PROJECT
TOPIC: ONLINE TESTING SYSTEM
OBJET ORIENTED PROGRAMMING (OOP)

SUBMITTED TO:

MS IQRA SHAHZAD

SUBMITTED BY:

MUHAMMAD AMAAN IKRAM
ABDULLAH SHAH
KHADIJA NOOR
MUHAMMAD RAFEH

BSSE 2B-MORNING

ONLINE TESTING SYSTEM

DISCRIPTION:

The given code represents an **Online Test application** using Java Swing for the graphical user interface (**GUI**). Let's go through the code and understand its working.

The code starts by importing the necessary packages, including javax.swing and java.awt, which provide classes for creating GUI components.

The **Quiz** class extends **JFrame** and implements the **ActionListener** interface. It represents the main application window and handles the user's actions.

The class contains several instance variables, including labels, frames (f1 and f2), text fields, radio buttons, and buttons for the test application.

The constructor **Quiz(String s)** initializes the GUI components, sets their properties, and adds them to the frames. It creates two frames: frame for the login page, f1 for the instruction page, and f2 for the test questions.

The **actionPerformed** method handles various events triggered by buttons. Here's an overview of each event:

Login Button Action:
It checks the entered username and password and validates them.
If the login is successful, it hides the login frame (frame) and displays the instruction frame (f1).
If the login fails, it shows an error message using JOptionPane.

"Start Test" Button Action:
It hides the instruction frame (f1) and displays the test frame (f2).

"Next" Button Action:
It checks the selected answer and updates the count if it is correct.

It increments the current variable to move to the next question.
If it is the last question, it disables the "Next" button and changes the text of the "Bookmark" button to "Result."

"Bookmark" Button Action:
It adds a bookmark button dynamically for a particular question and increments the x variable.
It stores the current question number in the m array.
It increments the current variable to move to the next question.
If it is the last question, it changes the text of the "Bookmark" button to "Result."
Bookmark Button (Dynamic) Action:

It restores the bookmarked question and sets the current question to the bookmarked question number.
It disables the bookmark button after restoration.

"Result" Button Action:
It checks the selected answer for the current question and updates the count if it is correct.
It calculates the percentage and displays a message box with the result.
It exits the application.
The set() method sets the question and options based on the current question number (current) and updates the label and radio buttons accordingly.

The check() method checks if the selected answer for the current question is correct and returns a boolean value.

In the main method, an instance of the Quiz class is created to start the application.

Overall, the code creates a login page, an instruction page, and a test page with multiple-choice questions. It allows the user to log in, read the instructions, attempt the test, bookmark questions, and view the result at the end.

CODE

```
package com.mycompany.quiz;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Quiz extends JFrame implements ActionListener {
        JLabel label;
        JFrame f2,f1,frame;
        JTextField usernameField;
        JPasswordField passwordField;
          JRadioButton radioButton[] = new JRadioButton[5];
          JButton btnNext, btnBookmark,loginButton,b4;
          ButtonGroup bg;
          int count = 0, current = 0, x = 1, y=1,now = 0;
          int m[] = new int[10];

          Quiz(String s) {
            //LOG IN PAGE
            frame = new JFrame(s);

            JLabel usernameLabel = new JLabel("Username:");
            JLabel passwordLabel = new JLabel("Password:");
            JLabel login = new JLabel("LOG IN");
            usernameField = new JTextField();
            passwordField = new JPasswordField();
            //button login
            loginButton = new JButton("Login");

            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(600, 350);
            frame.setLayout(null);
            frame.setLocation(250,100);
```

```
 login.setFont(new Font("times new roman", Font.BOLD, 28));
  login.setForeground(Color.black);
  login.setBounds(400, 60, 300, 20);

 loginButton.setBounds(450, 180, 100, 30);
 loginButton.setBackground(Color.orange);
 loginButton.setForeground(Color.black);

 usernameLabel.setBounds(280, 100, 200, 20);
 passwordLabel.setBounds(280, 130, 200, 20);
usernameField.setBounds(350, 100, 200, 20);
 passwordField.setBounds(350, 130, 200, 20);

  ImageIcon bg1 = new ImageIcon("src\\img.jpg");
 JLabel l3 = new JLabel(bg1);
 l3.setBounds(0, 0, 600, 350);

 frame.setVisible(true);

 frame.add(usernameLabel);
 frame.add(usernameField);
 frame.add(passwordLabel);
 frame.add(passwordField);
 frame.add(loginButton);
 frame.add(login);
frame.add(l3);


 //instruction page and rest
 //

 f1= new JFrame(s);
 f2=new JFrame(s);
```

```
        JTextArea l1 = new JTextArea ("\n\tInstructions: \n \t> This is a free online test. DO NOT pay \n \t> Total number of question : 10 \n \t> Time Allowed : 30 minutes \n \t> Each question carry 1 mark \n \t> Atleast 50% marks to pass this test \n \t> Best of luck " );
            l1.setBounds(0,0,600,350);
             l1.setOpaque(false);
           l1.setEditable(false);



             b4 = new JButton ("Start Test");
            b4.setBounds(100,240,100,30);

            b4.setBackground(Color.orange);
            b4.setForeground(Color.black);
            b4.setBounds(200,240,100,30);
             b4.addActionListener(this);

               label = new JLabel();
                    f2.add(label);
                    bg = new ButtonGroup();
                    for (int i = 0; i < 5; i++) {
                              radioButton[i] = new JRadioButton();
                       radioButton[i].setOpaque(false);
                       radioButton[i].setBackground(null);

                              f2.add(radioButton[i]);
                              bg.add(radioButton[i]);
                    }
                    btnNext = new JButton("Next");
                btnNext.setBackground(Color.orange);
                btnNext.setForeground(Color.black);
                    btnBookmark = new JButton("Bookmark");
               btnBookmark.setBackground(Color.orange);
               btnBookmark.setForeground(Color.black);

                    btnNext.addActionListener(this);
```

```
     btnBookmark.addActionListener(this);
 loginButton.addActionListener(this);


     f2.add(btnNext);
     f2.add(btnBookmark);
     set();
     label.setBounds(30, 40, 450, 20);

     radioButton[0].setBounds(50, 80, 200, 20);
     radioButton[1].setBounds(50, 110, 200, 20);
     radioButton[2].setBounds(50, 140, 200, 20);
     radioButton[3].setBounds(50, 170, 200, 20);


     btnNext.setBounds(100, 240, 100, 30);
     btnBookmark.setBounds(270, 240, 100, 30);

   ImageIcon bg2 = new ImageIcon("src\\img2.jpg");
  JLabel l4 = new JLabel(bg2);
  l4.setBounds(0, 0, 600, 350);

     f2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     f2.setLayout(null);
     f2.setLocation(250, 100);
     f2.setSize(600, 350);
 f2.add(l4);
 f2.setVisible(false);


ImageIcon bg3 = new ImageIcon("src\\img2.jpg");
  JLabel l5 = new JLabel(bg3);
  l5.setBounds(0, 0, 600, 350);f1.setVisible(false);
 f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 f1.setLocation(250,100);
 f1.setSize(600,350);
 f1.add(b4);
```

```
                f1.add(l1);

                 f1.add(l5);
                 f1.setVisible(false);

        }

        public void actionPerformed(ActionEvent e) {


     // Handle login button action
   // Handle login button action
if (e.getSource() == loginButton) {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());

    // TODO: Perform login validation here
    if (username.equals("admin") && password.equals("password")) {
        // Login successful, open the instruction page
        frame.setVisible(false);
        f1.setVisible(true);
    } else {
        // Invalid login, display error message
        JOptionPane.showMessageDialog(this, "Invalid username or password",
"Login Error", JOptionPane.ERROR_MESSAGE);
    }
}



if(e.getSource()==b4){
    f1.setVisible(false);
    f2.setVisible(true);
}

    // Handle "Next" button action
    if (e.getSource() == btnNext) {
        if (check())
        count = count + 1;
    current++;
    set();
    if (current == 9) {
        btnNext.setEnabled(false);
        btnBookmark.setText("Result");
    }
}

// Handle "Bookmark" button actions

if (e.getActionCommand().equals("Bookmark")) {
                          JButton bk = new JButton("Bookmark" + x);
                          bk.setBounds(480, 20 + 30 * x, 100, 30);
                          f2.add(bk);
                          bk.addActionListener(this);
                    bk.setBackground(Color.orange);
                    bk.setForeground(Color.black);
                          m[x] = current;
                          x++;
                          current++;
                          set();
                          if (current == 9)
                                    btnBookmark.setText("Result");

                }
                for (int i = 0, y = 1; i < x; i++, y++) {
                          if (e.getActionCommand().equals("Bookmark" + y)) {
                                    if (check())
                                              count = count + 1;
                                    now = current;
                                    current = m[y];
                                    set();
                                    ((JButton) e.getSource()).setEnabled(false);
                                    current = now;
                          }
                }
    // Handle "Result" button action
    if (e.getActionCommand().equals("Result")) {
        if (check())
            count = count + 1;
        current++;
        int percent = (count * 100) / 10;
        if (percent >= 50) {
            JOptionPane.showMessageDialog(this, "Correct answers: " + count +
"\nWrong answers: " + (10 - count) + "\nPercentage: " + percent + "%\nPass");
        } else {
            JOptionPane.showMessageDialog(this, "Correct answers: " + count +
"\nWrong answers: " + (10 - count) + "\nPercentage: " + percent + "%\nFail");
        }
        System.exit(0);
    }
}
```

```
     // SET Questions with options
          void set() {
                    radioButton[4].setSelected(true);
                    if (current == 0) {
                              label.setText("Que1: Which concepts is NOT associated with
Object-Oriented Programming?");
                              radioButton[0].setText("inheritance");
                              radioButton[1].setText("abstraction");
                              radioButton[2].setText("polymorphism");
                              radioButton[3].setText("sequential programing");
                    }
                    if (current == 1) {
                              label.setText("Que2: What is the process of creating an object
called in Java?");
                              radioButton[0].setText("instanciation");
                              radioButton[1].setText("inheritance");
                              radioButton[2].setText("polymorphism");
```

```
            radioButton[3].setText("abstraction");
        }
        if (current == 2) {
            label.setText("Que3: Which keyword is used to inherit a class
in Java?");
            radioButton[0].setText("super class");
            radioButton[1].setText("super");
            radioButton[2].setText("inherit");
            radioButton[3].setText("extend");
        }
        if (current == 3) {
            label.setText("Que4: What is access level of class member can
be accessed from class?");
            radioButton[0].setText("public");
            radioButton[1].setText("default");
            radioButton[2].setText("protected");
            radioButton[3].setText("private");
        }
        if (current == 4) {
            label.setText("Que5: Which OOP principle allows an object to
exhibit multiple forms?");
            radioButton[0].setText("encapsulation");
            radioButton[1].setText("polymorphism");
            radioButton[2].setText("inheritance");
            radioButton[3].setText("abstraction");
        }
        if (current == 5) {
            label.setText("Que6: How to read entire file in one line using
java 8?");
            radioButton[0].setText("Files.readAllLines()");
            radioButton[1].setText("Files.read()");
            radioButton[2].setText("Files.readFile()");
            radioButton[3].setText("Files.lines()");
        }
        if (current == 6) {
            label.setText("Que7:  What is the process of wrapping data and
methods into a single unit?");
```

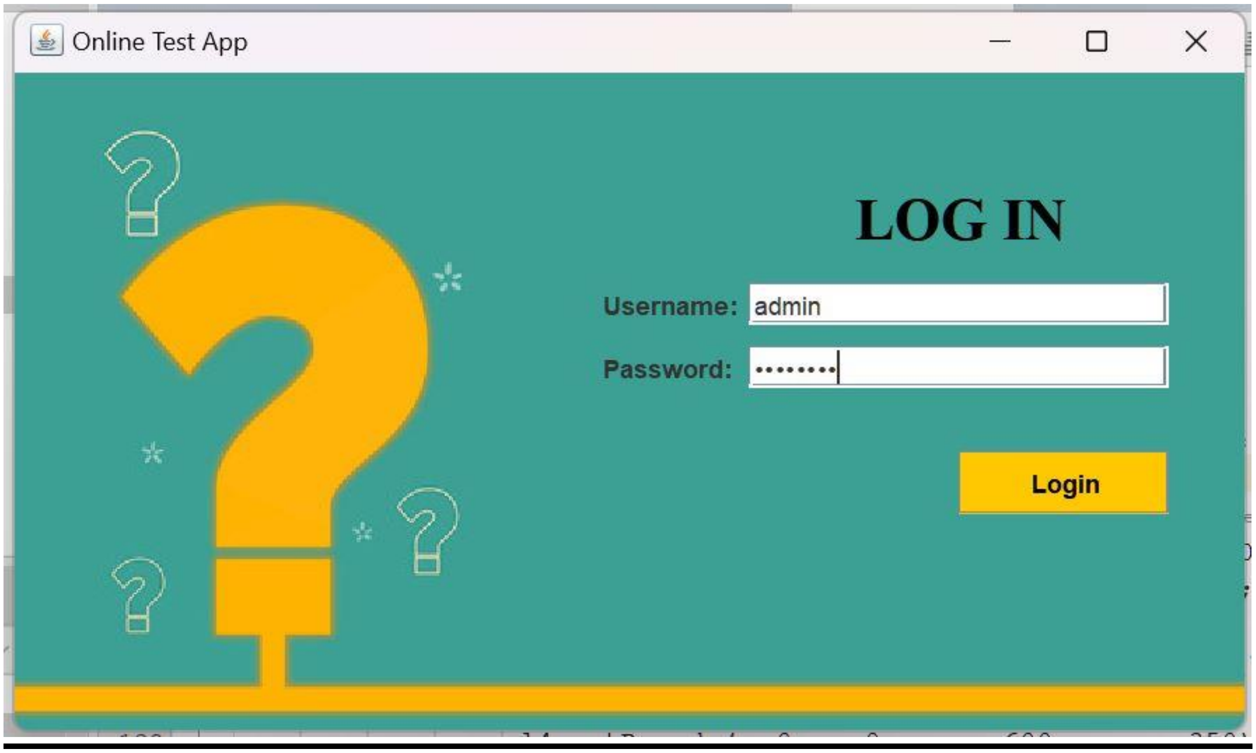
```
            radioButton[0].setText("abstraction");
            radioButton[1].setText("polymorphism");
            radioButton[2].setText("inheritance");
            radioButton[3].setText("encapsulation");
        }
        if (current == 7) {
            label.setText("Que8: Which method is automatically called when an object is created?");
            radioButton[0].setText("constructor()");
            radioButton[1].setText("final()");
            radioButton[2].setText("main()");
            radioButton[3].setText("tostring()");
        }
        if (current == 8) {
            label.setText("Que9: Which keyword is used to prevent a class from being inherited?");
            radioButton[0].setText("final");
            radioButton[1].setText("abstract");
            radioButton[2].setText("private");
            radioButton[3].setText("static");
        }
        if (current == 9) {
            label.setText("Que10: Which method is used to override a superclass method?");
            radioButton[0].setText("overload");
            radioButton[1].setText("inherit");
            radioButton[2].setText("extend");
            radioButton[3].setText("override");
        }
        label.setBounds(30, 40, 450, 20);

    }

    // declare right answers.
    boolean check() {
        if (current == 0)
            return (radioButton[3].isSelected());
```

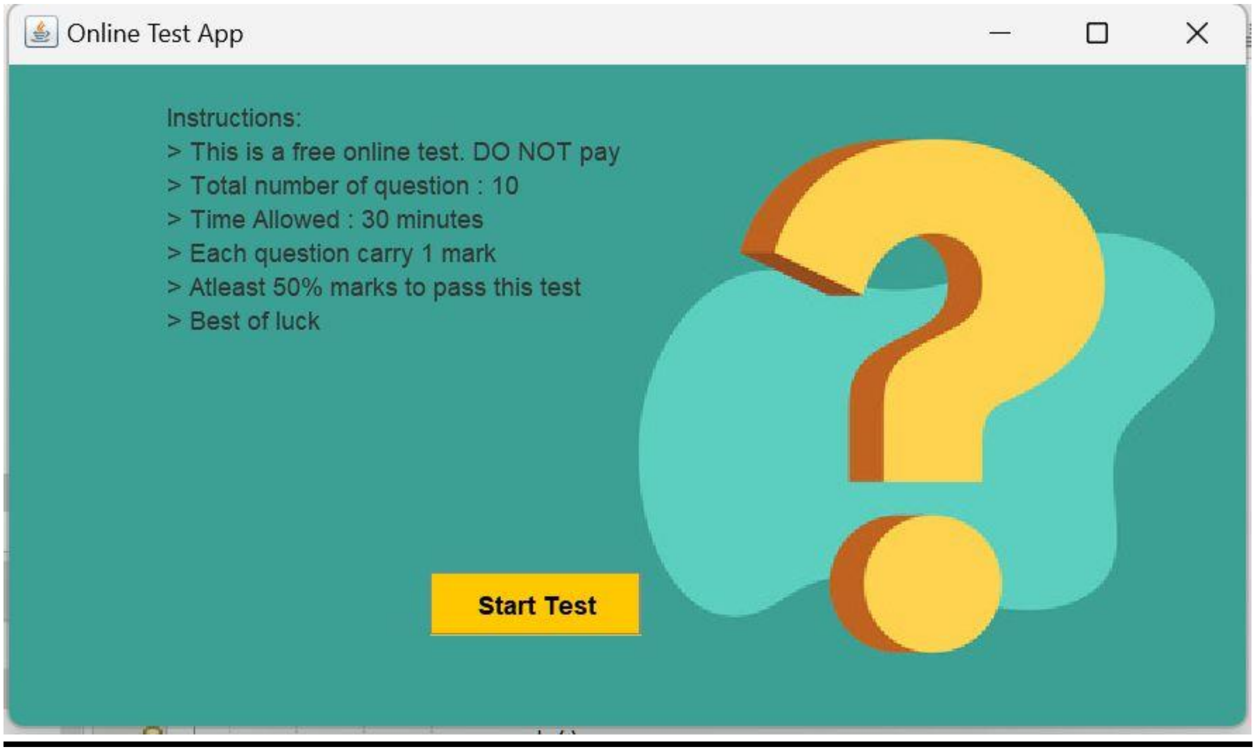
```
                    if (current == 1)
                              return (radioButton[0].isSelected());
                    if (current == 2)
                              return (radioButton[3].isSelected());
                    if (current == 3)
                              return (radioButton[1].isSelected());
                    if (current == 4)
                              return (radioButton[1].isSelected());
                    if (current == 5)
                              return (radioButton[0].isSelected());
                    if (current == 6)
                              return (radioButton[3].isSelected());
                    if (current == 7)
                              return (radioButton[0].isSelected());
                    if (current == 8)
                              return (radioButton[0].isSelected());
                    if (current == 9)
                              return (radioButton[3].isSelected());
                    return false;
          }
    public static void main(String[] args) {
        new Quiz("Online Test App");
    }
}
```

OUTPUT



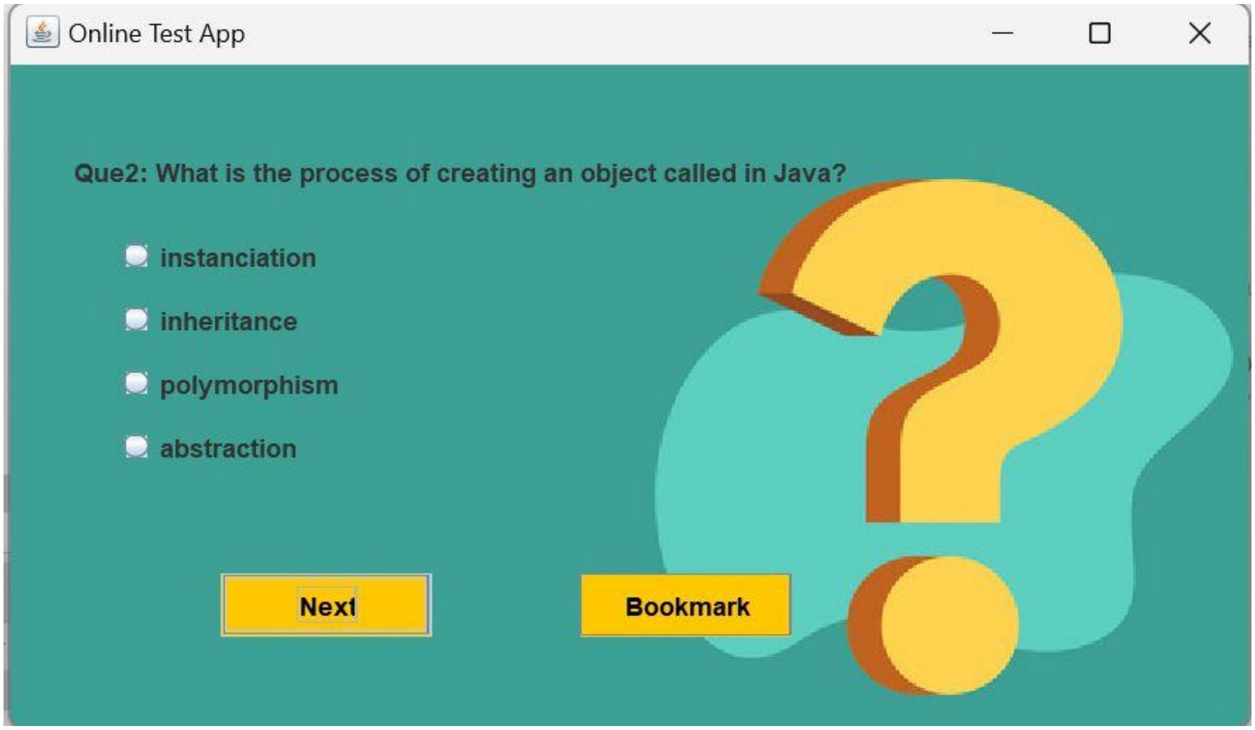
Username: admin

Password: ••••••••



Instructions:

> This is a free online test. DO NOT pay
> Total number of question : 10
> Time Allowed : 30 minutes
> Each question carry 1 mark
> Atleast 50% marks to pass this test
> Best of luck



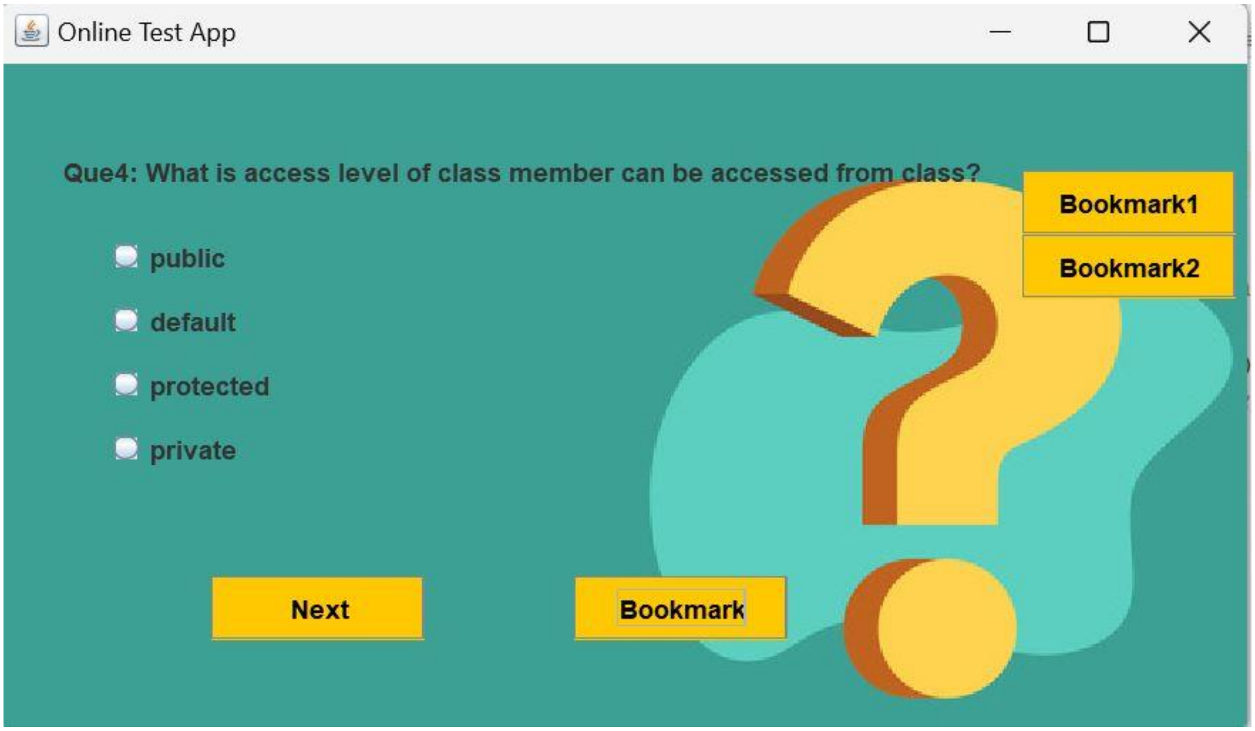
Que2: What is the process of creating an object called in Java?

○ instanciation

○ inheritance

○ polymorphism

○ abstraction



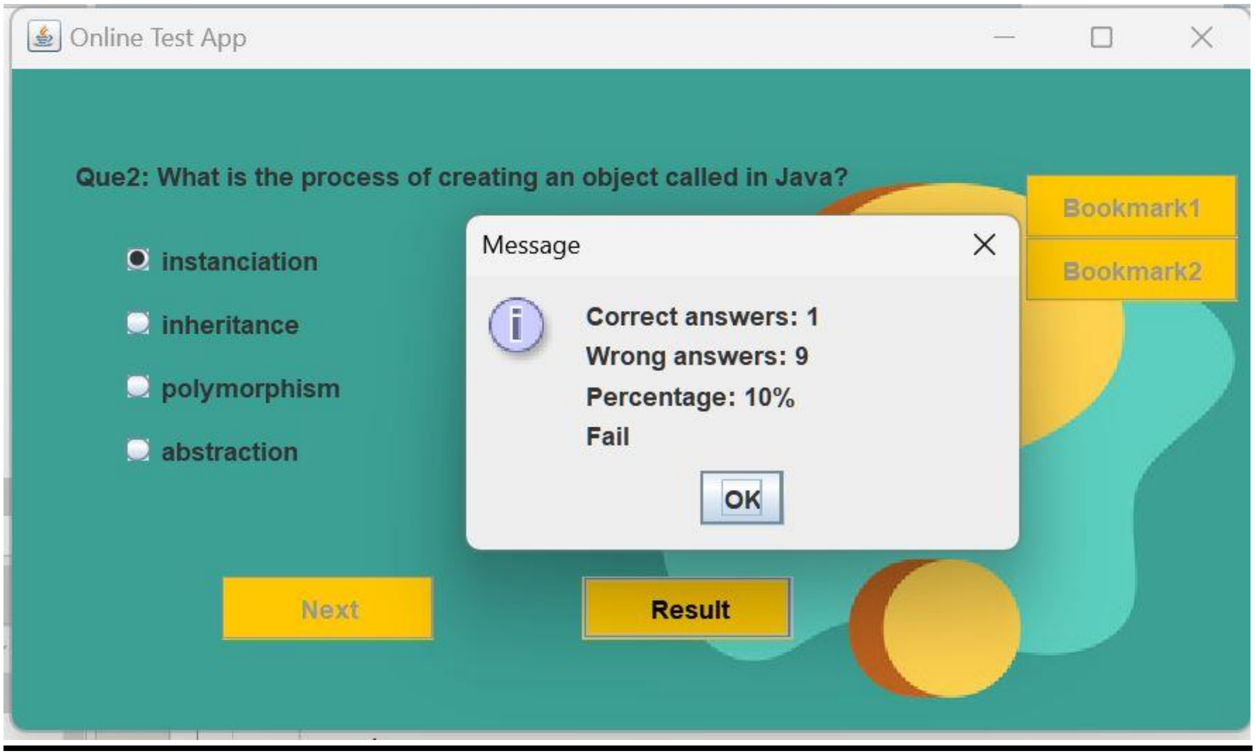
Que4: What is access level of class member can be accessed from class?

○ public

○ default

○ protected

○ private



Que2: What is the process of creating an object called in Java?

● instanciation

○ inheritance

○ polymorphism

○ abstraction

Correct answers: 1
Wrong answers: 9
Percentage: 10%
Fail