static method");

jb[3].setText("private method");

}

if(qno==2)

{

l.setText("3: Which keyword is used by method to refer to the object that invoked it?");

jb[0].setText("import");

jb[1].setText("catch");

jb[2].setText("abstract");

jb[3].setText("this");

}

if(qno==3)

{

l.setText("4: Which operator's used by Java runtime implementations to free memory of an object when not needed?");

jb[0].setText("delete");

jb[1].setText("free");

jb[2].setText("new");

jb[3].setText("none of the above");

}

if(qno==4)

{

l.setText("5: Which of these method of class String is used to extract a single character from a String object?");

jb[0].setText("CHARAT()");

jb[1].setText("chatat()");

jb[2].setText("charAt()");

jb[3].setText("ChatAt()");

}

if(qno==5)

{

l.setText("6: Which of these constructors is used to create an empty String object?");

jb[0].setText("String()");

jb[1].setText("String(void)");

jb[2].setText("String(0)");

jb[3].setText("none of the above");

}

if(qno==6)

{

l.setText("7: What is the prototype of the default constructor of this class?\n" +

" public class prototype { } ");

jb[0].setText("prototype()");

jb[1].setText("prototype(void)");

jb[2].setText("public prototype(void)");

jb[3].setText("public prototype()");

}

if(qno==7)

{

l.setText("8: Find the error in the following code: byte b = 50; b = b \* 50;");

jb[0].setText("b can not contain value 100, limited by its range.");

jb[1].setText("\* has converted b \* 50 into int, which cannot be converted to byte without casting.");

jb[2].setText("b can not contain value 50.");

jb[3].setText("No error in this code");

}

if(qno==8)

{

l.setText("9: What is Truncation in Java?");

jb[0].setText("Floating-point value assigned to an integer type.");

jb[1].setText("Integer value assigned to floating type.");

jb[2].setText("Floating-point value assigned to an Floating type.");

jb[3].setText("Integer value assigned to floating type.");

}

if(qno==9)

{

l.setText("10: Which of the following statements are incorrect?");

jb[0].setText("String is a class");

jb[1].setText("String in java are mutable");

jb[2].setText("Every string is an object of class String");

jb[3].setText("A peer class of String called StringBuffer allows strings to be altered");

}

if(qno==10)

{

l.setText("11: What is Recursion in Java?");

jb[0].setText("Recursion is a class.");

jb[1].setText("A process of defining a method that calls other methods repeatedly.");

jb[2].setText("A process of defining a method that calls itself repeatedly.");

jb[3].setText("A process of defining a method that succesively calls other methods.");

}

if(qno==11)

{

l.setText("12:Which of these data types is used by operating system to manage the Recursion in Java?");

jb[0].setText("Array");

jb[1].setText("Stack");

jb[2].setText("Queue");

jb[3].setText("Tree");

}

if(qno==12)

{

l.setText("13: Which of these will happen if recursive method does not have a base case?");

jb[0].setText("An infinite loop occurs");

jb[1].setText("System stops the program after some time.");

jb[2].setText(" After 1000000 calls it will be automatically stopped.");

jb[3].setText("None of the mentioned");

}

if(qno==13)

{

l.setText("14: Which of these is not abstract?");

jb[0].setText("Thread");

jb[1].setText("Abstract list");

jb[2].setText("List");

jb[3].setText("None of the above");

}

if(qno==14)

{

l.setText("15: Which of the following statements are incorrect?");

jb[0].setText("A recursive method must have a base case.");

jb[1].setText("Recursion always uses stack.");

jb[2].setText("Recursion is faster than loops to call the function repeatedly using a stack.");

jb[3].setText("Recursion is managed by Java’s Run – Time environment.");

}

if(qno==15)

{

l.setText("16: Which of these keywords are used to define an abstract class?");

jb[0].setText("abst");

jb[1].setText("abstract");

jb[2].setText("Abstract");

jb[3].setText("abstract class");

}

if(qno==16)

{

l.setText("17: Which operator is used to generate an instance of an exception than can be thrown by using throw?");

jb[0].setText("new");

jb[1].setText("malloc");

jb[2].setText("alloc");

jb[3].setText("thrown");

}

if(qno==17)

{

l.setText("18: Which of these handles the exception when no catch is used?");

jb[0].setText("Default handler");

jb[1].setText("finally");

jb[2].setText("throw handler");

jb[3].setText("Java run time system");

}

if(qno==18)

{

l.setText("19: Which of these is an correct way of defining generic class?");

jb[0].setText("class name(T1, T2, …, Tn) { /\* … \*/ }");

jb[1].setText("class name { /\* … \*/ }");

jb[2].setText("class name[T1, T2, …, Tn] { /\* … \*/ }");

jb[3].setText("class name{T1, T2, …, Tn} { /\* … \*/ }");

}

if(qno==19)

{

l.setText("20: Which of these type parameters is used for a generic class to return and accept any type of object?");

jb[0].setText("K");

jb[1].setText("N");

jb[2].setText("T");

jb[3].setText("V");

}

atem.setText("Attempted - "+atm1);

unatem.setText("Unattempted - "+(max-atm1));

}

public boolean isAtempted()

{

if(qno>=0&&qno<max)

return(jb[0].isSelected()||jb[1].isSelected()||jb[2].isSelected()||jb[3].isSelected());

return false;

}

public int getAns()

{

if(jb[0].isSelected())

return 0;

if(jb[1].isSelected())

return 1;

if(jb[2].isSelected())

return 2;

if(jb[3].isSelected())

return 3;

return 4;

}

public int getScore()

{

System.out.println("Test Analysis");

System.out.println("Q.no.\tYour\tCorrect");

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

{

if(ans[i]==answerKey(i))

cor1++;

if(ans[i]!=4)

System.out.println((i+1)+"\t"+(ans[i]+1)+"\t"+(answerKey(i)+1));

else

System.out.println((i+1)+"\t-\t"+(answerKey(i)+1));

}

System.out.println("\n - Unattempted");

cor2=atm1-cor1;

return cor1;

}

public int answerKey(int qno)

{

if(qno==0)

return 2;

if(qno==1)

return 0;

if(qno==2)

return 3;

if(qno==3)

return 3;

if(qno==4)

return 2;

if(qno==5)

return 2;

if(qno==6)

return 3;

if(qno==7)

return 1;

if(qno==8)

return 0;

if(qno==9)

return 1;

if(qno==10)

return 1;

if(qno==11)

return 1;

if(qno==12)

return 0;

if(qno==13)

return 0;

if(qno==14)

return 3;

if(qno==15)

return 1;

if(qno==16)

return 0;

if(qno==17)

return 0;

if(qno==18)

return 1;

if(qno==19)

return 2;

return 9;

}

public static void main (String [] args)

{

new opensequence(secs/60);

}

}

class opensequence //instructions

{

public opensequence(int mins)

{

JFrame jfrm = new JFrame("Quiz System");

JButton start = new JButton("Start");

JLabel a =new JLabel(" EXAM \nINSTRUCTIONS ");

JLabel b= new JLabel("1.The duration of the exam is "+mins+" minutes.");

JLabel d= new JLabel("2.The exam will be automatically submitted after the time ends.");

JLabel e= new JLabel("3.The student can navigate to any question through the dropdown provided on the right.");

JLabel f= new JLabel("4.The result will be shown once you have submitted the exam.");

JLabel g= new JLabel("5.Time left is shown on the top right corner of the screen.");

JLabel h= new JLabel("(4 marks) for correct answer (-1 marks) for incorrect answer (0 marks) for unanswered ");

int x=80;

int y=60;

a.setBounds(320,y-40,300,20);

b.setBounds(x,y,700,20);

d.setBounds(x,y+20,700,20);

e.setBounds(x,y+40,700,20);

f.setBounds(x,y+60,700,20);

g.setBounds(x,y+80,700,20);

h.setBounds(x,y+100,700,20);

start.setBounds(350,y+140,100,30);

jfrm.add(start);

jfrm.add(a);

jfrm.add(b);

jfrm.add(d);

jfrm.add(e);

jfrm.add(f);

jfrm.add(g);

jfrm.add(h);

start.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

jfrm.setVisible(false);

new OnlineTest();

}

});

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jfrm.setLocation(600,400);

jfrm.setSize(800,260);

jfrm.setLayout(null);

jfrm.setVisible(true);

}

public static void main(String args[])

{new opensequence(0);}

}

class endsequence //results display

{

public endsequence(int max,int sc,int cor1,int atm1)

{

JFrame jfrm = new JFrame("Quiz System");

jfrm.setSize(800,260);

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel jlab = new JLabel(" You have successfully completed the exam.Please click on the button to display your results");

JLabel score = new JLabel("YOUR TOTAL SCORE : "+sc);

JLabel attempted = new JLabel("NUMBER OF ATTEMPTED QUESTIONS : "+atm1);

JLabel unattempted = new JLabel("NUMBER OF UNATTEMPTED QUESTIONS : "+(max-atm1));

JLabel crct = new JLabel("NUMBER OF CORRECT ANSWERS : "+cor1);

JLabel wrng = new JLabel("NUMBER OF WRONG ANSWERS : "+(atm1-cor1));

score.setBounds(230,50,300,20);

crct.setBounds(230,80,300,20);

wrng.setBounds(230,110,300,20);

attempted.setBounds(230,140,300,20);

unattempted.setBounds(230,170,300,20);

jfrm.add(jlab);

jlab.setBounds(70,50,700,30);

JButton jbtnBeta = new JButton("Result");

jbtnBeta.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

jbtnBeta.setVisible(false);

jlab.setVisible(false);

jfrm.add(score);

jfrm.add(crct);

jfrm.add(wrng);

jfrm.add(attempted);

jfrm.add(unattempted);

jfrm.remove(jlab);

jfrm.remove(jbtnBeta);

}

});

jbtnBeta.setBounds(350,170,100,30);

jfrm.add(jbtnBeta);

jfrm.setLocation(600,400);

jfrm.setLayout(null);

jfrm.setVisible(true);

}

public static void main(String args[])

{new endsequence(0,0,0,0);}

}