**20MCA132 OBJECT ORIENTED**

**PROGRAMMING LAB**

**MICRO-Project**

**MCQ**

SUBMITTED BY

VIVIN V. ABRAHAM

R MCA-B-2020-S2

ROLL NO : 42

**PROGRAM**

**File-1**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Quiz extends JFrame implements ActionListener

{

JButton b1,b2;

JTextField t1;

Quiz()

{

setBounds(400,200,1200,500);

getContentPane().setBackground(Color.WHITE);

setLayout(null);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/login.jpeg"));

JLabel l1=new JLabel(i1);

l1.setBounds(0,0,600,500);

add(l1);

JLabel l2=new JLabel("Time For Quiz");

l2.setFont(new Font("Script MT Bold",Font.BOLD,46));

l2.setForeground(new Color(30,144,104));

l2.setBounds(730,60,400,45);

add(l2);

JLabel l3=new JLabel("Enter Your Name");

l3.setFont(new Font("Mongolian Baiti",Font.BOLD,28));

l3.setForeground(Color.GREEN);

l3.setBounds(770,150,300,20);

add(l3);

t1 =new JTextField();

t1.setFont(new Font("Times New Roman",Font.BOLD,20));

t1.setBounds(735,200,300,25);

add(t1);

b1=new JButton("Rules");

b1.setBounds(735,270,120,25);

b1.setBackground(new Color(30,144,254));

b1.setForeground(Color.WHITE);

b1.addActionListener(this);

add(b1);

b2=new JButton("Exit");

b2.setBounds(915,270,120,25);

b2.setBackground(new Color(30,144,254));

b2.setForeground(Color.WHITE);

b2.addActionListener(this);

add(b2);

setVisible(true);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==b1){

String name =t1.getText();

this.setVisible(false);

new Rules(name);

}else{

System.exit(0);

}

}

public static void main(String[] args)

{

new Quiz();

}

}

**File-2**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Rules extends JFrame implements ActionListener{

JButton b1,b2;

String username;

Rules(String username){

this.username=username;

setBounds(600,200,800,650);

getContentPane().setBackground(Color.WHITE);

setLayout(null);

JLabel l1 = new JLabel("Welcome "+username+" to Quiz");

l1.setForeground(new Color(30,144,135));

l1.setFont(new Font("Algerian",Font.BOLD,34));

l1.setBounds(200,50,700,30);

add(l1);

JLabel l2=new JLabel("");

l2.setFont(new Font("Tahoma",Font.PLAIN,16));

l2.setBounds(80,90,600,350);

l2.setText( "<html>"+

"1. You are trained to be a programmer and not a story teller, answer point to point" + "<br><br>" +

"2. Do not unnecessarily smile at the person sitting next to you, they may also not know the answer" + "<br><br>" +

"3. You may have lot of options in life but here all the questions are compulsory" + "<br><br>" +

"4. Crying is allowed but please do so quietly." + "<br><br>" +

"5. Only a fool asks and a wise answers (Be wise, not otherwise)" + "<br><br>" +

"6. Do not get nervous if your friend is answering more questions " + "<br><br>" +

"<html>");

add(l2);

b1 =new JButton("Back");

b1.setBounds(250,430,100,30);

b1.setBackground(new Color(30,144,255));

b1.setForeground(Color.WHITE);

b1.addActionListener(this);

add(b1);

b2 =new JButton("Start");

b2.setBounds(400,430,100,30);

b2.setBackground(new Color(30,144,255));

b2.setForeground(Color.WHITE);

b2.addActionListener(this);

add(b2);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==b1){

this.setVisible(false);

new Quiz().setVisible(true);

}

else if(ae.getSource()==b2){

this.setVisible(false);

new TimeQuiz(username).setVisible(true);

}

}

public static void main(String[] args){

new Rules("");

}

}

**File-3**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class TimeQuiz extends JFrame implements ActionListener{

JButton next,lifeline,submit;

public static int count=0;

public static int timer=15;

public static int ans\_given=0;

public static int score=0;

JRadioButton opt1,opt2,opt3,opt4;

JLabel Q1,Qn1;

String q[][]=new String[10][5];

String pa[][]=new String[10][1];

String qa[][]=new String[10][2];

ButtonGroup options;

String username;

TimeQuiz(String username)

{

this.username=username;

setBounds(250,50,1440,950);

getContentPane().setBackground(Color.WHITE);

setLayout(null);

ImageIcon il=new ImageIcon(ClassLoader.getSystemResource("icons/quiz.jpg"));

JLabel l1=new JLabel(il);

l1.setBounds(0,0,1440,392);

add(l1);

Q1=new JLabel("");

Q1.setFont (new Font("Tahoma",Font.PLAIN,24));

Q1.setBounds(100,450,50,30);

add(Q1);

Qn1=new JLabel("");

Qn1.setFont (new Font("Tahoma",Font.PLAIN,24));

Qn1.setBounds(150,450,900,30);

add(Qn1);

q[0][0] = "Which is used to find and fix bugs in the Java programs.?";

q[0][1] = "JVM";

q[0][2] = "JDB";

q[0][3] = "JDK";

q[0][4] = "JRE";

q[1][0] = "What is the return type of the hashCode() method in the Object class?";

q[1][1] = "int";

q[1][2] = "Object";

q[1][3] = "long";

q[1][4] = "void";

q[2][0] = "Which package contains the Random class?";

q[2][1] = "java.util package";

q[2][2] = "java.lang package";

q[2][3] = "java.awt package";

q[2][4] = "java.io package";

q[3][0] = "An interface with no fields or methods is known as?";

q[3][1] = "Runnable Interface";

q[3][2] = "Abstract Interface";

q[3][3] = "Marker Interface";

q[3][4] = "CharSequence Interface";

q[4][0] = "In which memory a String is stored, when we create a string using new operator?";

q[4][1] = "Stack";

q[4][2] = "String memory";

q[4][3] = "Random storage space";

q[4][4] = "Heap memory";

q[5][0] = "Which of the following is a marker interface?";

q[5][1] = "Runnable interface";

q[5][2] = "Remote interface";

q[5][3] = "Readable interface";

q[5][4] = "Result interface";

q[6][0] = "Which keyword is used for accessing the features of a package?";

q[6][1] = "import";

q[6][2] = "package";

q[6][3] = "extends";

q[6][4] = "export";

q[7][0] = "In java, jar stands for?";

q[7][1] = "Java Archive Runner";

q[7][2] = "Java Archive";

q[7][3] = "Java Application Resource";

q[7][4] = "Java Application Runner";

q[8][0] = "Which of the following is a mutable class in java?";

q[8][1] = "java.lang.StringBuilder";

q[8][2] = "java.lang.Short";

q[8][3] = "java.lang.Byte";

q[8][4] = "java.lang.String";

q[9][0] = "Which of the following option leads to the portability and security of Java?";

q[9][1] = "Bytecode is executed by JVM";

q[9][2] = "The applet makes the Java code secure and portable";

q[9][3] = "Use of exception handling";

q[9][4] = "Dynamic binding between objects";

qa[0][1] = "JDB";

qa[1][1] = "int";

qa[2][1] = "java.util package";

qa[3][1] = "Marker Interface";

qa[4][1] = "Heap memory";

qa[5][1] = "Remote interface";

qa[6][1] = "import";

qa[7][1] = "Java Archive";

qa[8][1] = "java.lang.StringBuilder";

qa[9][1] = "Bytecode is executed by JVM";

opt1 =new JRadioButton("");

opt1.setBounds(170,520,700,30);

opt1.setFont(new Font("Dialog",Font.PLAIN,20));

opt1.setBackground(Color.WHITE);

add(opt1);

opt2 =new JRadioButton("");

opt2.setBounds(170,560,700,30);

opt2.setFont(new Font("Dialog",Font.PLAIN,20));

opt2.setBackground(Color.WHITE);

add(opt2);

opt3 =new JRadioButton("");

opt3.setBounds(170,600,700,30);

opt3.setFont(new Font("Dialog",Font.PLAIN,20));

opt3.setBackground(Color.WHITE);

add(opt3);

opt4 =new JRadioButton("");

opt4.setBounds(170,640,700,30);

opt4.setFont(new Font("Dialog",Font.PLAIN,20));

opt4.setBackground(Color.WHITE);

add(opt4);

options = new ButtonGroup();

options.add(opt1);

options.add(opt2);

options.add(opt3);

options.add(opt4);

next = new JButton("Next");

next.setFont(new Font("Tahoma",Font.PLAIN,22));

next.setBackground(new Color(30,144,255));

next.setForeground(Color.WHITE);

next.setBounds(1100,550,200,40);

next.addActionListener(this);

add(next);

lifeline = new JButton("50-50 Lifeline");

lifeline.setFont(new Font("Tahoma",Font.PLAIN,22));

lifeline.setBackground(new Color(30,144,255));

lifeline.setForeground(Color.WHITE);

lifeline.setBounds(1100,630,200,40);

lifeline.addActionListener(this);

add(lifeline);

submit = new JButton("Submit");

submit.setFont(new Font("Tahoma",Font.PLAIN,22));

submit.setEnabled(false);

submit.setBackground(new Color(30,144,255));

submit.setForeground(Color.WHITE);

submit.setBounds(1100,710,200,40);

submit.addActionListener(this);

add(submit);

start(0);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==next){

repaint();

opt1.setEnabled(true);

opt2.setEnabled(true);

opt3.setEnabled(true);

opt4.setEnabled(true);

ans\_given=1;

if(options.getSelection()==null){

pa[count][0]="";

}else{

pa[count][0]=options.getSelection().getActionCommand();

}

if(count==8){

next.setEnabled(false);

submit.setEnabled(true);

}

count++;

start(count);

}else if(ae.getSource()==submit){

ans\_given=1;

if(options.getSelection()==null){

pa[count][0]="";

}else{

pa[count][0]=options.getSelection().getActionCommand();

}

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

if(pa[i][0].equals(qa[i][1])){

score+=10;

}else{

score+=0;

}

}

this.setVisible(false);

new Score(username,score).setVisible(true);

}else if(ae.getSource()==lifeline){

if(count==1||count==4|| count==6||count==8||count==9)

{

opt2.setEnabled(false);

opt3.setEnabled(false);

}

else

{

opt1.setEnabled(false);

opt4.setEnabled(false);

}

lifeline.setEnabled(false);

}

}

public void paint(Graphics g)

{

super.paint(g);

String time ="Time Left - "+timer+ " seconds";

g.setColor(Color.RED);

g.setFont(new Font("Tahoma",Font.BOLD,25));

if(timer>0){

g.drawString(time,1100,500);

}else{

g.drawString("Times Up!!",1100,500);

}

timer--;

try{

Thread.sleep(1000);

repaint();

}catch(Exception e){

e.printStackTrace();

}

if(ans\_given==1){

ans\_given=0;

timer=15;

}else if(timer<0){

timer=15;

opt1.setEnabled(true);

opt2.setEnabled(true);

opt3.setEnabled(true);

opt4.setEnabled(true);

if(count==8)

{

next.setEnabled(false);

submit.setEnabled(true);

}

if(count==9){

if(options.getSelection()==null){

pa[count][0]="";

}else{

pa[count][0]=options.getSelection().getActionCommand();

}

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

if(pa[i][0].equals(qa[i][1])){

score+=10;

}else{

score+=0;

}

}

this.setVisible(false);

new Score(username,score).setVisible(true);

}else{

if(options.getSelection()==null){

pa[count][0]="";

}else{

pa[count][0]=options.getSelection().getActionCommand();

}

count++;

start(count);

}

}

}

public void start(int count){

Q1.setText(""+(count+1) +". ");

Qn1.setText(q[count][0]);

opt1.setLabel(q[count][1]);

opt1.setActionCommand(q[count][1]);

opt2.setLabel(q[count][2]);

opt2.setActionCommand(q[count][2]);

opt3.setLabel(q[count][3]);

opt3.setActionCommand(q[count][3]);

opt4.setLabel(q[count][4]);

opt4.setActionCommand(q[count][4]);

options.clearSelection();

}

public static void main(String[] args)

{

new TimeQuiz("").setVisible(true);

}

}

**File-4**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Score extends JFrame implements ActionListener

{

Score(String username,int score)

{

setBounds(600,150,750,550);

getContentPane().setBackground(Color.WHITE);

setLayout(null);

ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("icons/score.png"));

Image i2=i1.getImage().getScaledInstance(300,250, Image.SCALE\_DEFAULT);

ImageIcon i3=new ImageIcon(i2);

JLabel l1 =new JLabel(i3);

l1.setBounds(0,200,300,250);

add(l1);

JLabel l2 = new JLabel("Thank you "+username+" for playing MCQ");

l2.setBounds(180,60,700,60);

l2.setForeground(Color.red);

l2.setFont(new Font("Script MT Bold",Font.PLAIN,36));

add(l2);

JLabel l3 = new JLabel("Your Score : " +score);

l3.setBounds(350,200,300,30);

l3.setFont(new Font("Jokerman",Font.PLAIN,26));

l3.setForeground(new Color(199,21,133));

add(l3);

JButton b1=new JButton("Exit");

b1.setBackground(Color.BLUE);

b1.setFont(new Font("Times New Roman",Font.BOLD,24));

b1.setForeground(Color.WHITE);

b1.addActionListener(this);

b1.setBounds(400,270,120,30);

add(b1);

}

public void actionPerformed(ActionEvent ae){

this.setVisible(false);

System.exit(0);

}

public static void main(String[] args)

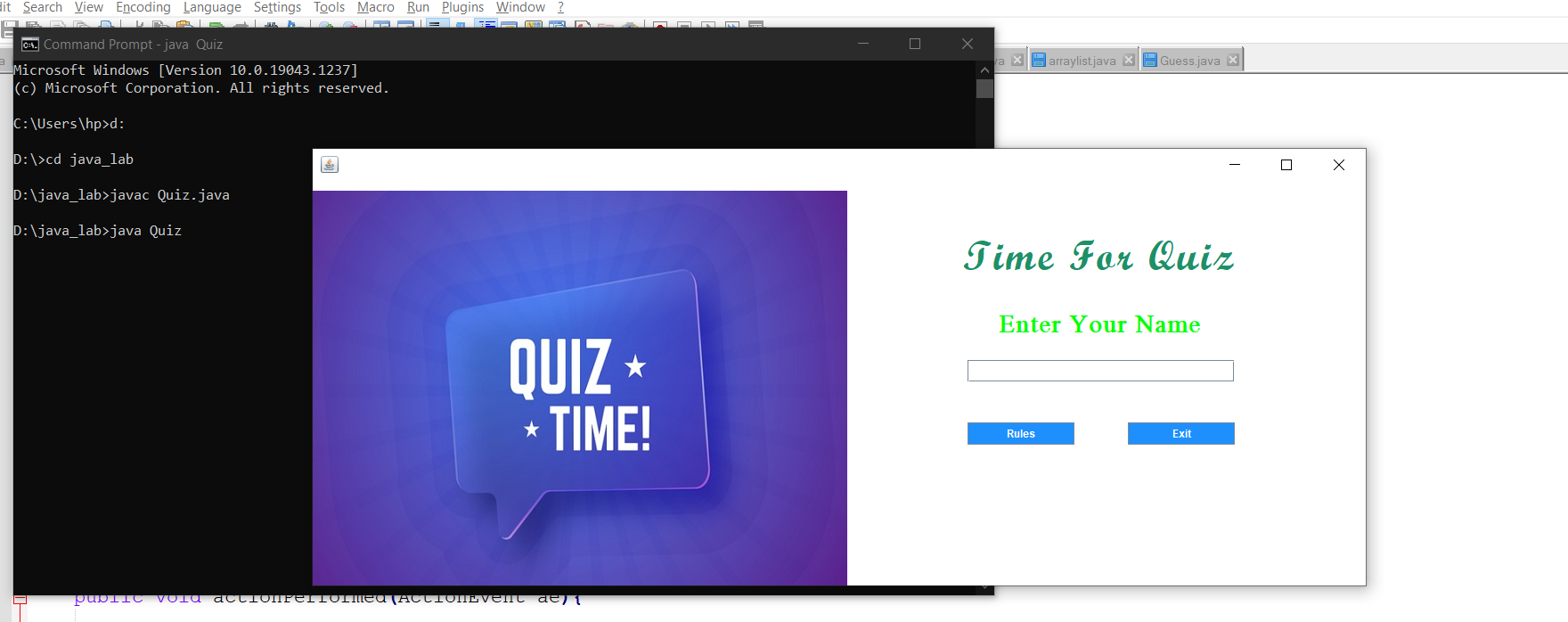
{

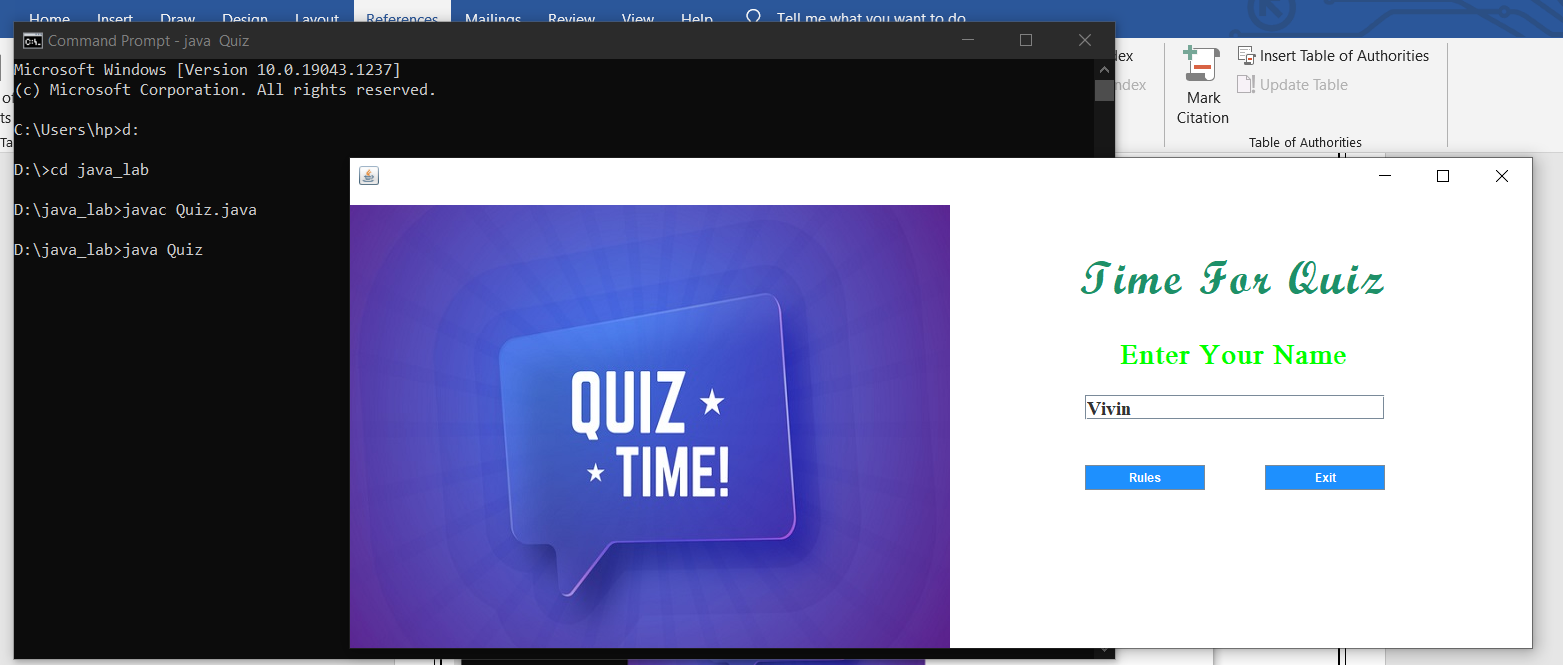
new Score("",0).setVisible(true);

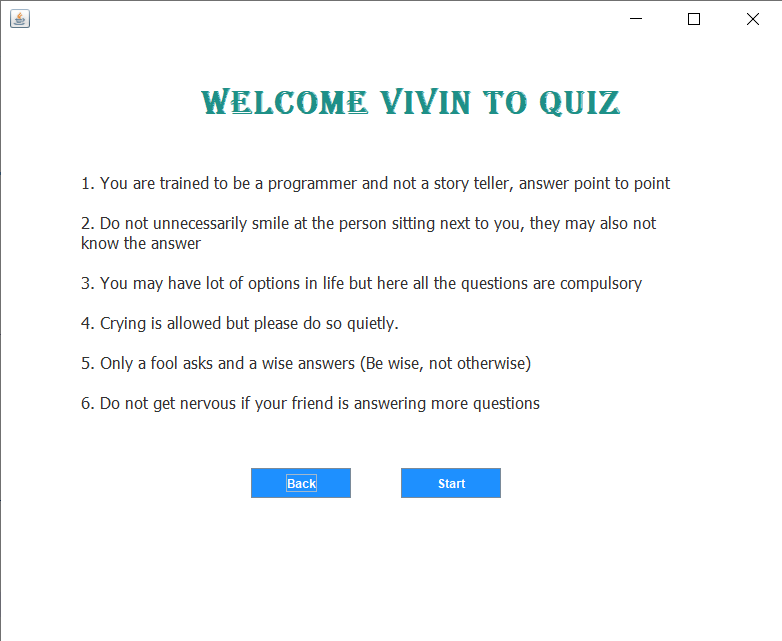
}

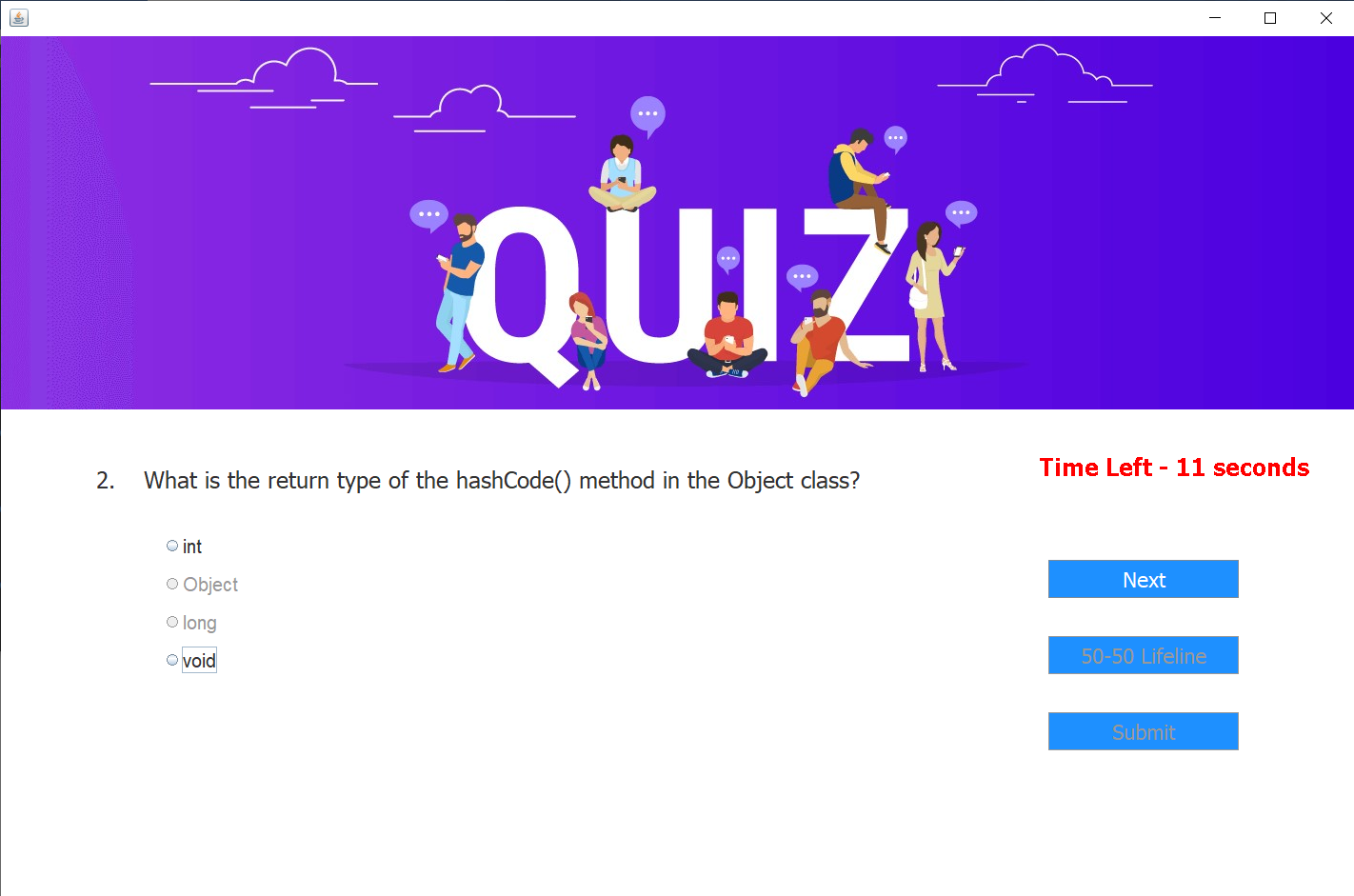
}

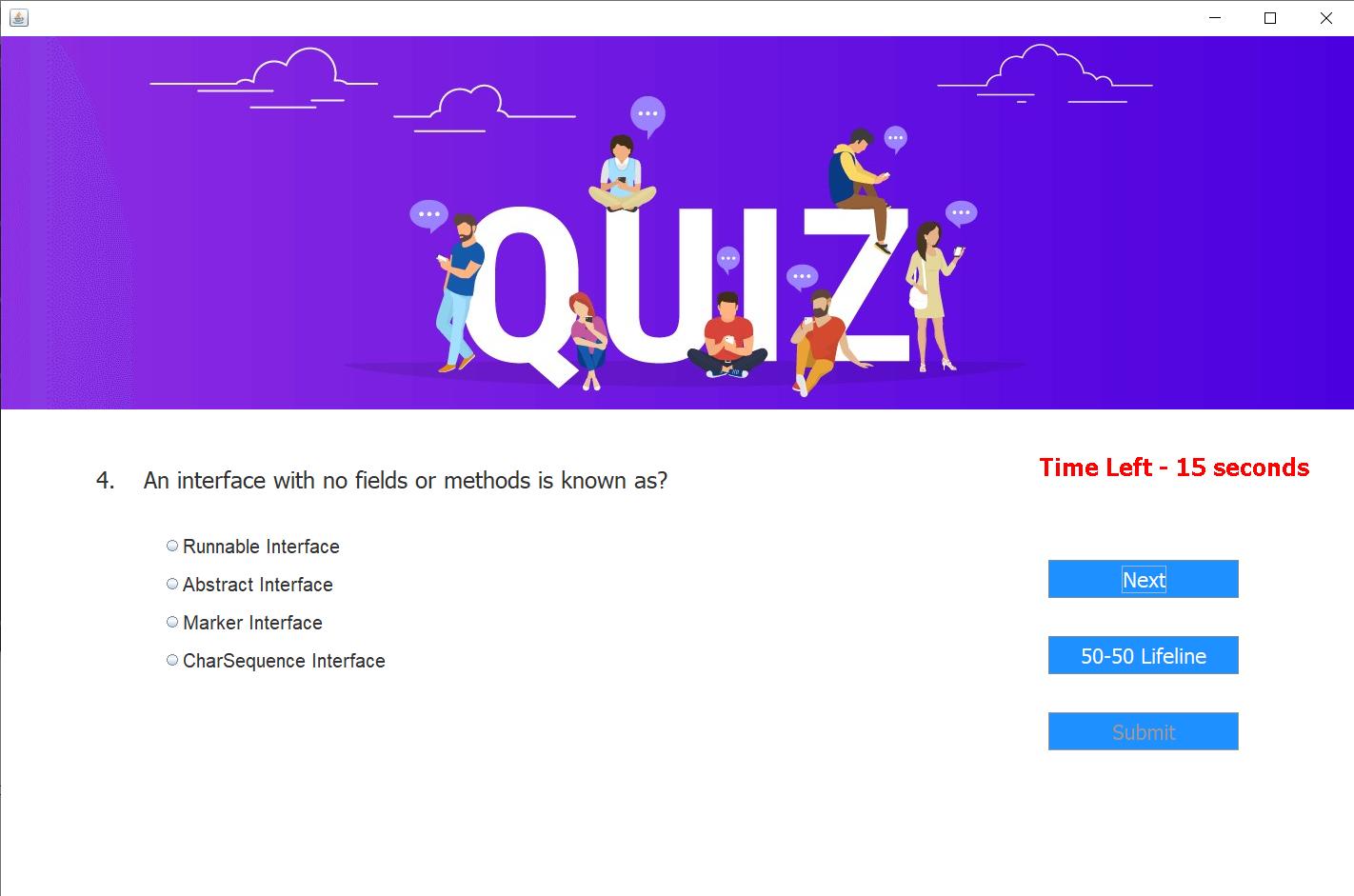
**OUTPUT**

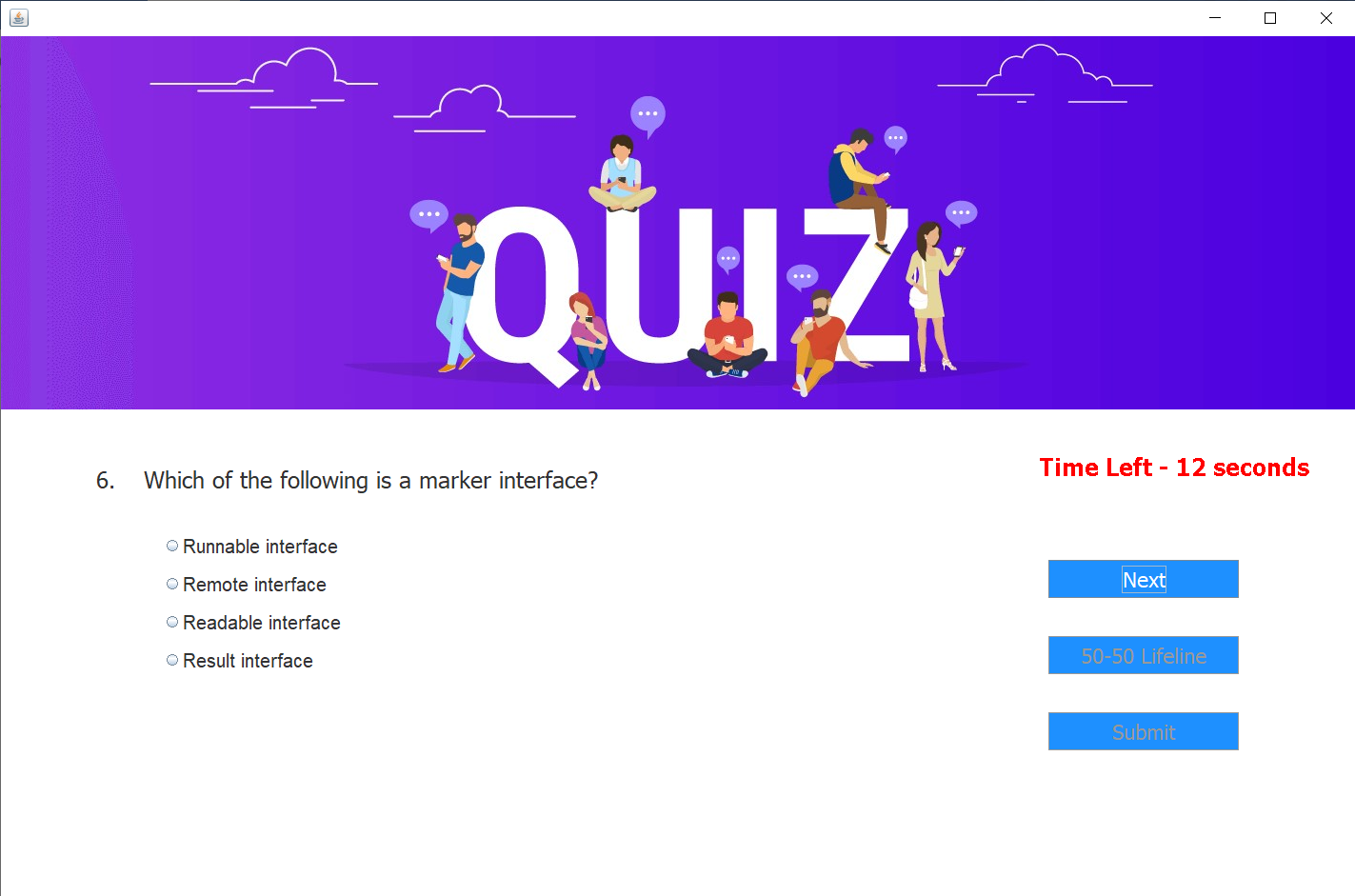
****

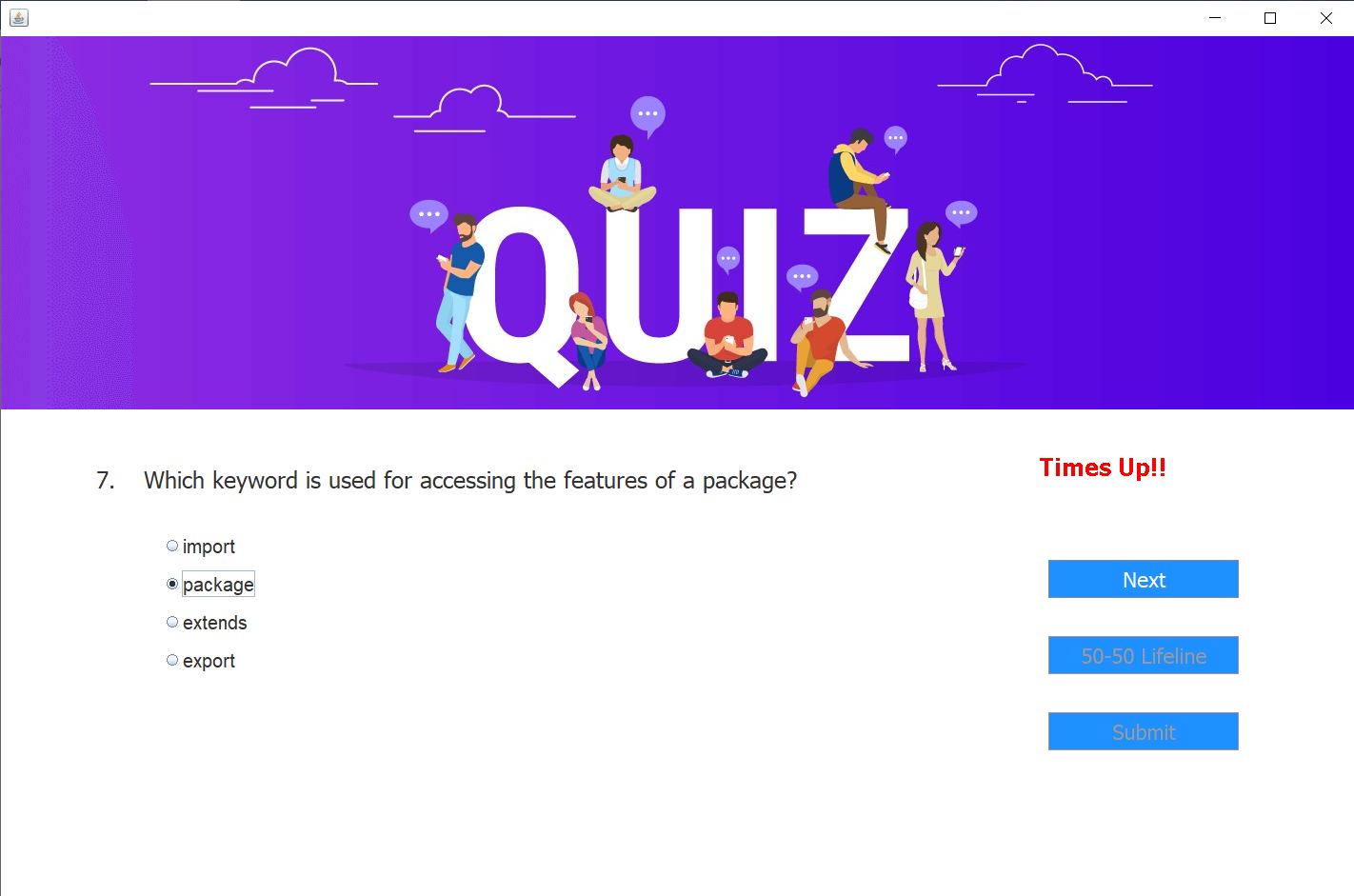
****

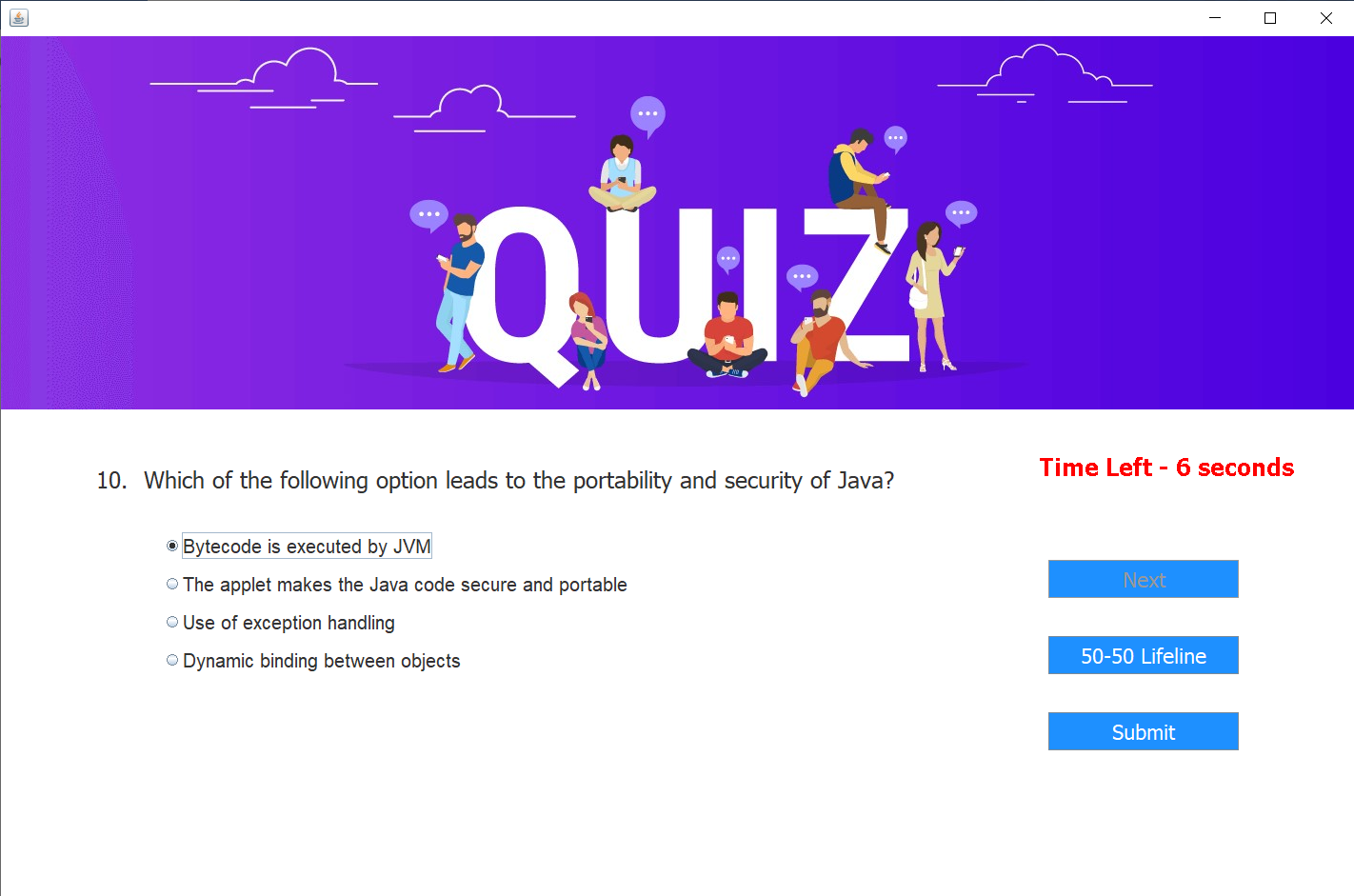
****

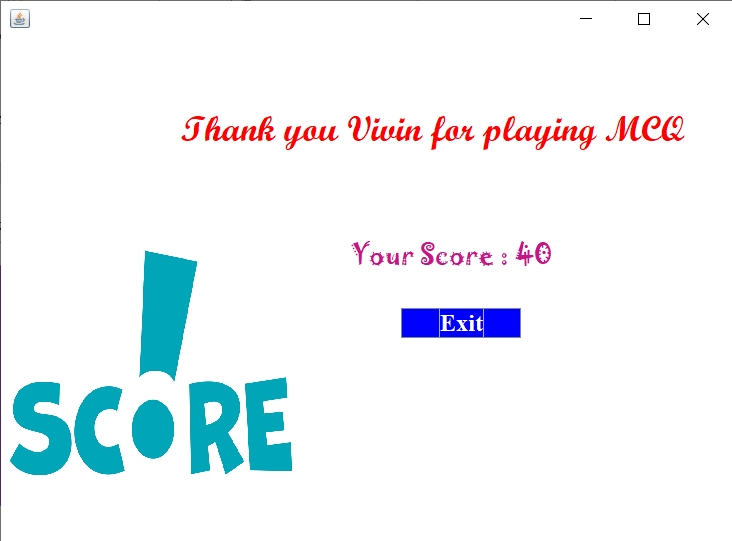
****

****

****

****

****

****