



Southern Luzon State University  
College of Engineering  
Computer Engineering Department



## **“ComEngage”**

**A quiz application**

*Submitted by:*

John Rafael H. Brigildo

BSCpE III GF

June 13, 2023

### Concept Creation:

Lack of material understanding: This is the most prevalent issue, and it may be caused by a variety of circumstances, including a lack of study time, or difficulties with the topic. For computer engineering students, a quiz application may be designed to help them learn and review material from their lessons. The software would have a question with choices containing questions for each chapter of each computer engineering subject. Students may use the quiz application to practice answering questions on the topic they are studying in class. This might assist them in identifying any areas where they want further explanation or practice.

### Title:

The **ComEngage** app is a helpful tool for students studying computer engineering. There is different set of questions with multiple choices. Additionally, it has study tool including lecture notes.

**ComEngage** may assist students in improving their understanding of the subject matter, identifying areas that need additional practice or explanation, boosting their confidence, and doing better on examinations.

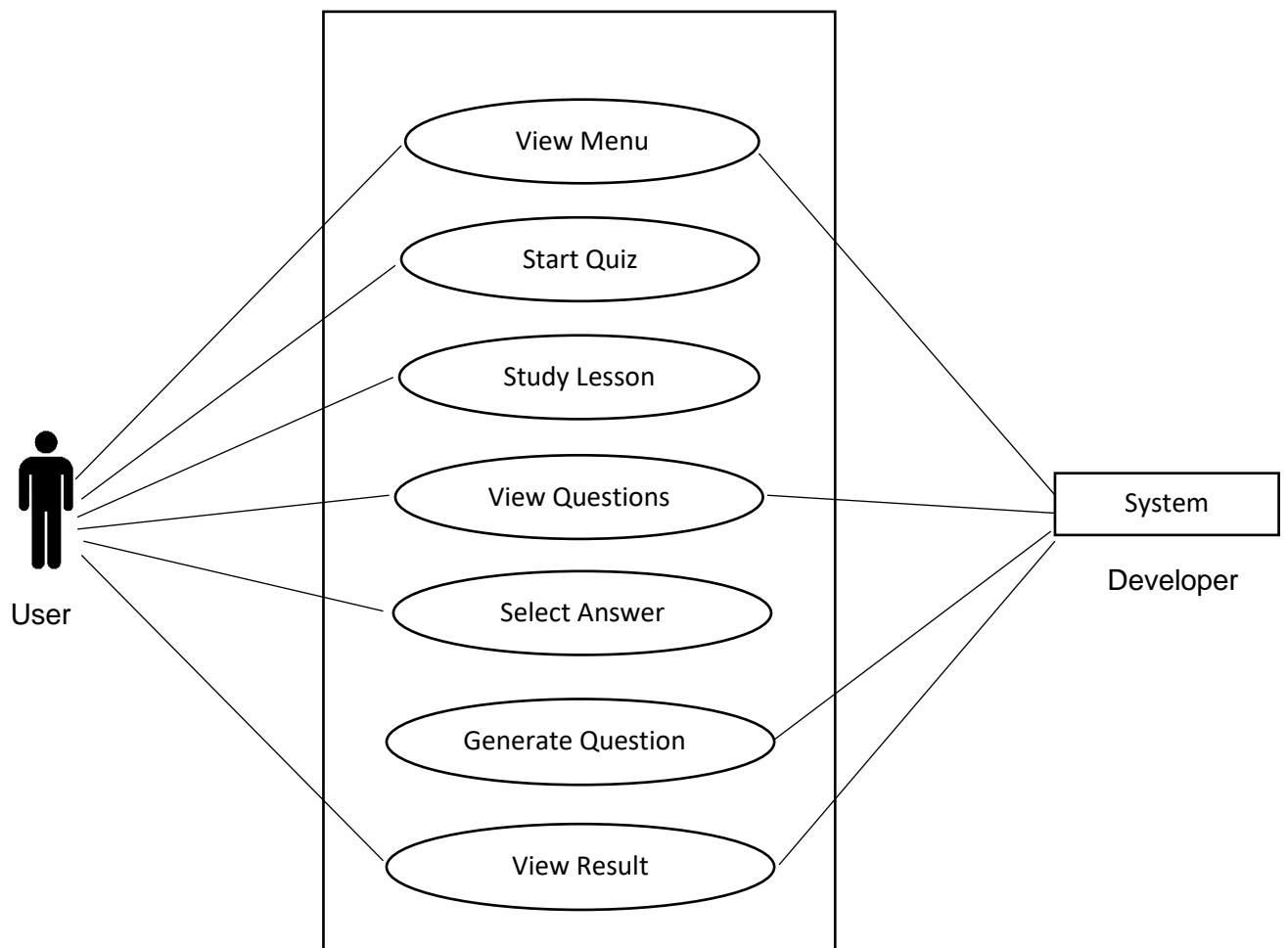
### Intended Audience:

CompEngage is a quiz app designed for computer engineering students who went to test and improve their knowledge of various subjects in the course.

### List of Features:

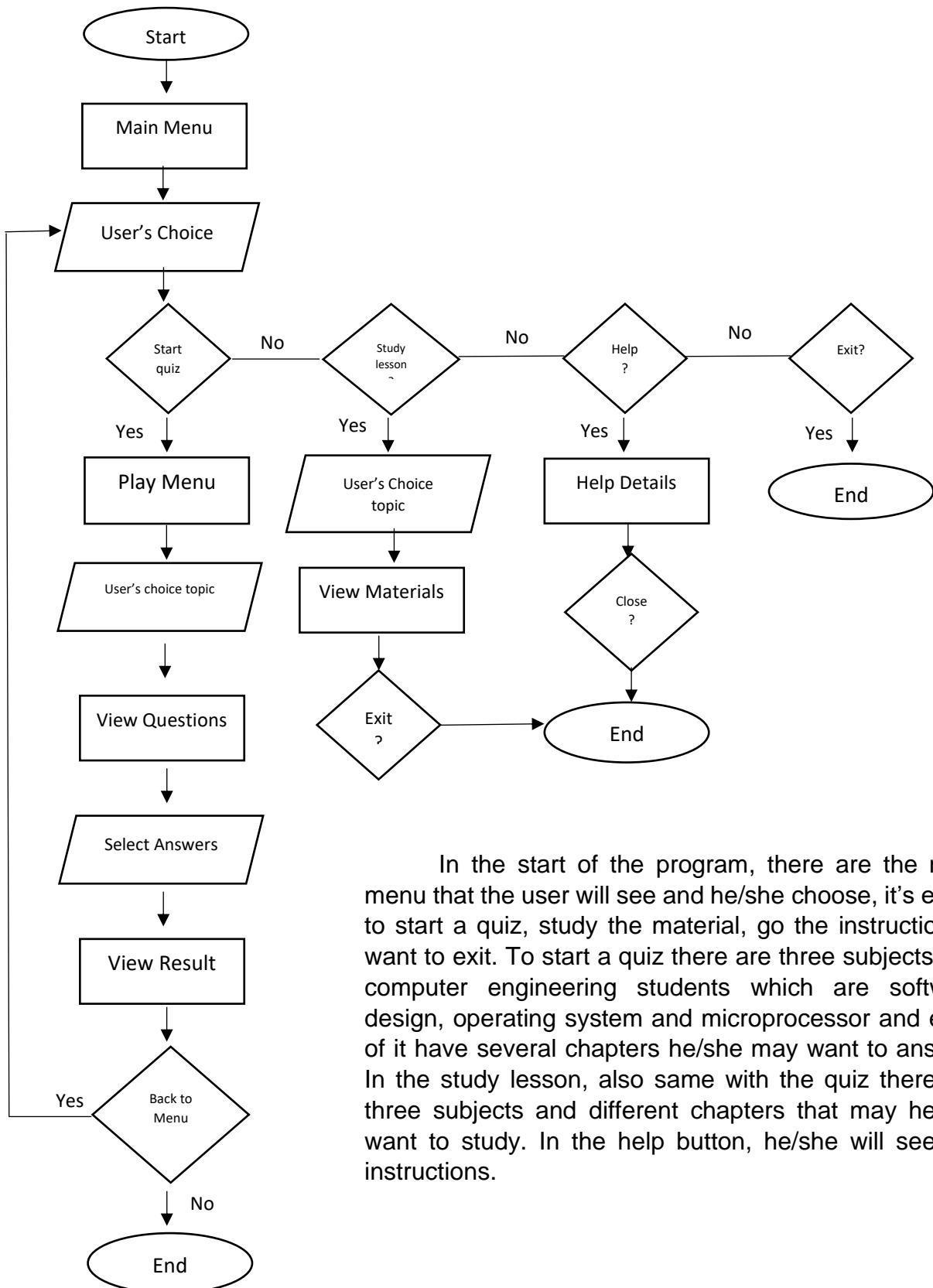
- Multiple Subjects and Chapters: The ComEngage app includes a variety of computer engineering subject and chapters. This allows students to practice answering questions on the material they are learning, regardless of what subject or chapter they are currently studying.
- Answer Questions: Students may respond to questions in a number of ways using the ComEngage app, including multiple choice.
- Show Results, Accuracy and Suggestions/Comments: After students respond to questions, the ComEngage app displays their scores, along with their accuracy and comments or recommendations for improvement. This enables them to monitor their development and pinpoint areas that need more practice.
- Study Materials: The ComEngage app has a number of study resources, including lecture notes. This enables students to choose the materials that are most effective for them and to learn the content in a number of ways.
- Random Choices: ComEngage app's random selection of questions from the question bank.

Use Case Diagram:



In this case diagram, the user will view the menu, start a quiz, study the material of each subject and chapters and view the result of her/his quiz. In the system which is the developer of the software, he will also view the menu, view questions and answer, generate the questions and view the results.

Flowchart:



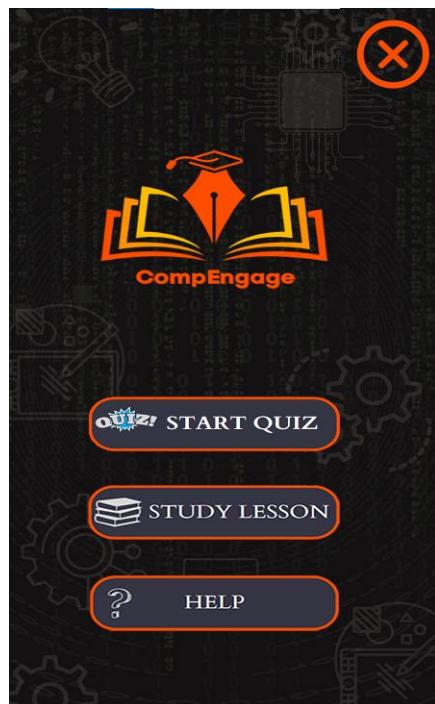
In the start of the program, there are the main menu that the user will see and he/she choose, it's either to start a quiz, study the material, go the instruction or want to exit. To start a quiz there are three subjects of a computer engineering students which are software design, operating system and microprocessor and each of it have several chapters he/she may want to answer. In the study lesson, also same with the quiz there are three subjects and different chapters that may he/she want to study. In the help button, he/she will see the instructions.

Application logo:

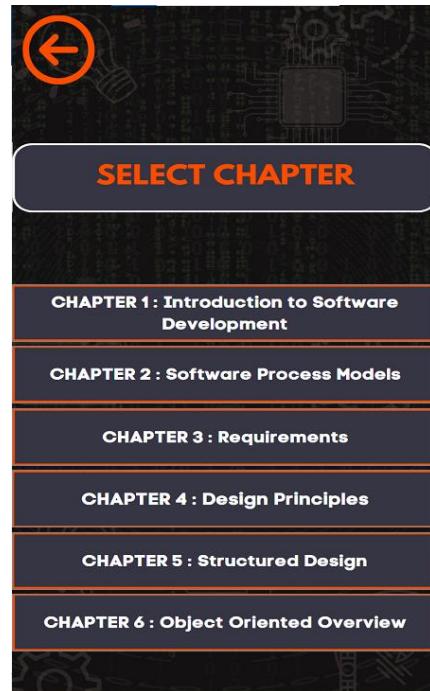


The logo of CompEngage, the quiz app, brilliantly reflects the idea of knowledge and academic brilliance. At its heart, a book sits open, its pages symbolizing the abundance of knowledge that awaits within. It depicts the immense library of information that users may explore and interact with on the platform. Adjacent to the book, a sleek pen gently depicts the act of learning, stressing the value of curiosity and intellectual progress. Its presence symbolizes the capacity to convey concepts and ideas, enabling a dynamic atmosphere for interchange and discovery. Topping off the composition, a prestigious toga hat perches on the book and pen, reflecting the spirit of academics and success. This classic emblem promotes the pursuit of higher education and acts as a tribute to the users' dedication to widening their intellectual horizons. The logo of CompEngage serves as a visual expression of the app's purpose to empower people through knowledge, establishing a community of lifelong learners.

Main User Interface:



## Multiple Subjects and Chapters:



## Answer Question:

A detailed view of a question page. At the top is a large orange back arrow. Below it is a section titled "CHAPTER 1: Introduction to Software Development". The text discusses software engineering as a process involving project management, configuration management, scheduling, estimation, baseline building, scheduling, managing people, and several other things. It notes that software development is a narrowing of the focus of software engineering just that part concerned with the creation of the actual software. It broadens the focus of programming to include analysis, design, and issues. A section titled "How to Develop Software?" follows, explaining that despite being only part of software engineering, software development is the heart of every software project. It describes a team of developers working in concert to create code. A section titled "In order to do software development, you need the following:" lists requirements: a small, well-integrated team, good communication among team members, and specific tools like version control systems.

## Show Results, Accuracy and Suggestions/Comments:

A results summary screen. At the top is a large orange back arrow. In the center, the text "YOU GOT" is displayed above "70%" and "SCORE: 7 out of 10". Below this, the message "Keep up the good work!" is shown. At the bottom are two buttons: "BACK TO MENU" and "EXIT". The background features a dark circuit board pattern.

## Random Choices:

**CHAPTER 1: Introduction to Software Development**

Quiz: 10 items

Is the process of taking a set of requirements from a user, analyzing them, designing a solution to the problem, and then implementing that solution on a computer.

Software Design  
Software Engineering  
Software Development  
Software Programming

**CHAPTER 1: Introduction to Software Development**

Quiz: 10 items

Is the process of taking a set of requirements from a user, analyzing them, designing a solution to the problem, and then implementing that solution on a computer.

Software Design  
Software Programming  
Software Engineering  
Software Development

## List of Codes:

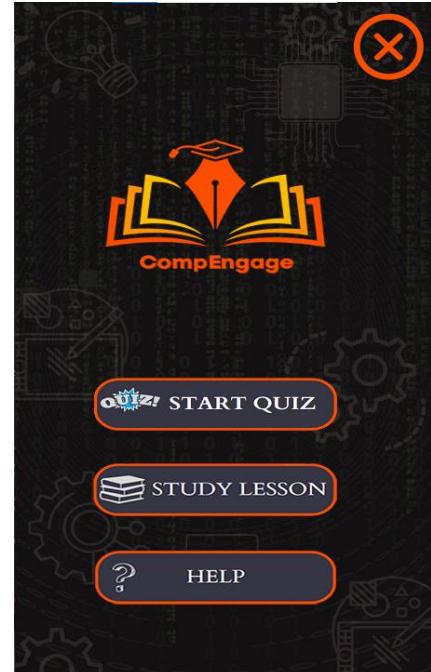
```
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;

public class Home extends JFrame {

    private JPanel contentPane;
    public static Clip clip;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        Home frame = new Home();
        frame.setVisible(true);
        //to play background music
        playSound("sounds/mainsound.wav");
    }

    /**
     * Create the frame.
     */
    public Home() {
```

## Main User Interface:



```

// Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        //help button
        JButton helpbtn = new JButton("");
        helpbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        helpbtn.setIcon(new
        ImageIcon(Home.class.getResource("/images/helphover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        helpbtn.setIcon(new
        ImageIcon(Home.class.getResource("/images/help.png")));
    }

    @Override
    public void mouseClicked(MouseEvent e)
    {
        help fHelp = new help();
        setVisible(false);
        fHelp.setVisible(true);
    }
});
        helpbtn.setIcon(new
        ImageIcon(Home.class.getResource("/images/help.png")));
        helpbtn.setBackground(new Color(0,0,0,0));
        helpbtn.setOpaque(false);
        helpbtn.setBorder(null);
        helpbtn.setBounds(84, 590, 263, 58);
        contentPane.add(helpbtn);

        //study button
        JButton studybtn = new JButton("");
        studybtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e)
    {
        studybtn.setIcon(new
        ImageIcon(Home.class.getResource("/images/studyhover.png")));
    }
});

```

```

        }

        @Override
        public void mouseExited(MouseEvent e)
{
            studybtn.setIcon(new
ImageIcon(Home.class.getResource("/images/study.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            study fStudy = new study();
            setVisible(false);
            fStudy.setVisible(true);
        }

    });
    studybtn.setIcon(new
ImageIcon(Home.class.getResource("/images/study.png")));
    studybtn.setBackground(new Color(0,0,0));
    studybtn.setOpaque(false);
    studybtn.setBorder(null);
    studybtn.setBounds(84, 496, 263, 58);
    contentPane.add(studybtn);

    //start quiz button
    JButton startbtn = new JButton("");
    startbtn.setBackground(new Color(0,0,0));
    startbtn.setOpaque(false);
    startbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent
e) {
        startbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/starthover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
{
        startbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/start.png")));
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        quizmenu newQuizmenu = new
quizmenu();
        setVisible(false);
        newQuizmenu.setVisible(true);
    }
});
    startbtn.setBorder(null);
    startbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/start.png")));
    startbtn.setBounds(84, 403, 263, 58);
}

```

```

contentPane.add(startbtn);

//logo label
JLabel logolabel = new JLabel();
logolabel.setOpaque(true);
logolabel.setBackground(new Color(0,0,0,0));
logolabel.setIcon(new
ImageIcon(Home.class.getResource("/images/logo.png")));
logolabel.setBounds(0, 0, 263, 210);

//logo panel
 JPanel logopanel = new RoundedPanel(30);
logopanel.setBounds(84, 129, 263, 210);
contentPane.add(logopanel);
logopanel.setOpaque(false);
logopanel.setLayout(null);
logopanel.add(logolabel);

//exit button
 JButton exitbtn = new JButton("");
exitbtn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent
e) {
        exitbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/exithover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
{
        exitbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/exit button.png")));
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        System.exit(0);
    }
});
exitbtn.setBorder(null);
exitbtn.setIcon(new
ImageIcon(Home.class.getResource("/images/exit button.png")));
exitbtn.setOpaque(false);
exitbtn.setBounds(362, 11, 78, 74);
contentPane.add(exitbtn);

//background label
JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}

//method to play sound
public static void playSound(String fileName) {
}

```

```

        try {
            File soundFile = new File(fileName);
            AudioInputStream audioInputStream =
            AudioSystem.getAudioInputStream(soundFile);
            clip = AudioSystem.getClip();
            clip.open(audioInputStream);
            clip.start();
            clip.loop(Clip.LOOP_CONTINUOUSLY);
        }
        catch (Exception e) {
            System.err.println(e.getMessage());
        }
    }
}

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

```

```

public class quizmenu extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        quizmenu frame = new quizmenu();
        frame.setVisible(true);

    }

    /**
     * Create the frame.
     */
    public quizmenu() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        //backbutton
        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new MouseAdapter()
{
    @Override

```

Start quiz button



```

        public void mouseEntered(MouseEvent
e) {
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
        }
        @Override
        public void mouseExited(MouseEvent e)
{
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            Home fhome = new Home();
            setVisible(false);
            fhome.setVisible(true);
        }
    );
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

    JLabel subjectlabel = new JLabel("");
    subjectlabel.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/subject.png")))
;
    subjectlabel.setBounds(63, 144, 318, 80);
    contentPane.add(subjectlabel);

    //Software Desing button
    JButton SDbtn = new JButton("");
    SDbtn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent
e) {
            SDbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/SDhover.png")
));
        }
        @Override
        public void mouseExited(MouseEvent e)
{
            SDbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/SD.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            SDquiz fSDquiz = new SDquiz();

```

```

        setVisible(false);
        fSDquiz.setVisible(true);

    }

});

SDbtn.setBorder(null);
SDbtn.setOpaque(false);
SDbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/SD.png")));
SDbtn.setBounds(73, 333, 293, 62);
contentPane.add(SDbtn);

//Microprocessor button
JButton microbtn = new JButton("");
microbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent
e) {
        microbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/microhover.pn
g")));
    }
    public void mouseExited(MouseEvent e)
{
        microbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/micro.png")));
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        MICROQuiz fMICROQuiz = new
MICROQuiz();
        setVisible(false);
        fMICROQuiz.setVisible(true);
    }
});
microbtn.setBorder(null);
microbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/micro.png")));
microbtn.setBounds(73, 543, 293, 62);
contentPane.add(microbtn);

//Operating System button
JButton OSbtn = new JButton("");
OSbtn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent
e) {
        OSbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/OShover.png")
));
    }
    public void mouseExited(MouseEvent e)
{
}
});

```

```

        OSbtn.setIcon(new
        ImageIcon(quizmenu.class.getResource("/images/OS.png")));
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        OSquiz fOSquiz = new OSquiz();
        setVisible(false);
        fOSquiz.setVisible(true);
    }
});
OSbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/OS.png")));
OSbtn.setBorder(null);
OSbtn.setBounds(73, 439, 293, 62);
contentPane.add(OSbtn);

JLabel quizmenulabel = new JLabel("");
quizmenulabel.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/bg.png")));
quizmenulabel.setBounds(0, 0, 450, 730);
contentPane.add(quizmenulabel);
}
}

```

```

import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.util.*;

```

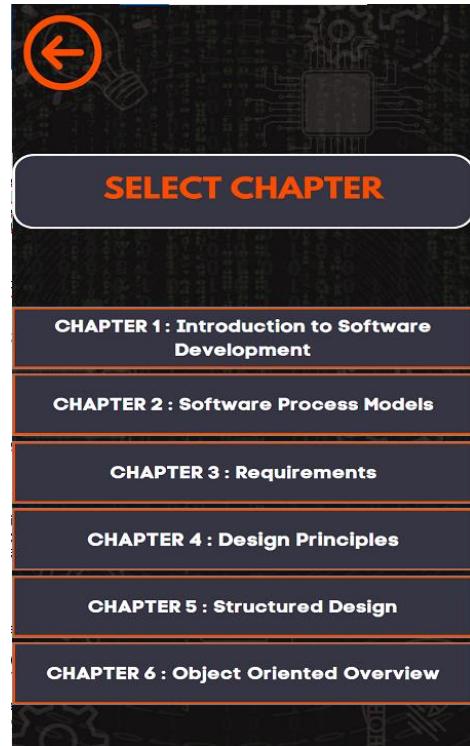
```

public class SDquiz extends JFrame {

    private JPanel contentPane;
    public static Clip clip1;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        SDquiz frame = new SDquiz();
        frame.setVisible(true);
    }
    /**
     * Create the frame.
     */
    public SDquiz() {
        // Set the icon image of the frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);
    }
}

```

Software Design button



```

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 450, 730);
setLocationRelativeTo(null);
setResizable(false);
contentPane = new JPanel();
setContentPane(contentPane);
contentPane.setLayout(null);

JButton backbtn = new JButton();
backbtn.setOpaque(false);
backbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent
e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
{
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        quizmenu fquizmenu = new
quizmenu();
        setVisible(false);
        fquizmenu.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {

```

```

ch1_SD();
    setVisible(false);
    fch1_SD.setVisible(true);
    Home.clip.stop();

    playSound("sounds/quizsound.wav");
        }
    });
    ch1btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch1.png")));
    ch1btn.setBounds(0, 295, 450, 61);
    contentPane.add(ch1btn);

    JButton ch2btn = new JButton("");
    ch2btn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent
e) {
            ch2_SD fCh2_SD = new
ch2_SD();
            setVisible(false);
            fCh2_SD.setVisible(true);
            Home.clip.stop();

            playSound("sounds/quizsound.wav");
            }
        });
    ch2btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch2.png")));
    ch2btn.setBounds(0, 360, 450, 61);
    contentPane.add(ch2btn);

    JButton ch3btn = new JButton("");
    ch3btn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent
e) {
            ch3_SD fCh3_SD = new
ch3_SD();
            setVisible(false);
            fCh3_SD.setVisible(true);
            Home.clip.stop();

            playSound("sounds/quizsound.wav");
            }
        });
    ch3btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch3.png")));
    ch3btn.setBounds(0, 426, 450, 61);
    contentPane.add(ch3btn);

    JButton ch4btn = new JButton("");
    ch4btn.addMouseListener(new MouseAdapter() {
        @Override

```

```

        public void mouseClicked(MouseEvent
e) {
                ch4_SD fCh4_SD = new
ch4_SD();
                setVisible(false);
fCh4_SD.setVisible(true);
Home.clip.stop();

                playSound("sounds/quizsound.wav");
}
});
ch4btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch4.png")));
ch4btn.setBounds(0, 491, 450, 61);
contentPane.add(ch4btn);

 JButton ch5btn = new JButton("");
ch5btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
                ch5_SD fCh5_SD = new
ch5_SD();
                setVisible(false);
fCh5_SD.setVisible(true);
Home.clip.stop();

                playSound("sounds/quizsound.wav");
}
});
ch5btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch5.png")));
ch5btn.setBounds(0, 557, 450, 61);
contentPane.add(ch5btn);

 JButton ch6btn = new JButton("");
ch6btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
                ch6_SD fCh6_SD = new
ch6_SD();
                setVisible(false);
fCh6_SD.setVisible(true);
Home.clip.stop();

                playSound("sounds/quizsound.wav");
}
});
ch6btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch6.png")));
ch6btn.setBounds(0, 622, 450, 61);
contentPane.add(ch6btn);

JLabel SDquizlabel = new JLabel("");

```

```
SDquizlabel.setIcon(new  
ImageIcon(SDquiz.class.getResource("/images/bg.png")));  
SDquizlabel.setBounds(0, 0, 450, 730);  
contentPane.add(SDquizlabel);  
}  
//to play loop sound  
    public static void playSound(String fileName) {  
        try {  
            File soundFile = new File(fileName);  
            AudioInputStream audioInputStream =  
AudioSystem.getAudioInputStream(soundFile);  
            clip1 = AudioSystem.getClip();  
            clip1.open(audioInputStream);  
            clip1.start();  
            clip1.loop(Clip.LOOP_CONTINUOUSLY);  
        } catch (Exception e) {  
            System.err.println(e.getMessage());  
        }  
    }  
    //to play sound  
    public static void  
playSoundNotLoop(String fileName) {  
        try {  
            File soundFile = new  
File(fileName);  
            AudioInputStream  
audioInputStream =  
AudioSystem.getAudioInputStream(soundFile);  
            clip1 = AudioSystem.getClip();  
            clip1.open(audioInputStream);  
            clip1.start();  
        } catch (Exception e) {  
  
System.err.println(e.getMessage());  
        }  
    }  
}
```

```

public class ch1_SD extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
    JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;

    //question and choices in an array
    String[][] ch1questions = {
        {"Is the process of taking a set of
        requirements from a \n"
        + "user, analyzing them,
        designing a solution to the \n"
        + "problem, and then
        implementing that solution on a \ncomputer.",
        "Software Development",
        "Software Programming", "Software Design", "Software
        Engineering"},

        {"Is central to software
        development, but it's not \nthe whole thing.", "Programming",
        "Refactoring",
        "Encoding", "Testing"},

        {"Encapsulates what you're
        going to do to implement\n your project.", "Project Plan", "Project
        Design",
        "Project Research", "Project Study"},

        {"Involves a process and
        includes software development,\nbut it also includes the entire
        management side of\ncreating a computer."
        + "program that people will use",
        "Software Engineering", "Software Programming", "Software
        Design", "Software Development"},

        {"Keeping a team together
        across several projects\n is a major job of the _____",
        "Team's Manager", "Team's Leader",
        "Programmer", "Client"},

        {"_____ have fewer lines of
        communication than \nlarger ones.", "Small teams", "Large
        teams", "Company teams", "Programmer teams"},

        {"It is a methodologies that
        strongly encourage\ncustomers to be part of the development
        team"
        + "and,\neven better, to be on
        site daily.", "Agile Development", "Scrum", "Waterfall","code and
        fix"},

        {"Managing _____
        during a project is one of \nthe single most important skills a
        software developer\n"
        + "can have.", "requirements
        churn", "requirements chum", "requirements digging",
        "requirements development"},

}

```

## Chapter 1 Software Design quiz

**CHAPTER 1: Introduction to Software Development**

**Quiz: 10 items**

Is the process of taking a set of requirements from a user, analyzing them, designing a solution to the problem, and then implementing that solution on a computer.

**Software Programming**

**Software Design**

**Software Engineering**

**Software Development**

```

        {"It is essential to controlling
requirements and \nrequirements churn during"
+ " a project.", "Communication
with customer", "Communication with manager", "Communication
with other", "Communication with worker"},

        {"Every project, no matter how
big or small, follows a\n_____","process","steps","level",
"stage"},

};

public ch1_SD() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(ch1_SD.class.getResource("/images/ch1SD.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("_____"
);

    lblNewLabel.setForeground(Color.WHITE);
    lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
    lblNewLabel.setBounds(10, 202, 440,
29);
    contentPane.add(lblNewLabel);

    JPanel buttonPanel = new JPanel();
    buttonPanel.setOpaque(false);
    buttonPanel.setBackground(new
Color(0, 0, 0, 0));

```

```
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkAnswer((JButton) e.getSource());
            }
        });
    }
    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel();
    bglabel.setIcon(new
ImageIcon(ch1_SD.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //initial values
    random_ans = new Random();
    //show questions
    showQuestion();

}

//method to show questions
private void showQuestion() {

    questionsTextArea.setText(ch1questions[currentQuestio
n][0]);
}
```

```

        String[] answers =
Arrays.copyOfRange(ch1questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new Font("Arial",
Font.BOLD, 17));
        }
    }
    //method to randomize choices
    private void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j = random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }

    // method to check answer
    private void checkAnswer(JButton button) {
        if
(button.getText().equals(ch1questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done on answering
            if (currentQuestion >= ch1questions.length) {
                Result_SD.Result_ch1SD fCh1sd = new
Result_SD().new Result_ch1SD();
                setVisible(false);
                fCh1sd.setVisible(true);

            } else {
                //continue to show question
                showQuestion();
            }
        } else {
            currentQuestion++;
            if(currentQuestion > 9) {
                Result_SD.Result_ch1SD fCh1sd = new
Result_SD().new Result_ch1SD();
                setVisible(false);
                fCh1sd.setVisible(true);
                SDquiz.clip1.stop();

                playSoundNotLoop("sounds/resultbgmusic.wav");

            } else {
                showQuestion();
            }
        }
    }
}

class ch2_SD extends JFrame {

```

Chapter 2 Software Design quiz

```

private JPanel contentPane;
private JButton[] answerButtons = new
JButton[4];

private JTextArea questionsTextArea;
private int currentQuestion;
private Random random_ans;
public static int scores;
//question and choices in an array
String[][] ch2questions = {
    {"It is a model that have more
clearly defined phases, \r\n"
        + "and more
requirements for sign-off on completion of \r\n"
        + "a phase
before moving on to the next phase.",

        "Plan-driven", "Agile
development", "Lean Software", "Research-driven"},

    {"The person who generates the
requirements for the \r\n"
        + "product and
prioritizes them.", "Product owner", "Scrum master",
        "Project manager", "Product
master"},

    {"Phases in this models tend to
blur and to be less\r\n"
        + "
documentation of work products required.", "Agile
Development", "Plan-driven",
        "Lean software",
        "Research-driven"},

    {"It provides improved progress
visibility for both the\r\n"
        + "customer and
project management.", "Evolutionary prototyping", "Code and fix",
        "Waterfall", "iterative"},

    {"It can influence the size of the
team or, less often, \r\n"
        + "the types of
tools available to the team.", "Cost", "Time",
        "Quality", "Features"},

    {"Is your delivery schedule and is
unfortunately many \r\n"
        + "times
imposed on you from the outside.", "Time", "Cost",
        "Quality",
        "Features"},

    {"It is a process model take a
minimal amount of time\r\n"
        + "to understand
the problem and then start coding.", "Code and fix",
        "Evolutionary prototyping", "Waterfall", "iterative"},

    {"Are what the product actually
does. This is what\r\n"
        + "developers
should always focus on.", "Features", "Cost",
}

```

**CHAPTER 2 : Software Process Models**

**Quiz: 10 items**

**It is a model that have more clearly defined phases, and more requirements for sign-off on completion of a phase before moving on to the next phase.**

**Research-driven**

**Plan-driven**

**Agile development**

**Lean Software**

```

        "Quality",
        "Time"}, {"It was created in 1970 by
Winston Royce, and addresses\r\n"
+ "all of the
standard life cycle phases."
+ " a project.", "Waterfall", "Code
and fix", "Evolutionary prototyping", "iterative"},
 {"Is the number and severity of
defects you're willing \r\n"
+ "to release
with.", "Quality", "Cost",
"Features",
"Time"}, };

public ch2_SD() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/ch2.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("____");
    lblNewLabel.setForeground(Color.WHITE);
    lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
    lblNewLabel.setBounds(10, 202, 440,
29);
    contentPane.add(lblNewLabel);
}

```

```

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();
        questionsTextArea.setEditable(false);

        questionsTextArea.setForeground(Color.WHITE);
        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    private void showQuestion() {

```

```

        questionsTextArea.setText(ch2questions[currentQuestion][0]);
        String[] answers =
    Arrays.copyOfRange(ch2questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }
    }
    //method to randomize choices
    private void shuffleAnswers(String[]
answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }

    // method to check answer
    private void checkAnswer(JButton
button) {
        if
(button.getText().equals(ch2questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done on
answering
            if (currentQuestion >=
ch2questions.length) {
                Result_SD.Result_ch2SD fCh2sd = new
Result_SD().new Result_ch2SD();
                setVisible(false);
                fCh2sd.setVisible(true);
            } else {
                //continue to show question
                showQuestion();
            }
        } else {
            currentQuestion++;
            if(currentQuestion > 9) {
                Result_SD.Result_ch2SD
fCh2sd = new Result_SD().new Result_ch2SD();
                setVisible(false);
                fCh2sd.setVisible(true);
                SDquiz.clip1.stop();

                playSoundNotLoop("sounds/resultbgmusic.wav");
            }
        }
    }
}

```

<pre>         showQuestion();     }  }  } </pre>	
<pre> public class ch3_SD extends JFrame {      private JPanel contentPane;     private JButton[] answerButtons = new JButton[4];     private JTextArea questionsTextArea;     private int currentQuestion;     private Random random_ans;     public static int scores;     //question and choices in an array     String[][] ch3questions = {         {"Which type of requirements represent the features \r\n", "that the user will see and be able to use?", "user requirements", "domain requirements", "non-functional requirements", "non-requirements"}, {"What type of requirements are imposed on you by the \r\n", "application domain of the program?", "Domain requirements", "Hardware requirements", "User requirements", "Software requirements"}, {"What is the key idea in requirements gathering \r\n", "for most agile methodologies?", "User story", "Software testing", "Graphic design", "Database administration"}, {"What acronym is used to describe the characteristics \r\n", "of a good user story?", "INVEST", "KANBAN", "SCRUM", "LEAN"}, {"What is the key idea in requirements gathering \r\n", "for most agile methodologies?", "User story", "Software testing", "Graphic design", "Database administration"}, {"Who writes the acceptance criteria in an agile \r\n", "+ environment?", "Product owner", "CEO", "Developers", "Scrum Master"}, {"What is the name of the list that contains the total \r\n", ""}} </pre>	<p>Chapter 3 Software Design quiz</p>  <p><b>CHAPTER 3 : Requirements</b></p> <hr/> <p><b>Quiz: 15 items</b></p> <p>Which type of requirements represent the features that the user will see and be able to use?</p> <p>non-functional requirements</p> <p>domain requirements</p> <p>non-requirements</p> <p>user requirements</p>

+ "number of things needed to create the product?", "Product backlog", "Functional specification", "Plan card stack", "Development list"}, {"Who is responsible for decomposing stories into \ntasks?", "The development team", "The stakeholders", "The project manager", "The product owner"}, {"What practice is usually implemented in testing the \r\n"}, + "traceability of a feature in Agile?", "TDD and acceptance criteria", "Feature branching", "Manual testing", "SRE"}, {"What is one reason for splitting user stories?", "To simplify the development process", "To make estimation more difficult", "To focus on the size of the story", "To increase the value to the customer"}, {"When you first write the functional specification, there\r\n"}, + "will be one or two things you don't know.", "Open Issues", "Scenario's of typical usage", "Author's name", "Design and New Feature Ideas"}, {"This is usually either the development manager or the \r\n"}, + "project manager, depending on how your company sets\r\n", + "up development projects.", "Author's name", "Open Issues", "Scenario's of typical usage", "Design and New Feature Ideas"}, {"This is a substantive discussion about what the \r\n"}, + "product owner really wants from the story.", "Conversation", "Card", "Confirmation", "Customer"}, {"It must describe a feature or a service that the \r\n"}, + "customer wants.", "Valuable", "Estimable", "Negotiable", "Independent"}, {"The key idea here is that the team needs to know when \r\n"}, + "all the requirements for the task have been completed.", "Measureable", "Estimable", "Valuable", "Negotiable"}  
 };  

```

public ch3_SD() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
}

```

```

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/ch3.png")));
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
15 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();

```

```

        questionsTextArea.setEditable(false);

        questionsTextArea.setForeground(Color.WHITE);
        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    private void showQuestion() {

        questionsTextArea.setText(ch3questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch3questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }
    }
    //method to randomize choices
    private void shuffleAnswers(String[]
answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
randomAns.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void checkAnswer(JButton
button) {
        if
(button.getText().equals(ch3questions[currentQuestion][1])) {
            currentQuestion++;
        }
    }
}

```

```

        scores++;
        //show the result if the user is done on
answering
        if (currentQuestion >=
ch3questions.length) {
            Result_SD.Result_ch3SD fCh3sd = new
Result_SD().new Result_ch3SD();
            setVisible(false);
            fCh3sd.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 14) {
            Result_SD.Result_ch3SD
fCh3sd = new Result_SD().new Result_ch3SD();
            setVisible(false);
            fCh3sd.setVisible(true);
            SDquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    } else {
        showQuestion();
    }
}
}

```

```

class ch4_SD extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch4questions = {
        {"What did J. Rittel and Melvin
M. Webber define?", "Wicked problems",
        "Complicated domains", "Tame problems", "Sorting problems"},

        {"According to Conklin, Rittel and
Webber, what is \r\n" + "the approach
of real wicked problem solvers?", "An opportunity-driven", "A
circular approach", "A linear approach", "A
waterfall approach"},

        {"What is design in software
development?", "Heuristic", "Fixed methodology",}
    }
}

```

Chapter 4 Software Design quiz

```

        "Algorithmic",
"Ruleset"),
        {"Which principle is closely
related to modularity \r\n"
         + "and suggests
separating functional pieces of a \r\n"
         + "design
cleanly?", "Separation of concerns", "Adaptability", "Simplicity",
"Fitness of purpose",
        {"What is the term for the degree
to which classes \r\n"
         + "depend on
each other in object-oriented design?", "Coupling",
"Polymorphism",
         "Inheritance",
"Abstraction"),
        {"What is the complement of
loose coupling in \r\n"
         + "modules?",
"High cohesion", "Coupling level", "Low cohesion", "Intermodule
dependence",
        {"What is the desirable
characteristic of keeping in \r\n"
         + "mind during
software development?", "Portability", "Aesthetics",
"Color", "Size"},
        {"According to the context, what
can be used as a \r\n"
         + "design
language?", "Diagrams", "Text messages", "Sculptures", "Audio
recordings"},
        {"Can formal processes be
imposed to create software \r\n"
         + "designs?",
"No", "It depends on the software being
designed", "Yes", "Sometimes",
        {"Which step of Bill Curtis'
process for software \r\n"
         + "design
involves implementing plans?", "Step 4", "Step 1", "Step 2", "Step
3",
        };
public ch4_SD() {
    // Set the icon image of the frame
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
}

```

**CHAPTER 4 : Design Principles**

**Quiz: 10 items**

What did J. Rittel and Melvin M. Webber define?

Sorting problems

Complicated domains

Wicked problems

Tame problems

```

        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/ch4.png")));
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
10 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();
        questionsTextArea.setEditable(false);
    }
}

```

```

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
contentPane.add(questionsTextArea);

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabell.setBounds(0, 0, 450, 730);
contentPane.add(bglabell);

//initial values
currentQuestion = 0;
scores = 0;
//assign random
random_ans = new
Random();
//show questions
showQuestion();
}

//method to show questions
private void showQuestion() {

questionsTextArea.setText(ch4questions[currentQuestion][0]);
String[] answers =
Arrays.copyOfRange(ch4questions[currentQuestion], 1, 5);
shuffleAnswers(answers);
for (int i = 0; i < 4; i++) {
answerButtons[i].setText(answers[i]);
answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
}
}

//shuffle options
private void shuffleAnswers(String[]
answers) {
for (int i = 0; i < answers.length; i++) {
int j =
random_ans.nextInt(answers.length);
String temp = answers[i];
answers[i] = answers[j];
answers[j] = temp;
}
}

// method to check answer
private void checkAnswer(JButton
button) {
if
(button.getText().equals(ch4questions[currentQuestion][1])) {
currentQuestion++;
scores++;
}
}

```

```

        //show the result if the user is done on
answering
        if (currentQuestion >=
ch4questions.length) {
            Result_SD.Result_ch4SD fCh4sd = new
Result_SD().new Result_ch4SD();
            setVisible(false);
            fCh4sd.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {
            Result_SD.Result_ch4SD
fCh4sd = new Result_SD().new Result_ch4SD();
            setVisible(false);
            fCh4sd.setVisible(true);
            SDquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
        else {
            showQuestion();
        }
    }
}
}

```

```

class ch5_SD extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch5questions = {
        {"The process continues until
each sub-problem is \r\n",
         + "small enough
to be either trivial"
         + "or very easy
to solve.",

         "structured programming",
         "object-oriented programming", "step-wise refinement", "top-down
decomposition"},

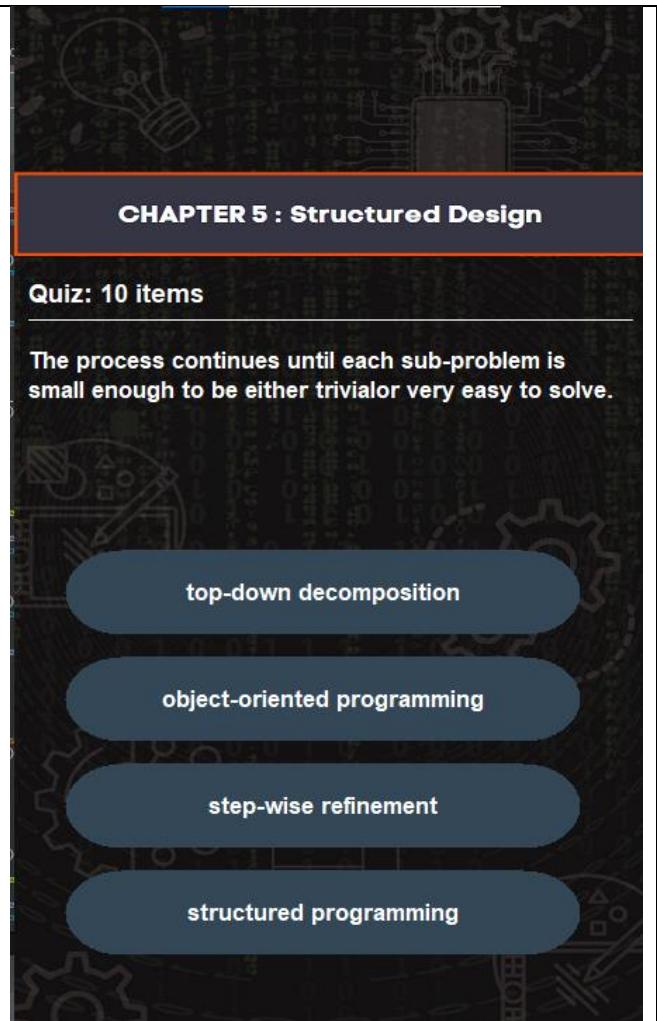
        {"Who formalized the structured
design technique in \r\n",
         + "his 1971
paper?", "Niklaus Wirth", "Bill Gates",
         "Alan Turing", "Tim Berners-
Lee"},

    }
}

```

Chapter 5 Software Design quiz

Parnas propose in his \"On the Criteria to Be Used in Systems into Modules\"?", "Modularity", "Object-oriented analysis and design", "Information hiding", "characteristics of modularity?", "Tight coupling", "Information hiding", "Loose coupling", "Encapsulation"}, "What is the objective of encapsulating services "Which form of coupling is not considered good?", "Control coupling", "Parameter coupling", "Global-data coupling", "Data coupling"}, {"According to information hiding, what should be and visible only to the operations it?", "Data and behaviors in a module", "Only data in external modules", "All data in the program", "Data and behaviors in all modules"}, {"What is the process to sort the circularly of text?", "Sort by the index words", "Sort alphabetically", "Sort by the last word in each line", "Randomly sort the lines"}, {"_____ describes a set of classic design "methodologies.", "Structured design", "Structured programming", "Modular decomposition", "Top-down decomposition"}, {"It is the immediate precursor to the modern oriented methodologies and the ideas of encapsulation and hiding.", "Modular decomposition", "Structured design", "Structured programming", "Top-down decomposition"}, };  <pre>public ch5_SD() {     // Set the icon image of the frame</pre>
---



## CHAPTER 5 : Structured Design

**Quiz: 10 items**

The process continues until each sub-problem is small enough to be either trivial or very easy to solve.

top-down decomposition

object-oriented programming

step-wise refinement

structured programming

```
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new ImageIcon(Home.class.getResource("/images/ch5.png")));
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
10 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
        }
    }
}
```

```

        answerButtons[i].addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent e) {
        checkAnswer(( JButton ) e.getSource());
    }
});
}

contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
contentPane.add(questionsTextArea);

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);

//initial values
currentQuestion = 0;
scores = 0;
//assign random
random_ans = new
Random();
//show questions
showQuestion();
}
//method to show questions
private void showQuestion() {

questionsTextArea.setText(ch5questions[currentQuestio
n][0]);
String[] answers =
Arrays.copyOfRange(ch5questions[currentQuestion], 1, 5);
shuffleAnswers(answers);
for (int i = 0; i < 4; i++) {
    answerButtons[i].setText(answers[i]);
    answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
}
}

//shuffle options
private void shuffleAnswers(String[]
answers) {
    for (int i = 0; i < answers.length; i++) {
        int j =
random_ans.nextInt(answers.length);
}
}

```

```

        String temp = answers[i];
        answers[i] = answers[j];
        answers[j] = temp;
    }
}

// method to check answer
private void checkAnswer(JButton
button) {
    if
(button.getText().equals(ch5questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done on
answering
        if (currentQuestion >=
ch5questions.length) {
            Result_SD.Result_ch5SD fCh5sd = new
Result_SD().new Result_ch5SD();
            setVisible(false);
            fCh5sd.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {
            Result_SD.Result_ch5SD
fCh5sd = new Result_SD().new Result_ch5SD();
            setVisible(false);
            fCh5sd.setVisible(true);
            SDquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    }
}
}

class ch6_SD extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch6questions = {
        {"Class diagrams have three
sections. What are they?",
```

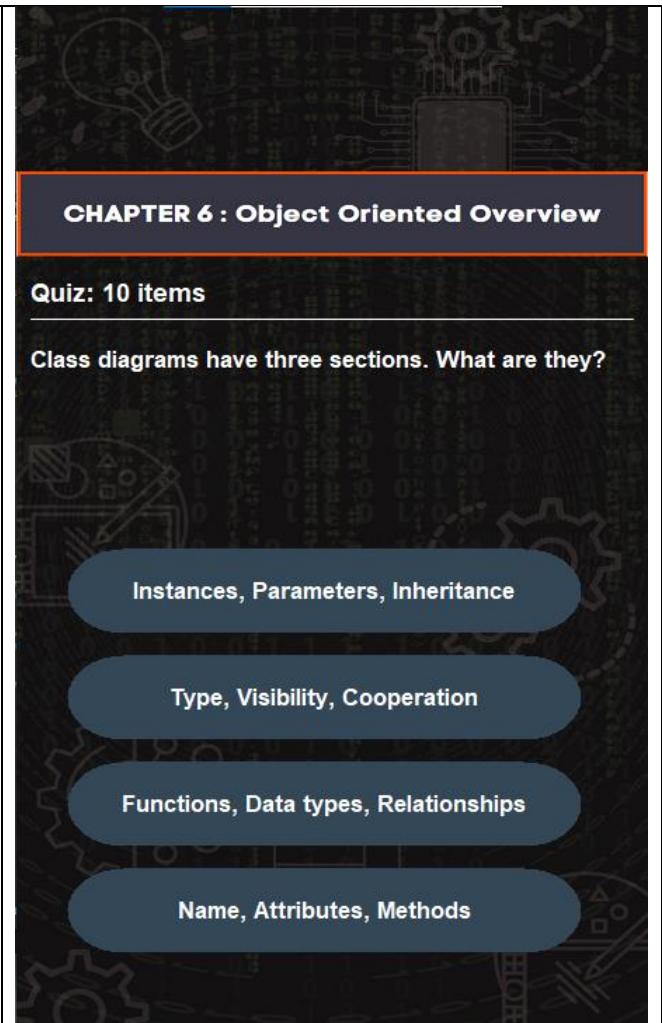
Chapter 6 Software Design quiz

"Name, Attributes, Methods",  
 "Instances, Parameters, Inheritance", "Functions, Data types,  
 Relationships", "Type, Visibility, Cooperation",  
 {"A \_\_\_\_\_ is a description of  
 what a program does \r\n"  
     + "in a particular  
 situation.", "use case", "class diagram",  
         "user story", "user  
 requirements"},  
     {"An outside agent that gets the  
 ball rolling.", "actor", "user",  
         "programmer",  
     "manager"},  
     {"Which step comes after  
 identifying objects, "  
     + "methods,  
     \and algorithms in the design process?", "Implementation and  
     Testing", "Release/Maintenance/Evolution", "Requirements  
     Gathering and Analysis", "Feature list creation"},  
     {"What is the purpose of  
 scenarios in generating \r\n"  
     + "feature  
 lists?", "To help generate the feature list", "To provide user  
 support",  
         "To show the program's  
 functionality", "To test the program"},  
         {"What is the name of the  
 document that contains "  
     + "the \ninitial  
 set of requirements?", "Feature list", "User story", "Problem  
 statement", "Program description"},  
         {"What does OOD stand for?.",  
 "Object-oriented design", "Object-oriented diagram", "Object-  
 oriented device", "Object-oriented development"},  
         {"What are classes in object-  
 oriented programming?", "Templates for objects", "Shared data  
 areas", "Instances of objects", "Methods that change state of  
 object"},  
         {"It is concerned with developing  
 an object-"  
     +  
 "oriented\nsystem model to satisfy \r\n"  
     +  
 "requirements.", "OOD", "OOA", "OOP", "OOS"},  
         {"It is concerned with developing  
 an object model "  
     + "of the  
     \application domain.", "OOA", "OOP", "OOS", "OOD"},  
     };  
  

```

public ch6_SD() {
    // Set the icon image of the frame
    setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);
}

```



**CHAPTER 6 : Object Oriented Overview**

**Quiz: 10 items**

**Class diagrams have three sections. What are they?**

**Instances, Parameters, Inheritance**

**Type, Visibility, Cooperation**

**Functions, Data types, Relationships**

**Name, Attributes, Methods**

```

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/ch6.png")));
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
10 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            })
        }
    }
}

```

```

        }

    }

    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //assign random
    random_ans = new
Random();
    //show questions
    showQuestion();
}

//method to show questions
private void showQuestion() {

    questionsTextArea.setText(ch6questions[currentQuestio
n][0]);
    String[] answers =
    Arrays.copyOfRange(ch6questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }

    }
    //shuffle options
    private void shuffleAnswers(String[]
answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
}

```

```

        }
        // method to check answer
    private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch6questions[currentQuestion][1])) {

        currentQuestion++;
        scores++;
        //show the result if
the user is done on answering
        if
(currentQuestion >= ch6questions.length) {

            Result_SD.Result_ch6SD fCh6sd = new
Result_SD().new Result_ch6SD();
            setVisible(false);

            fCh6sd.setVisible(true);
        } else {
            //continue to
show question
            showQuestion();
        }
    } else {

        currentQuestion++;
        if(currentQuestion > 9) {

            Result_SD.Result_ch6SD fCh6sd = new
Result_SD().new Result_ch6SD();
            setVisible(false);
            fCh6sd.setVisible(true);
            SDquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
        else {

            showQuestion();
        }
    }
}

```

```

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.util.Arrays;
import java.util.Random;

```

Operating system quiz button

```

public class OSquiz extends JFrame {

    private JPanel contentPane;
    public static Clip clip1;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        OSquiz frame = new OSquiz();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public OSquiz() {
        // Set the icon image of the frame

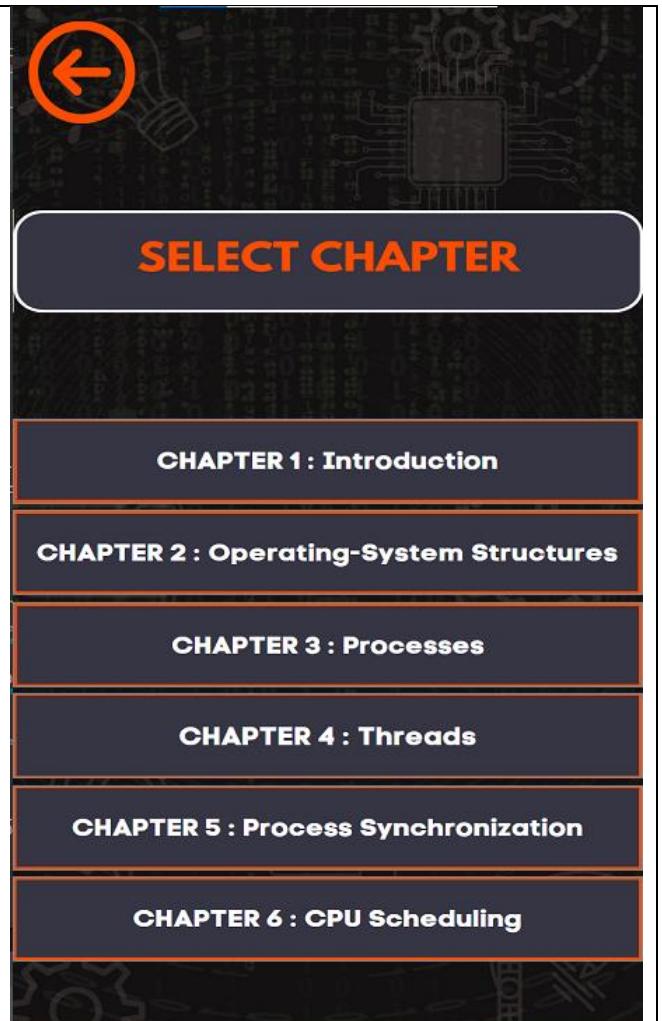
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseEntered(MouseEvent e)
            {
                backbtn.setIcon(new ImageIcon(quizmenu.class.getResource("/images/backbuttonhover.png")));
            }
            @Override
            public void mouseExited(MouseEvent e)
            {
                backbtn.setIcon(new ImageIcon(quizmenu.class.getResource("/images/backbutton.png")));
            }
            @Override
            public void mouseClicked(MouseEvent e)
            {
                quizmenu fquizmenu = new quizmenu();
                setVisible(false);
            }
        });
    }
}

```



```

        fquizmenu.setVisible(true);
    }
});

backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch1_OS fCh1_OS = new
ch1_OS();
        setVisible(false);
        fCh1_OS.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch1btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch1.png")));
ch1btn.setBounds(0, 295, 450, 61);
contentPane.add(ch1btn);

JButton ch2btn = new JButton("");
ch2btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch2_OS fCh2_OS = new
ch2_OS();
        setVisible(false);
        fCh2_OS.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch2btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch2.png")));
ch2btn.setBounds(0, 360, 450, 61);
contentPane.add(ch2btn);

JButton ch3btn = new JButton("");

```

```

ch3btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch3_OS fCh3_OS = new
ch3_OS();
        setVisible(false);
        fCh3_OS.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch3btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch3.png")));
ch3btn.setBounds(0, 426, 450, 61);
contentPane.add(ch3btn);

JButton ch4btn = new JButton("");
ch4btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch4_OS fCh4_OS = new
ch4_OS();
        setVisible(false);
        fCh4_OS.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch4btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch4.png")));
ch4btn.setBounds(0, 491, 450, 61);
contentPane.add(ch4btn);

JButton ch5btn = new JButton("");
ch5btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch5_OS fCh5_OS = new
ch5_OS();
        setVisible(false);
        fCh5_OS.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch5btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch5.png")));
ch5btn.setBounds(0, 557, 450, 61);
contentPane.add(ch5btn);

```

```

        JButton ch6btn = new JButton("");
        ch6btn.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent
e) {
                ch6_OS fCh6_OS = new
                ch6_OS();
                setVisible(false);
                fCh6_OS.setVisible(true);
                Home.clip.stop();

                playSound("sounds/quizsound.wav");
            }
        });
        ch6btn.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/OSch6.png")));
        ch6btn.setBounds(0, 622, 450, 61);
        contentPane.add(ch6btn);

        JLabel OSquizlabel = new JLabel("");
        OSquizlabel.setIcon(new
ImageIcon(OSquiz.class.getResource("/images/bg.png")));
        OSquizlabel.setBounds(0, 0, 450, 730);
        contentPane.add(OSquizlabel);
    }
}

```

```

class ch1_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
    JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][][] ch1questions = {
        {"It request to the OS to allow
user to wait for I/O \r\n"
            + "completion.",",
        "system call", "system
application", "system software", "system hardware"},,
        {"One or more CPUs, device
controllers connect \r\n"
            + "through
common bus providing access to shared \r\n"
            + "memory.",,
        "Computer-system operation", "Computer-system application",
        "Computer-system hardware",
        "Computer-system structure"},,
        {"It is available via Internet to
anyone willing to \r\n"
            + "pay.", "Public
cloud", "Local cloud",,
    }
}

```

```

        "Hybrid cloud", "Private
cloud"},

        {"It is used when source CPU
type different from target\r\n"
         + "type (i.e.
PowerPC to Intel x86).", "Emulation", "Virtualization", "Cloud
Computing", "Embedded system"},

        {"Any mechanism for controlling
access of processes \r\n"
         + "or users to
resources defined by the OS.", "Protection", "Security",
"Firewall", "Privilege
escalation"},

        {"It allows user to change to
effective ID with more \r\n"
         + "rights.",

"Privilege escalation", "Security", "Firewall", "Protection"},

        {"Is a software generated
interrupt caused either by\r\n"
         + "an error or a
user request.", "trap", "error", "polling","vectored"},

        {"It contains entry for each I/O
device indicating \r\n"
         + "its type,
address, and state.", "device-status table", "device-state table",
"device-type table", "device-address table"},

        {"It has one program counter
specifying location of \r\n"
         + "next
instruction to execute.", "Single-threaded process", "Multi-
threaded process", "Dual-threaded process","Multiplexing"},

        {"Is logical extension in which
CPU switches jobs so \r\n"
         + "frequently
that users can interact with each job \r\n"
         + "while it is

running, creating interactive
computing.", "Timesharing", "Multiprogramming", "Multiprocessing",
"Scheduling"},

};

public ch1_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);
}

```

```

JLabel ch1label = new JLabel("");
ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch1.png")));
ch1label.setBounds(0, 116, 450, 64);
contentPane.add(ch1label);

JLabel quiznolabel = new JLabel("Quiz:
10 items");

quiznolabel.setForeground(Color.WHITE);
quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
quiznolabel.setBounds(10, 191, 132, 29);
contentPane.add(quiznolabel);

JLabel lblNewLabel = new
JLabel("_____");
lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
contentPane.add(lblNewLabel);

JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer(( JButton) e.getSource());
        }
    });
}
contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);
questionsTextArea.setForeground(Color.WHITE);

```

```

        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {
        questionsTextArea.setText(ch1questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch1questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
        if
(button.getText().equals(ch1questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done
on answering
    }
}

```

```

        if (currentQuestion >=
ch1questions.length) {
    Result_OS.Result_ch1OS
fCh1OS = new Result_OS().new Result_ch1OS();
    setVisible(false);
    fCh1OS.setVisible(true);
} else {
    //continue to show question
    showQuestion();
}
} else {
    currentQuestion++;
    if(currentQuestion > 9) {

        Result_OS.Result_ch1OS fCh1OS = new
Result_OS().new Result_ch1OS();
        setVisible(false);
        fCh1OS.setVisible(true);
        OSquiz.clip1.stop();

        playSoundNotLoop("sounds/resultbgmusic.wav");
    }
    else {
        showQuestion();
    }
}

}

//to play loop sound
public static void playSound(String
fileName) {
    try {
        File soundFile = new File(fileName);
        AudioInputStream audioInputStream =
AudioSystem.getAudioInputStream(soundFile);
        clip1 = AudioSystem.getClip();
        clip1.open(audioInputStream);
        clip1.start();
        clip1.loop(Clip.LOOP_CONTINUOUSLY);
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
}

//method to play sound
public static void playSoundNotLoop(String
fileName) {
    try {
        File soundFile = new File(fileName);
        AudioInputStream audioInputStream =
AudioSystem.getAudioInputStream(soundFile);
        Clip clip = AudioSystem.getClip();
        clip.open(audioInputStream);
        clip.start();
    } catch (Exception e) {

```

```

        System.err.println(e.getMessage());
    }
}

class ch1_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch1questions = {
        {"It request to the OS to allow user to wait for I/O \r\n" +
         + "completion.", "system call", "system application", "system software", "system hardware"},

        {"One or more CPUs, device controllers connect \r\n" +
         + "through common bus providing access to shared \r\n" +
         + "memory."},

        {"It is available via Internet to anyone willing to \r\n" +
         + "pay.", "Public cloud", "Local cloud", "Hybrid cloud", "Private cloud"},

        {"It is used when source CPU type different from target\r\n" +
         + "type (i.e. PowerPC to Intel x86).", "Emulation", "Virtualization", "Cloud Computing", "Embedded system"},

        {"Any mechanism for controlling access of processes \r\n" +
         + "or users to resources defined by the OS.", "Protection", "Security", "Firewall", "Privilege escalation"},

        {"It allows user to change to effective ID with more \r\n" +
         + "rights."},

        {"Is a software generated interrupt caused either by\r\n" +
         + "an error or a user request.", "trap", "error", "polling", "vectored"},

        {"It contains entry for each I/O device indicating \r\n"}
    };
}

```

## Chapter 1 Operating System quiz

**CHAPTER 1 : Introduction**

**Quiz: 10 items**

It request to the OS to allow user to wait for I/O completion.

system software

system call

system hardware

system application

```

+ "its type,
address, and state.", "device-status table", "device-state table",
"device-type table", "device-address table"},
{"It has one program counter
specifying location of \r\n"
+ "next
instruction to execute.", "Single-threaded process", "Multi-
threaded process", "Dual-threaded process", "Multiplexing",
 {"Is logical extension in which
CPU switches jobs so \r\n"
+ "frequently
that users can interact with each job \r\n"
+ "while it is
running, creating interactive
computing.", "Timesharing", "Multiprogramming", "Multiprocessing",
"Scheduling"),
};

public ch1_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch1.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("____")
;

    lblNewLabel.setForeground(Color.WHITE);
    lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));

```

```

29);
        lblNewLabel.setBounds(10, 202, 440,
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();
        questionsTextArea.setEditable(false);

        questionsTextArea.setForeground(Color.WHITE);
        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
}

```

```

//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch1questions[currentQuestion][0]);
    String[] answers =
    Arrays.copyOfRange(ch1questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }

}

//method to randomize choices
void shuffleAnswers(String[] answers) {
    for (int i = 0; i < answers.length; i++) {
        int j =
random_ans.nextInt(answers.length);
        String temp = answers[i];
        answers[i] = answers[j];
        answers[j] = temp;
    }
}

// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch1questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch1questions.length) {
            Result_OS.Result_ch1OS
fCh1OS = new Result_OS().new Result_ch1OS();
            setVisible(false);
            fCh1OS.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

Result_OS.Result_ch1OS fCh1OS = new
Result_OS().new Result_ch1OS();
            setVisible(false);
            fCh1OS.setVisible(true);
            OSquiz.clip1.stop();

playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    }
}

```

<pre>         else {             showQuestion();         }      } } </pre>	
<pre> class ch2_OS extends JFrame {      private JPanel contentPane;     private JButton[] answerButtons = new JButton[4];     private JTextArea questionsTextArea;     private int currentQuestion;     private Random random_ans;     public static int scores;     //question and choices in an array     String[][] ch2questions = {         {"What type of user interface is most commonly used?", "Graphical user interface", "Command-line interface", "Batch interface", "Menu-driven interface"},          {"What is one function that the operating system must \r\n", "+ \"provide for I/O devices?\", \"A means to do I/O\", \"Full control of I/O devices\", \"Open access to I/O devices\", \"Restrictions on I/O device usage\"},          {"What is one of the functions of an operating system \r\n", "+ \"for systems with multiple users?\", \"Resource allocation\", \"Debugging error codes\", "Running error-causing processes", "Assisting the user with system operation"},          {"Which of the following is a shell available on UNIX \r\n", "+ \"and Linux?\", \"C shell\", \"Golang shell\", \"Python shell\", \"JavaScript shell\"},          {"Which type of interface is typically used on \r\n", "+ \"smartphones and handheld tablet computers?\", \"Touchscreen\", \"Command-line\", "Keyboard", "Mouse"},          {"Which of the following is NOT a category of system \r\n", "+ \"programs?\", \"Spreadsheets\", \"Programming language support\", \"File manipulation\", \"Background services\"},          {"What system is used to store and retrieve \r\n", "+ \"configuration information?\", \"Registry\", \"Database\", \"Cache\", \"File system\"},     } } </pre>	<p>Chapter 2 Operating System quiz</p> <p><b>CHAPTER 2 : Operating-System Structures</b></p> <p><b>Quiz: 15 items</b></p> <p><b>What type of user interface is most commonly used?</b></p> <ul style="list-style-type: none"> <li>Command-line interface</li> <li>Graphical user interface</li> <li>Batch interface</li> <li>Menu-driven interface</li> </ul>

```

        {"What are services or
subsystems known as?", "Daemons", "Processes", "Kernels",
"Applications"},

        {"Which modern operating
system combines multiple \r\n"
         + "approaches
to address performance, security, and \r\n"
         + "usability
needs?", "Hybrid", "Monolithic", "Layered", "Microkernel"},

        {"What is the highest layer in the
UNIX operating \r\n"
         +
"system?", "The user", "The file system", "The kernel", "The system
call interface"},

        {"Developed by Open Handset
Alliance (mostly Google)", "Android", "iOS", "MAC OS", "Linux"},

        {"Limited by hardware
functionality, the original _____\r\n"
         + "operating
system had limited structuring.", "UNIX", "Android", "iOS", "MAC
OS"},

        {"Written to provide the most
functionality in the least\r\n"
         + "space.", "MS-
DOS", "UNIX", "MACH", "iOS"},

        {"It allow an OS to run on non
native hardware.", "Emulation", "Virtualization", "Cloud
Computing", "Traditional"},

        {"Specifying and designing an
OS is highly creative task of\r\n"
         +
"_____.", "software engineer", "software
manager", "project manager", "programmer"}
};

public ch2_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch2.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);
}

```

```
JLabel quiznolabel = new JLabel("Quiz:  
15 items");  
  
    quiznolabel.setForeground(Color.WHITE);  
    quiznolabel.setFont(new Font("Arial",  
Font.BOLD, 18));  
    quiznolabel.setBounds(10, 191, 132, 29);  
    contentPane.add(quiznolabel);  
  
    JLabel lblNewLabel = new  
JLabel("_____")  
);  
  
    lblNewLabel.setForeground(Color.WHITE);  
    lblNewLabel.setFont(new  
Font("Copperplate Gothic Bold", Font.BOLD, 17));  
    lblNewLabel.setBounds(10, 202, 440,  
29);  
    contentPane.add(lblNewLabel);  
  
    JPanel buttonPanel = new JPanel();  
    buttonPanel.setOpaque(false);  
    buttonPanel.setBackground(new  
Color(0, 0, 0));  
    buttonPanel.setBounds(36, 387, 367,  
290);  
    buttonPanel.setLayout(new  
GridLayout(4, 1, 6, 15));  
    //add buttons  
    answerButtons = new RoundButton[4];  
    for (int i = 0; i < 4; i++) {  
        answerButtons[i] = new RoundButton();  
        answerButtons[i].setBackground(new  
Color(51,71,86));  
        answerButtons[i].setBorder(null);  
        answerButtons[i].setForeground(Color.white);  
        buttonPanel.add(answerButtons[i]);  
        answerButtons[i].addActionListener(new  
ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                checkAnswer(( JButton ) e.getSource());  
            }  
});  
    }  
    contentPane.add(buttonPanel);  
    questionsTextArea = new JTextArea();  
    questionsTextArea.setEditable(false);  
  
    questionsTextArea.setForeground(Color.WHITE);  
    questionsTextArea.setFont(new  
Font("Arial", Font.BOLD, 17));  
    questionsTextArea.setOpaque(false);  
    questionsTextArea.setBounds(10, 242,  
430, 143);  
    contentPane.add(questionsTextArea);
```

```

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {

        questionsTextArea.setText(ch2questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch2questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }

        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
        if
(button.getText().equals(ch2questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done
            on answering
            if (currentQuestion >=
ch2questions.length) {
                Result_OS.Result_ch2OS
fCh2OS = new Result_OS().new Result_ch2OS();
                setVisible(false);
                fCh2OS.setVisible(true);
            }
        }
    }
}

```

```

        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 14) {

            Result_OS.Result_ch2OS fCh2OS = new
Result_OS().new Result_ch2OS();
            setVisible(false);
            fCh2OS.setVisible(true);
            OSquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    }
}
}

```

```

class ch3_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch3questions = {
        {"In which state is a process
waiting for some event \r\n",
         + "to occur?", "Waiting", "Terminated",
         "Ready", "Running"},

        {"Which scheduler selects which
process should be \r\n",
         + "executed next
and allocates CPU?", "Medium-term scheduler", "Long-term
scheduler", "Short-term scheduler", "No
scheduler is responsible"},

        {"Which mobile operating system
allows apps to run in \r\n",
         + "the
background with fewer limits?", "Android", "Blackberry",
         "Windows Phone",
         "iOS"},

        {"What is a mechanism that the
system must provide for \r\n"
    }
}

```

Chapter 3 Operating System quiz

```

+ "processes?",  

"Process creation and termination", "Multiprocessing", "Hardware  

support", "Context-switch time"},  

        {"Which system call is used to  

create a new process in \r\n"  

+ "UNIX?",  

"fork()", "del()",  

"terminate()",  

        {"In indirect communication, how  

can processes \r\n"  

+  

"communicate?", "If they share a common mailbox/port",  

"Through physical mail", "Through social media", "Through phone  

calls"},  

        {"What are the two operations  

provided by IPC facility \r\n"  

+ "for  

communication between processes?", "send(message) and  

receive(message)", "read(message) and write(message)",  

"create(message) and delete(message)", "respond(message) and  

request(message)"},  

        {"Which class in Java allows  

data to be sent to multiple \r\n"  

+ "recipients?",  

"MulticastSocket", "ServerSocket", "Socket", "DatagramSocket"},  

        {"What system provides named  

pipes?", "Both UNIX and Windows", "Only Windows", "Only  

UNIX", "Linux"},  

        {"Is communication with ordinary  

pipes unidirectional \r\n"  

+ "or  

bidirectional?", "Unidirectional", "Bidirectional", "Neither", "Both"},  

        };  

public ch3_OS() {  

    // Set the icon image of the frame  

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g  

etResource("/images/logo.png")));  

    setUndecorated(true);  

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  

    setBounds(100, 100, 450, 730);  

    setLocationRelativeTo(null);  

    setResizable(false);  

    contentPane = new JPanel();  

    setContentPane(contentPane);  

    contentPane.setLayout(null);  

JLabel ch1label = new JLabel("");  

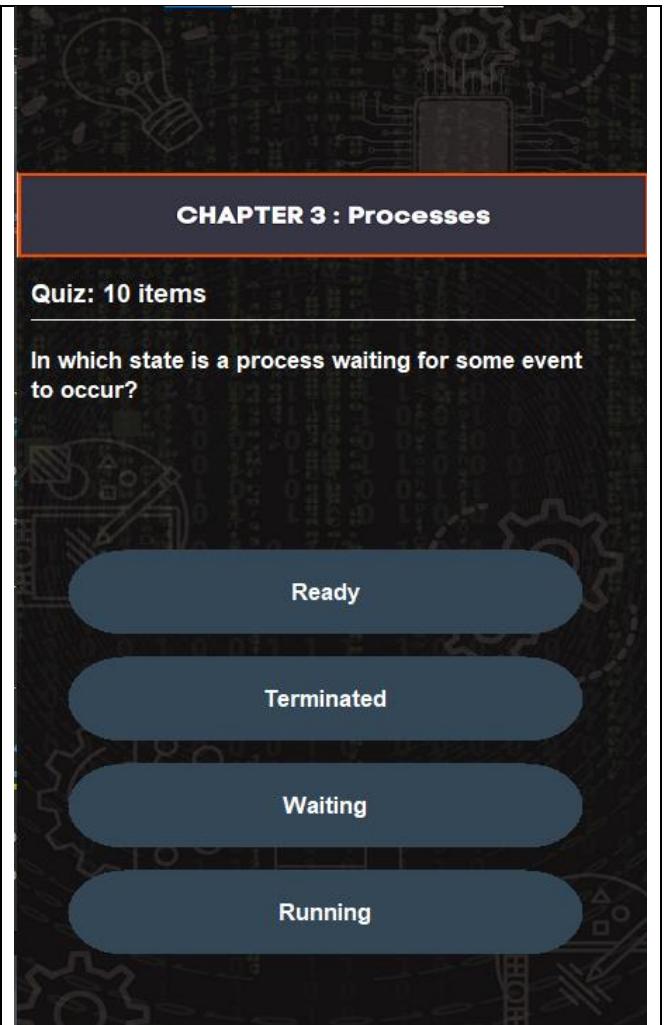
    ch1label.setIcon(new  

ImageIcon(Home.class.getResource("/images/OSch3.png")));  

    ch1label.setBounds(0, 116, 450, 64);  

    contentPane.add(ch1label);

```



## CHAPTER 3 : Processes

**Quiz: 10 items**

In which state is a process waiting for some event to occur?

Ready

Terminated

Waiting

Running

```
JLabel quiznolabel = new JLabel("Quiz:  
10 items");  
  
    quiznolabel.setForeground(Color.WHITE);  
    quiznolabel.setFont(new Font("Arial",  
Font.BOLD, 18));  
    quiznolabel.setBounds(10, 191, 132, 29);  
    contentPane.add(quiznolabel);  
  
    JLabel lblNewLabel = new  
JLabel("_____")  
);  
  
    lblNewLabel.setForeground(Color.WHITE);  
    lblNewLabel.setFont(new  
Font("Copperplate Gothic Bold", Font.BOLD, 17));  
    lblNewLabel.setBounds(10, 202, 440,  
29);  
    contentPane.add(lblNewLabel);  
  
    JPanel buttonPanel = new JPanel();  
    buttonPanel.setOpaque(false);  
    buttonPanel.setBackground(new  
Color(0, 0, 0));  
    buttonPanel.setBounds(36, 387, 367,  
290);  
    buttonPanel.setLayout(new  
GridLayout(4, 1, 6, 15));  
    //add buttons  
    answerButtons = new RoundButton[4];  
    for (int i = 0; i < 4; i++) {  
        answerButtons[i] = new RoundButton();  
        answerButtons[i].setBackground(new  
Color(51,71,86));  
        answerButtons[i].setBorder(null);  
        answerButtons[i].setForeground(Color.white);  
        buttonPanel.add(answerButtons[i]);  
        answerButtons[i].addActionListener(new  
ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                checkAnswer((JButton) e.getSource());  
            }  
        });  
    }  
    contentPane.add(buttonPanel);  
    questionsTextArea = new JTextArea();  
    questionsTextArea.setEditable(false);  
  
    questionsTextArea.setForeground(Color.WHITE);  
    questionsTextArea.setFont(new  
Font("Arial", Font.BOLD, 17));  
    questionsTextArea.setOpaque(false);  
    questionsTextArea.setBounds(10, 242,  
430, 143);  
    contentPane.add(questionsTextArea);
```

```

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {

        questionsTextArea.setText(ch3questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch3questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }

        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
        if
(button.getText().equals(ch3questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done
            on answering
            if (currentQuestion >=
ch3questions.length) {
                Result_OS.Result_ch3OS
fCh3OS = new Result_OS().new Result_ch3OS();
                setVisible(false);
                fCh3OS.setVisible(true);
            }
        }
    }
}

```

```

        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

            Result_OS.Result_ch3OS fCh3OS = new
Result_OS().new Result_ch3OS();
            setVisible(false);
            fCh3OS.setVisible(true);
            OSquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
        else {
            showQuestion();
        }
    }
}

class ch4_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch4questions = {
        {"Which of the following is NOT
an advantage of using \r\n",
         + "threads in an
application?", "Process creation is light-
weight", "Resource sharing is easier", "Thread switching has
lower overhead", "Efficiency can be increased"},

        {"Which type of thread library is
supported by the \r\n",
         + "Kernel?", "Kernel threads", "POSIX threads",
         "Windows threads", "Java
threads"},

        {"Which system allows many
user level threads to be \r\n",
         + "mapped to
many kernel threads?", "Solaris 9 and later", "Windows with the
ThreadFiber package", "GNU Portable
Threads", "Solaris prior to version 9"},

        {"What is a POSIX standard API
for thread creation \r\n"}
    }
}

```

## Chapter 4 Operating System quiz

```

+ "and
synchronization?", "IEEE 1003.1c", "HTML", "IRC", "FTP"},
        {"What is the mechanism used to
manage most of the \r\n"
+ "details of
threading in thread pools?", "Dispatch queue", "Block-based
concurrent programming",
        "Thread identification",
        "Windows API"},

        {"What are the two types of
dispatch queues?", "serial and concurrent", "signal handling and
scheduler activations", "system wide and process",
"asynchronous and synchronous"},

        {"On Linux systems, how is
thread cancellation \r\n"
+ "handled?",
"Through signals", "Through interrupts", "Using a special cancel
function", "Through exceptions"},

        {"Which primary API does
Windows implement?", "Windows API", "Android API", "Linux
API", "Mac API"},

        {"In Linux, what system call is
used for creating \r\n"
+ "a new
thread?", "clone()", "exec()", "wait()", "fork()",
        {"It distributes subsets of the
same data across \r\n"
+ "multiple
cores, same operation on each.", "Data parallelism", "Task
parallelism", "Memory parallelism", "hardware parallelism"},

};

public ch4_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch4.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");
}

```

## CHAPTER 4 : Threads

**Quiz: 10 items**

Which of the following is NOT an advantage of using threads in an application?

- Efficiency can be increased
- Process creation is light-weight
- Resource sharing is easier
- Thread switching has lower overhead

```

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();
        questionsTextArea.setEditable(false);

        questionsTextArea.setForeground(Color.WHITE);
        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");

```

```

        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {
        questionsTextArea.setText(ch4questions[currentQuestio
n][0]);
        String[] answers =
Arrays.copyOfRange(ch4questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
        if
(button.getText().equals(ch4questions[currentQuestion][1])) {
            currentQuestion++;
            scores++;
            //show the result if the user is done
on answering
            if (currentQuestion >=
ch4questions.length) {
                Result_OS.Result_ch4OS
fCh4OS = new Result_OS().new Result_ch4OS();
                setVisible(false);
                fCh4OS.setVisible(true);
            } else {
                //continue to show question
            }
        }
    }
}

```

```

        showQuestion();
    }
} else {
    currentQuestion++;
    if(currentQuestion > 9) {

        Result_OS.Result_ch4OS fCh4OS = new
Result_OS().new Result_ch4OS();
        setVisible(false);
        fCh4OS.setVisible(true);
        OSquiz.clip1.stop();

        playSoundNotLoop("sounds/resultbgmusic.wav");
    }
    else {
        showQuestion();
    }
}

}
class ch5_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch5questions = {
        {"Which mechanism is required
to maintain data \r\n"
         + "consistency
in cooperating processes?", "Mechanisms for orderly
execution", "Mutex", "Mutual exclusion", "Semaphore"},

        {"What is a synchronization tool
that does not require \r\n"
         + "busy
waiting?", "not provided", "turn variable",
         "atomic hardware instructions",
         "flag array"},

        {"What is the other name for a
binary semaphore?", "mutex lock", "counting semaphore",
         "atomic operation",
         "critical section"},

        {"What is the problem faced
when multiple readers need \r\n"
         + "to access a
shared data set concurrently?", "Allowing them to read at the
same time", "Deadlock between the readers", "Priority Inversion
between the processes", "Starvation of the writers"},

    }
}

```

```

        {"What is the purpose of
Semaphore mutex?", "Controls access to readcount", "Controls
writer access",
                     "Controls access to
multiple writers", "Controls access to shared data"},
                     {"What type of synchronization
mechanism is used by \r\n"
                     + "Windows XP
on uniprocessor systems?", "Interrupt masks", "Semaphores",
"Pthreads", "Spinlocks"},

                     {"What do concurrency-control
algorithms provide?", "Serializability", "Flexibility",
"Concurrency", "Efficiency"},

                     {"What is the main purpose of
using locks in the \r\n"
                     + "database
transaction?", "Access control", "Data validation", "Data
compression", "Encryption"},

                     {"It could disable interrupts.",
"Uniprocessors", "Single processor", "Multiprocessors", "Dual
processors"},

                     {"The integer value can range
over an unrestricted \r\n"
                     + "domain.", "Counting semaphore", "Binary semaphore", "Mutex
locks", "Semaphore mutex"},

};

public ch5_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch5.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
}

```

```

        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        "


lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer((JButton) e.getSource());
        }
    });
}
        contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);

```

```

//initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {

        questionsTextArea.setText(ch5questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch5questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }

        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch5questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch5questions.length) {
            Result_OS.Result_ch5OS
fCh5OS = new Result_OS().new Result_ch5OS();
            setVisible(false);
            fCh5OS.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

```

```

        Result_OS.Result_ch5OS fCh5OS = new
Result_OS().new Result_ch5OS();
        setVisible(false);
fCh5OS.setVisible(true);
OSquiz.clip1.stop();

playSoundNotLoop("sounds/resultbgmusic.wav");
    }
else {
        showQuestion();
    }

}
}

class ch4_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
//question and choices in an array
String[][] ch4questions = {
        {"Which of the following is NOT
an advantage of using \r\n"
            + "threads in an
application?", "Process creation is light-
weight", "Resource sharing is easier", "Thread switching has
lower overhead", "Efficiency can be increased"},

        {"Which type of thread library is
supported by the \r\n"
            + "Kernel?", "Kernel threads", "POSIX threads",
            "Windows threads", "Java
threads"},

        {"Which system allows many
user level threads to be \r\n"
            + "mapped to
many kernel threads?", "Solaris 9 and later", "Windows with the
ThreadFiber package", "GNU Portable
Threads", "Solaris prior to version 9"},

        {"What is a POSIX standard API
for thread creation \r\n"
            + "and
synchronization?", "IEEE 1003.1c", "HTML", "IRC", "FTP"},

        {"What is the mechanism used to
manage most of the \r\n"
            + "details of
threading in thread pools?", "Dispatch queue", "Block-based
concurrent programming",}
}

```

## Chapter 5 Operating System quiz

**CHAPTER 4 : Threads**

**Quiz: 10 items**

Which of the following is NOT an advantage of using threads in an application?

- Efficiency can be increased
- Thread switching has lower overhead
- Process creation is light-weight
- Resource sharing is easier

```

        "Thread identification",
"Windows API"},

        {"What are the two types of
dispatch queues?", "serial and concurrent", "signal handling and
scheduler activations", "system wide and process",
"asynchronous and synchronous"},

        {"On Linux systems, how is
thread cancellation \r\n"
            + "handled?"}

"Through signals", "Through interrupts", "Using a special cancel
function", "Through exceptions"},

        {"Which primary API does
Windows implement?", "Windows API", "Android API", "Linux
API", "Mac API"},

        {"In Linux, what system call is
used for creating \r\n"
            + "a new
thread?", "clone()", "exec()", "wait()", "fork()",
            {"It distributes subsets of the
same data across \r\n"
                + "multiple
cores, same operation on each.", "Data parallelism", "Task
parallelism", "Memory parallelism", "hardware parallelism"},

};

public ch4_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch4.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);
}

```

```

JLabel lblNewLabel = new
JLabel("_____");
lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
contentPane.add(lblNewLabel);

JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer((JButton) e.getSource());
        }
    });
}
contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
contentPane.add(questionsTextArea);

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);

//initial values
currentQuestion = 0;

```

```

        scores = 0;
        //assign random
        random_ans = new
Random();
                //show questions
                showQuestion();
}
//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch4questions[currentQuestio
n][0]);
    String[] answers =
Arrays.copyOfRange(ch4questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }

}
//method to randomize choices
void shuffleAnswers(String[] answers) {
for (int i = 0; i < answers.length; i++) {
    int j =
random_ans.nextInt(answers.length);
    String temp = answers[i];
    answers[i] = answers[j];
    answers[j] = temp;
}
// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch4questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch4questions.length) {
            Result_OS.Result_ch4OS
fCh4OS = new Result_OS().new Result_ch4OS();
            setVisible(false);
            fCh4OS.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

```

```

Result_OS.Result_ch4OS fCh4OS = new
Result_OS().new Result_ch4OS();
    setVisible(false);
    fCh4OS.setVisible(true);
    OSquiz.clip1.stop();

    playSoundNotLoop("sounds/resultbgmusic.wav");
}
else {
    showQuestion();
}

}

}

class ch5_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
 JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch5questions = {
        {"Which mechanism is required
to maintain data \r\n",
         + "consistency
in cooperating processes?", "Mechanisms for orderly
execution", "Mutex", "Mutual exclusion", "Semaphore"},

        {"What is a synchronization tool
that does not require \r\n",
         + "busy
waiting?", "not provided", "turn variable",
         "atomic hardware instructions",
         "flag array"},

        {"What is the other name for a
binary semaphore?", "mutex lock", "counting semaphore",
         "atomic operation",
         "critical section"},

        {"What is the problem faced
when multiple readers need \r\n",
         + "to access a
shared data set concurrently?", "Allowing them to read at the
same time", "Deadlock between the readers", "Priority Inversion
between the processes", "Starvation of the writers"},

        {"What is the purpose of
Semaphore mutex?", "Controls access to readcount", "Controls
writer access",
         "Controls access to
multiple writers", "Controls access to shared data"},

        {"What type of synchronization
mechanism is used by \r\n"
    }
}

```

```

+ "Windows XP
on uniprocessor systems?", "Interrupt masks", "Semaphores",
"Pthreads", "Spinlocks"},

        {"What do concurrency-control
algorithms provide?", "Serializability", "Flexibility",
"Concurrency", "Efficiency"},

        {"What is the main purpose of
using locks in the \r\n"
+ "database
transaction?", "Access control", "Data validation", "Data
compression", "Encryption"},

        {"It could disable interrupts.",

"Uniprocessors", "Single processor", "Multiprocessors", "Dual
processors"},

        {"The integer value can range
over an unrestricted \r\n"
+
"domain.", "Counting semaphore", "Binary semaphore", "Mutex
locks", "Semaphore mutex"},

};

public ch5_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch5.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("____");
    lblNewLabel.setForeground(Color.WHITE);

```

```

        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
            answerButtons[i].addActionListener(new
ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    checkAnswer((JButton) e.getSource());
                }
            });
        }
        contentPane.add(buttonPanel);
        questionsTextArea = new JTextArea();
        questionsTextArea.setEditable(false);

        questionsTextArea.setForeground(Color.WHITE);
        questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
        questionsTextArea.setOpaque(false);
        questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);

        //initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
    }
}

```

```

        showQuestion();
    }
    //method to show questions
    void showQuestion() {

        questionsTextArea.setText(ch5questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch5questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }

        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch5questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch5questions.length) {
            Result_OS.Result_ch5OS
fCh5OS = new Result_OS().new Result_ch5OS();
            setVisible(false);
            fCh5OS.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

            Result_OS.Result_ch5OS fCh5OS = new
Result_OS().new Result_ch5OS();
            setVisible(false);
            fCh5OS.setVisible(true);
            OSquiz.clip1.stop();
        }
    }
}

```

```

    playSoundNotLoop("sounds/resultbgmusic.wav");
}
else {
    showQuestion();
}

}
}
}

```

```

class ch6_OS extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch6questions = {
        {"Which cycle constitutes a
process execution?", "CPU execution and I/O wait",
"CPU execution and memory write", "CPU execution and disk
read", "CPU execution and network wait"},

        {"What is the name of the
scheduling algorithm \r\n"
+ "for
foreground processes in the partitioned \r\n"
+ "ready
queue?", "RR", "FCFS", "SJF", "Priority"},

        {"What is the name of the
scheduling method used \r\n"
+ "between
foreground and background queues?", "Fixed priority
scheduling", "SJF scheduling", "FIFO scheduling"},

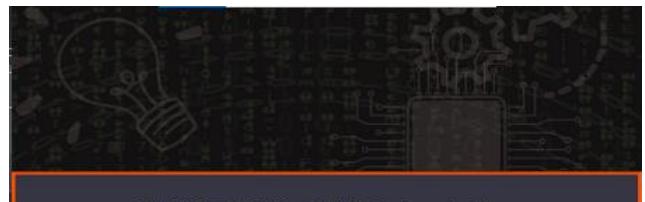
        {"What percentage of CPU time
is allocated to the \r\n"
+ "foreground
queue in RR scheduling?", "80%", "50%", "60%", "20%"},

        {"In which queue is a new job
initially served and \r\n"
+ "how much
time quantum does it receive?", "Q0 and 8 milliseconds", "Q1 and
16 milliseconds", "Q2 and 16 milliseconds"},

        {"What does SCS stand for in
multi-threading?", "System Contention Scope", "Software
Contention Scope", "Software Check Scope", "System Creation
System"}};

```

## Chapter 6 Operating System quiz



**CHAPTER 6 : CPU Scheduling**

**Quiz: 10 items**

**Which cycle constitutes a process execution?**

CPU execution and memory write

CPU execution and disk read

CPU execution and I/O wait

CPU execution and network wait

```

        {"What type of multiprocessing
allows only one \r\n"
+ "processor to
access system data structures?", "Asymmetric multiprocessing",
"Processor affinity", "Homogenous multiprocessing", "Symmetric
multiprocessing"},

        {"What is the range of real-time
priority in the \r\n"
+ "context of
multiple threads per core?", "0 to 99", "100 to 200", "0 to 100",
"100 to 140"},

        {"Time it takes for the dispatcher
to stop one \r\n"
+ "process and
start another running.", "Dispatch latency", "Turnaround
time", "Waiting time", "Response time"},

        {"Amount of time a process has
been waiting in the\r\n"
+ "ready
queue.", "Waiting time", "Turnaround time", "Dispatch
latency", "Response time"},

};

public ch6_OS() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/OSch6.png")));
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("_____"
);

```

```

lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
contentPane.add(lblNewLabel);

JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer((JButton) e.getSource());
        }
    });
}
contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
contentPane.add(questionsTextArea);

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);

//initial values
currentQuestion = 0;
scores = 0;
//assign random

```

```

        random_ans = new
Random();
                                //show questions
                showQuestion();
}
//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch6questions[currentQuestio
n][0]);
    String[] answers =
Arrays.copyOfRange(ch6questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }

}
//method to randomize choices
void shuffleAnswers(String[] answers) {
for (int i = 0; i < answers.length; i++) {
    int j =
random_ans.nextInt(answers.length);
    String temp = answers[i];
    answers[i] = answers[j];
    answers[j] = temp;
}
}

// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch6questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch6questions.length) {
            Result_OS.Result_ch6OS
fCh6OS = new Result_OS().new Result_ch6OS();
            setVisible(false);
            fCh6OS.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

Result_OS.Result_ch6OS fCh6OS = new
Result_OS().new Result_ch6OS();
            setVisible(false);
        }
    }
}

```

```

        fCh6OS.setVisible(true);
        OSquiz.clip1.stop();

        playSoundNotLoop("sounds/resultbgmusic.wav");
    }
    else {
        showQuestion();
    }

}
}

```

```

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.util.Arrays;
import java.util.Random;

public class MICROquiz extends JFrame {

    private JPanel contentPane;
    public static Clip clip1;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        MICROquiz frame = new MICROquiz();
        frame.setVisible(true);

    }

    /**
     * Create the frame.
     */
    public MICROquiz() {
        // Set the icon image of the frame

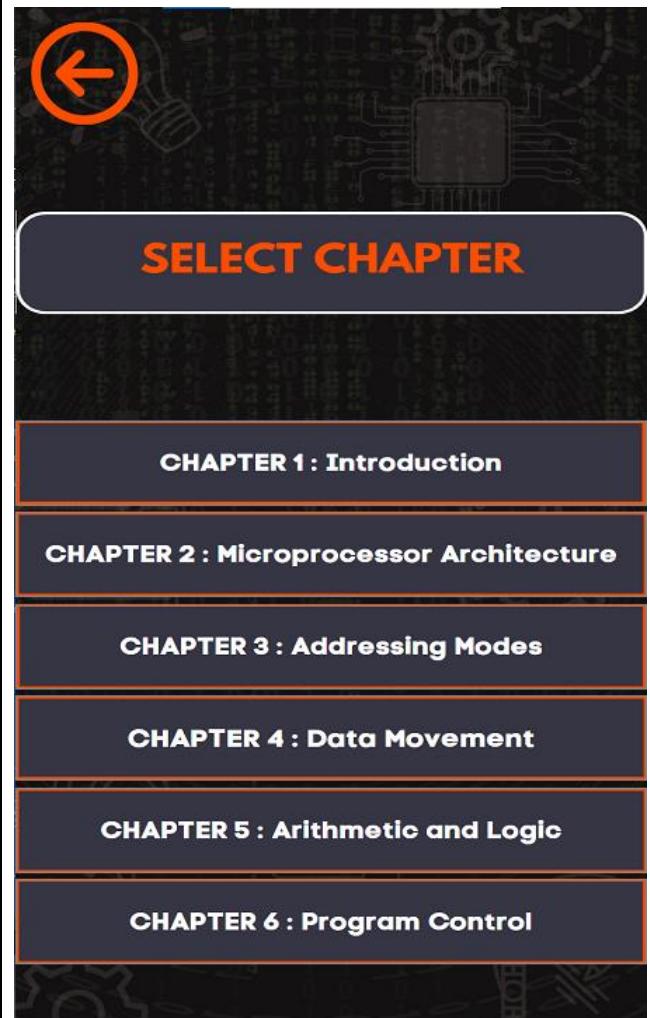
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);

```

Microprocess quiz button



```

backbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        backbtn.setIcon(new ImageIcon(MICROquiz.class.getResource("/images/backbuttonhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        backbtn.setIcon(new ImageIcon(MICROquiz.class.getResource("/images/backbutton.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e)
    {
        quizmenu fquizmenu = new quizmenu();
        fquizmenu.setVisible(false);
        fquizmenu.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new ImageIcon(MICROquiz.class.getResource("/images/backbutton.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new ImageIcon(MICROquiz.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e)
    {
        ch1_MICRO fCh1_MICRO =
new ch1_MICRO();
        fCh1_MICRO.setVisible(false);
        fCh1_MICRO.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch1btn.setIcon(new ImageIcon(MICROquiz.class.getResource("/images/OSch1.png")));

```

```

ch1btn.setBounds(0, 295, 450, 61);
contentPane.add(ch1btn);

JButton ch2btn = new JButton("");
ch2btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch2_MICRO fCh2_MICRO =
new ch2_MICRO();
        setVisible(false);
        fCh2_MICRO.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch2btn.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/MICROch2.p
ng")));
ch2btn.setBounds(0, 360, 450, 61);
contentPane.add(ch2btn);

JButton ch3btn = new JButton("");
ch3btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch3_MICRO fCh3_MICRO =
new ch3_MICRO();
        setVisible(false);
        fCh3_MICRO.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});
ch3btn.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/MICROch3.p
ng")));
ch3btn.setBounds(0, 426, 450, 61);
contentPane.add(ch3btn);

JButton ch4btn = new JButton("");
ch4btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch4_MICRO fCh4_MICRO =
new ch4_MICRO();
        setVisible(false);
        fCh4_MICRO.setVisible(true);
        Home.clip.stop();

        playSound("sounds/quizsound.wav");
    }
});

```

```

        });
        ch4btn.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/MICROch4.p
ng")));
        ch4btn.setBounds(0, 491, 450, 61);
        contentPane.add(ch4btn);

        JButton ch5btn = new JButton("");
        ch5btn.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent
e) {
                ch5_MICRO fCh5_MICRO =
new ch5_MICRO();
                setVisible(false);
                fCh5_MICRO.setVisible(true);
                Home.clip.stop();

                playSound("sounds/quizsound.wav");
            }
        });
        ch5btn.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/MICROch5.p
ng")));
        ch5btn.setBounds(0, 557, 450, 61);
        contentPane.add(ch5btn);

        JButton ch6btn = new JButton("");
        ch6btn.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent
e) {
                ch6_MICRO fCh6_MICRO =
new ch6_MICRO();
                setVisible(false);
                fCh6_MICRO.setVisible(true);
                Home.clip.stop();

                playSound("sounds/quizsound.wav");
            }
        });
        ch6btn.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/MICROch6.p
ng")));
        ch6btn.setBounds(0, 622, 450, 61);
        contentPane.add(ch6btn);

        JLabel MICROQuizlabel = new JLabel("");
        MICROQuizlabel.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/bg.png")));
        MICROQuizlabel.setBounds(0, 0, 450, 730);
        contentPane.add(MICROQuizlabel);
    }

//to play loop sound
    public static void playSound(String fileName) {
        try {

```

```

        File soundFile = new File(fileName);
        AudioInputStream audioInputStream =
    AudioSystem.getAudioInputStream(soundFile);
        clip1 = AudioSystem.getClip();
        clip1.open(audioInputStream);
        clip1.start();
        clip1.loop(Clip.LOOP_CONTINUOUSLY);
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
}

//to play sound
public static void
playSoundNotLoop(String fileName) {
    try {
        File soundFile = new File(fileName);
        AudioInputStream audioInputStream =
    AudioSystem.getAudioInputStream(soundFile);
        Clip clip = AudioSystem.getClip();
        clip.open(audioInputStream);
        clip.start();
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
}
}

```

```

class ch1_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch1questions = {
        {"What was the first mechanical
calculator?", "Abacus", "Slide rule",
"Analytical engine", "Odometer"},

        {"Which programming language
is named after Blaise \r\n"
+ "Pascal?", "Pascal", "Analytical Engine",
"Jacquard", "Ada"},

        {"What was the first modern
electromechanical \r\n"
+ "computer
system, constructed by Konrad Zuse?", "The Z3", "The Z4",
"The Z5", "The Z6"},

        {"When was the first electronic
computer, Colossus, \r\n"
+ "placed into
operation?", "1943", "1944", "1945", "1946"},

    }
}

```

Chapter 1 Microprocessor quiz

{"Which of the following companies developed a line \r\n" + "of machines that used punched cards for tabulation?", "IBM", "Monroe", "Bomar Corporation", "Bell Labs"}, {"The first handheld electronic calculator was \r\n" + "developed by:", "Bomar Corporation", "IBM", "Bell Labs", "Intel Corporation"}, {"Who developed the punched card for storing data?", "Herman Hollerith", "Michael Faraday", "Blaise Pascal", "Jacquard"}, {"What was the name of the first general-purpose, \r\n" + "programmable electronic computer system?", "ENIAC", "Colossus", "Z3", "The Bomar Brain"}, {"Which microprocessor was an updated version of \r\n" + "the Intel 4004?", "Intel 4040", "Intel 8008", "Intel 8080", "Motorola MC6800"}, {"When was the first 8-bit microprocessor released?", "1973", "1983", "1971", "1978"}, {"What device did Intel release in 1983 as an updated \r\n" + "version of the 8086 microprocessor?", "80286", "80285", "80386", "80486"}, {"What is the speed of execution of instructions in \r\n" + "the 8086 and 8088 microprocessors?", "400 ns", "400 ms", "2.5 MIPS", "1M byte"}, {"What was the first programming language constructed \r\n" + "of?", "Binary codes", "Mnemonic codes", "High-level codes", "Low-level codes"}, {"Who was the first modern person to develop a system \r\n" + "that accepted instructions and stored them in memory?", "John von Neumann", "Charles Babbage", "Grace Hopper", "Augusta Ada Byron"}, {"What was the first high-level programming language \r\n" + "called?", "FLOWMATIC", "COBOL", "FORTRAN", "ALGOL"}, };

```
public ch1_MICRO() {
    // Set the icon image of the frame
```

## CHAPTER 1 : Introduction

**Quiz: 15 items**

**What was the first mechanical calculator?**

**Slide rule**

**Analytical engine**

**Odometer**

**Abacus**

```
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new ImageIcon(Home.class.getResource("/images/OSch1.png")));
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
15 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
            buttonPanel.add(answerButtons[i]);
```

```

        answerButtons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkAnswer(( JButton ) e.getSource());
            }
        });
    }

    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(MICROquiz.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //assign random
    random_ans = new
Random();
    //show questions
    showQuestion();
}
//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch1questions[currentQuestio
n][0]);
    String[] answers =
    Arrays.copyOfRange(ch1questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }
}

//method to randomize choices
void shuffleAnswers(String[] answers) {
    for (int i = 0; i < answers.length; i++) {
        int j =
random_ans.nextInt(answers.length);
        String temp = answers[i];
        answers[i] = answers[j];
        answers[j] = temp;
    }
}

```

```

        answers[i] = answers[j];
        answers[j] = temp;
    }
}

// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch1questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch1questions.length) {

            Result_MICRO.Result_ch1MICRO fCh1MICRO = new
Result_MICRO().new Result_ch1MICRO();
                setVisible(false);
                fCh1MICRO.setVisible(true);
} else {
            //continue to show question
            showQuestion();
}
} else {
        currentQuestion++;
        if(currentQuestion > 14) {

            Result_MICRO.Result_ch1MICRO fCh1MICRO = new
Result_MICRO().new Result_ch1MICRO();
                setVisible(false);
                fCh1MICRO.setVisible(true);
                MICROquiz.clip1.stop();

playSoundNotLoop("sounds/resultbgmusic.wav");
}
else {
            showQuestion();
}
}
}
}

```

```

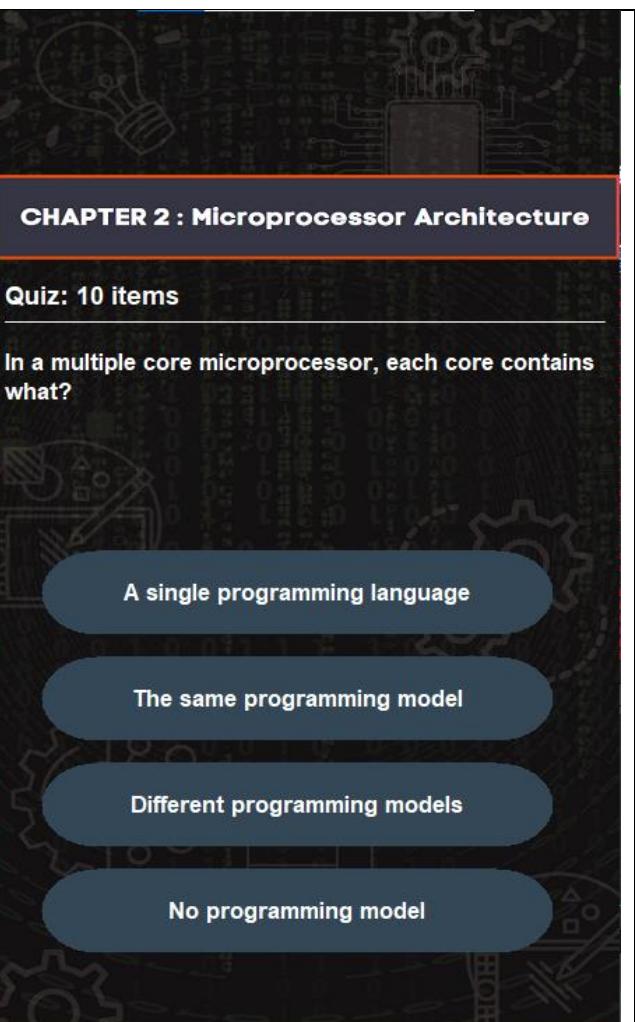
class ch2_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch2questions = {
        {"In a multiple core
microprocessor, each core "

```

Chapter 2 Microprocessor quiz

"contains\nwhat?",  
 +  
 "The same programming model",  
 "No programming model", "A single programming language",  
 "Different programming models",  
 {"Which microprocessors contain  
 full 32-bit internal \r\n"  
 +  
 "architectures?", "80386 through Core2", "8086 through the  
 80286",  
 "6502 and 6800", "8080 and  
 8085"},  
 {"What is the maximum  
 addressable memory space with a \r\n"  
 + "36-bit  
 address bus?", "64GBytes", "16GBytes",  
 "1MBytes", "4GBytes"},  
 {"Which register holds a part of  
 the result from a \r\n"  
 + "multiplication  
 or part of dividend before a division?", "RDX", "RDI", "RBP",  
 "RSI"},  
 {"Which flag indicates carry after  
 addition or borrow \r\n"  
 + "after  
 subtraction?", "C (carry)", "P (parity)",  
 "Overflow", "Zero flag"},  
 {"Which flag bit activates if a  
 word or doubleword is \r\n"  
 + "addressed on  
 a nonword or non-doubleword boundary?", "AC", "RF", "VM",  
 "ID"},  
 {"Which registers define the area  
 of memory used for \r\n"  
 + "the stack in  
 80386-Core2 microprocessors?", "SS and BP", "CS and SP", "ES  
 and BP", "FS and GS"},  
 {"What is the maximum memory  
 space that can be \r\n"  
 + "addressed in  
 real mode operation?", "the first 1M byte", "the first 10M byte",  
 "the first 5M byte", "the first 100M byte"},  
 {"Which register defines the start  
 of the code segment \r\n"  
 + "in a  
 system?", "Code segment register", "Stack pointer  
 register", "Pointer register", "Instruction pointer register"},  
 {"In protected mode, what does  
 the segment register \r\n"  
 + "contain  
 instead of a segment address?", "Selector", "Memory  
 location", "Access rights", "Descriptor"},  
 { };  
  
 public ch2\_MICRO() {  
 // Set the icon image of the frame



## CHAPTER 2 : Microprocessor Architecture

**Quiz: 10 items**

In a multiple core microprocessor, each core contains what?

A single programming language

The same programming model

Different programming models

No programming model

```
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel ch1label = new JLabel("");
        ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/MICROch2.png")));
;
        ch1label.setBounds(0, 116, 450, 64);
        contentPane.add(ch1label);

        JLabel quiznolabel = new JLabel("Quiz:
10 items");

        quiznolabel.setForeground(Color.WHITE);
        quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
        quiznolabel.setBounds(10, 191, 132, 29);
        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
;

        lblNewLabel.setForeground(Color.WHITE);
        lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
        lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setOpaque(false);
        buttonPanel.setBackground(new
Color(0, 0, 0, 0));
        buttonPanel.setBounds(36, 387, 367,
290);
        buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
        //add buttons
        answerButtons = new RoundButton[4];
        for (int i = 0; i < 4; i++) {
            answerButtons[i] = new RoundButton();
            answerButtons[i].setBackground(new
Color(51,71,86));
            answerButtons[i].setBorder(null);
            answerButtons[i].setForeground(Color.white);
```

```

        buttonPanel.add(answerButtons[i]);
        answerButtons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkAnswer((JButton) e.getSource());
            }
        });
    }
    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //assign random
    random_ans = new
Random();
    //show questions
    showQuestion();
}
//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch2questions[currentQuestio
n][0]);
    String[] answers =
Arrays.copyOfRange(ch2questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }
}
//method to randomize choices
void shuffleAnswers(String[] answers) {
    for (int i = 0; i < answers.length; i++) {
        int j =
random_ans.nextInt(answers.length);

```

```

        String temp = answers[i];
        answers[i] = answers[j];
        answers[j] = temp;
    }
}

// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch2questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch2questions.length) {

            Result_MICRO.Result_ch2MICRO fCh2MICRO = new
Result_MICRO().new Result_ch2MICRO();
            setVisible(false);
            fCh2MICRO.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

            Result_MICRO.Result_ch2MICRO fCh2MICRO = new
Result_MICRO().new Result_ch2MICRO();
            setVisible(false);
            fCh2MICRO.setVisible(true);
            MICROquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    }
}
}

```

```

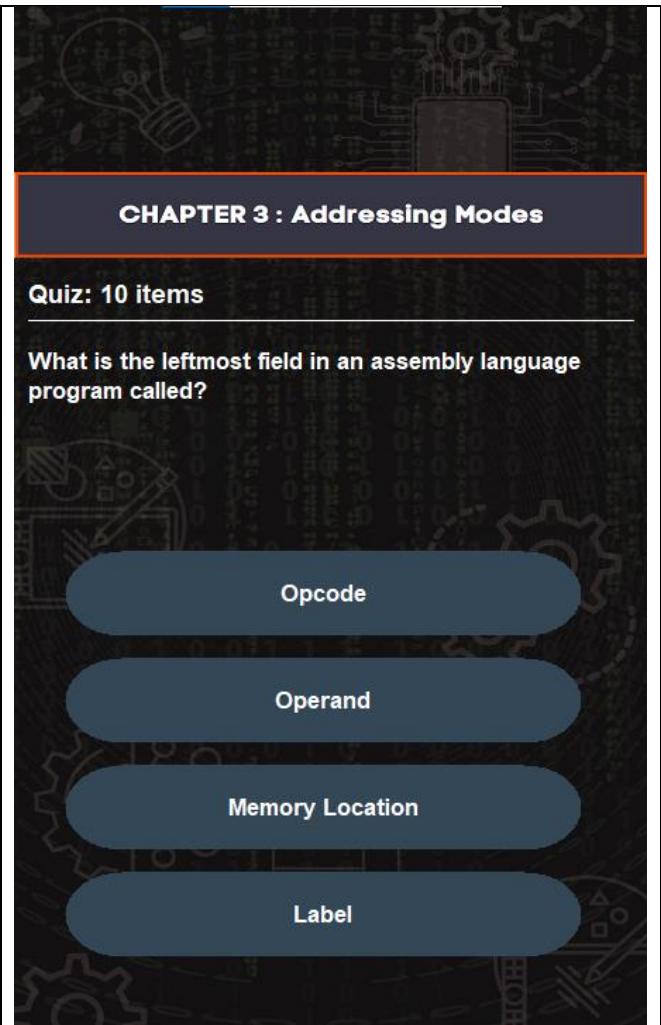
class ch3_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][][] ch3questions = {

```

Chapter 3 Microprocessor quiz

<p>assembly language \r\n" called?", {"Memory Location"}, {"What does MOV instruction do?", "Copies the source into the destination", "Moves the source to the destination", "source from the destination"}, {"Which type of addressing operates upon a byte or word \r\n"}, {"How is hexadecimal data represented in assembly \r\n"}, {"Which form of direct data addressing applies to a MOV \r\n"}, {"Which register holds a selector that accesses a \r\n"}, {"base address of the stack segment \r\n"}, {"in protected mode operation?", "SS register", "ES register", "SP register", "BP register"}, {"Which register(s) address memory in the data segment \r\n"}, {"by default on 80386 and above?", "EAX, EBX, ECX, EDX, EDI, and ESI", "BP", "BX, DI, and SI", "AL and CL"}, {"In the real mode, what is the maximum contents of a \r\n"}, {"32-bit register used for addressing memory?", "0000FFFFH", "11111111H", "00000000H", "FFFFFFFH"}, {"Which register holds the relative position of an \r\n"}, {"element in a memory array?", "Index register", "Stack segment register", "Base register", "BP register"}, }; </p>
---



## CHAPTER 3 : Addressing Modes

---

**Quiz: 10 items**

What is the leftmost field in an assembly language program called?

Opcode

Operand

Memory Location

Label

```

public ch3_MICRO() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/MICROch3.png")));
;

    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("_____");
;

    lblNewLabel.setForeground(Color.WHITE);
    lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
    lblNewLabel.setBounds(10, 202, 440,
29);
    contentPane.add(lblNewLabel);

    JPanel buttonPanel = new JPanel();
    buttonPanel.setOpaque(false);
    buttonPanel.setBackground(new
Color(0, 0, 0));
    buttonPanel.setBounds(36, 387, 367,
290);
    buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
    //add buttons
    answerButtons = new RoundButton[4];
    for (int i = 0; i < 4; i++) {
        answerButtons[i] = new RoundButton();
}

```

```

        answerButtons[i].setBackground(new
Color(51,71,86));
        answerButtons[i].setBorder(null);
        answerButtons[i].setForeground(Color.white);
        buttonPanel.add(answerButtons[i]);
        answerButtons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkAnswer((JButton) e.getSource());
            }
        });
    }

    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //assign random
    random_ans = new
Random();
    //show questions
    showQuestion();
}

//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch3questions[currentQuestio
n][0]);
    String[] answers =
Arrays.copyOfRange(ch3questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }
}
//method to randomize choices

```

```

        void shuffleAnswers(String[] answers) {
            for (int i = 0; i < answers.length; i++) {
                int j =
random_ans.nextInt(answers.length);
                String temp = answers[i];
                answers[i] = answers[j];
                answers[j] = temp;
            }
        }
        // method to check answer
        private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch3questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch3questions.length) {

            Result_MICRO.Result_ch3MICRO fCh3MICRO = new
Result_MICRO().new Result_ch3MICRO();
            setVisible(false);
            fCh3MICRO.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
        if(currentQuestion > 9) {

            Result_MICRO.Result_ch3MICRO fCh3MICRO = new
Result_MICRO().new Result_ch3MICRO();
            setVisible(false);
            fCh3MICRO.setVisible(true);
            MICROquiz.clip1.stop();

            playSoundNotLoop("sounds/resultbgmusic.wav");
        }
    }
}
}

```

```

class ch4_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;

```

Chapter 4 Microprocessor quiz

```

public static int scores;
//question and choices in an array
String[][] ch4questions = {
    {"Which type of instruction does
NOT fall under data \r\n",
        + "movement
instructions?",,
        {"JMP", "LODS", "LEA", "MOV"},,
        {"What program can be used to
create machine language \r\n"
            + "patches in
DOS and Visual for Windows?",, "DEBUG",, "Python",,
            "HTML",, "Java"},,
        {"What does the first 6 bits of the
first opcode byte of\r\n"
            + "an instruction
indicate?",, "Binary Opcode",, "Data Flow Direction",
            "Addressing Mode",
        "Displacement Mode"},,
        {"In register addressing, what
does the R/M field \r\n"
            + "select when
the MOD field is 00, 01 or 10?",, "One of the data memory
addressing modes",, "The instruction type",, "A fixed memory
location",, "A register"},,
        {"What is the meaning of the
R/M field when the MOD \r\n"
            + "field contains
a 00, 01, or 10?",, "Memory Addressing",, "Register Immidiate
Data",
            "Hardware I/O
Operations",, "Logic and Bitwise Operation"},,
        {"What is the opcode for the
given instruction?",, "100010",, "101110",, "110001",, "011010"},,
        {"What is the MOD field in the
displacement mode of \r\n"
            + "addressing
for 16-bit instructions?",, "00",, "10",, "11",, "01"},,
        {"Do MOV instructions referring
to memory through a \r\n"
            + "pointer need
a PTR directive?",, "No",, "Only if it's a DWORD",, "Yes",, "Depends
on the type of memory"},,
        {"Which instructions are used to
store and retrieve \r\n"
            + "data from the
LIFO stack memory?",, "PUSH/POP",
        "INC/DEC",, "ADD/SUB",, "JMP/JC"},,
        {"Which addressing mode allows
immediate data to be \r\n"
            + "pushed onto
the stack but not popped off the stack?",, "Immediate
addressing",, "Segment register addressing",, "Memory-addressing
PUSH",, "PUSH instruction with flags"},,
    };
}

```

## CHAPTER 4 : Data Movement

**Quiz: 10 items**

**Which type of instruction does NOT fall under data movement instructions?**

LEA

LODS

MOV

JMP

```

public ch4_MICRO() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new ImageIcon(Home.class.getResource("/images/MICROch4.png")));
;
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
    contentPane.add(quiznolabel);

    JLabel lblNewLabel = new
JLabel("_____");
;

    lblNewLabel.setForeground(Color.WHITE);
    lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
    lblNewLabel.setBounds(10, 202, 440,
29);
    contentPane.add(lblNewLabel);

    JPanel buttonPanel = new JPanel();
    buttonPanel.setOpaque(false);
    buttonPanel.setBackground(new
Color(0, 0, 0));
    buttonPanel.setBounds(36, 387, 367,
290);
    buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
    //add buttons
    answerButtons = new RoundButton[4];
    for (int i = 0; i < 4; i++) {
        answerButtons[i] = new RoundButton();
        answerButtons[i].setBackground(new
Color(51,71,86));
}
}

```

```

        answerButtons[i].setBorder(null);
        answerButtons[i].setForeground(Color.white);
        buttonPanel.add(answerButtons[i]);
        answerButtons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                checkAnswer((JButton) e.getSource());
            }
        });
    }
    contentPane.add(buttonPanel);
    questionsTextArea = new JTextArea();
    questionsTextArea.setEditable(false);

    questionsTextArea.setForeground(Color.WHITE);
    questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
    questionsTextArea.setOpaque(false);
    questionsTextArea.setBounds(10, 242,
430, 143);
    contentPane.add(questionsTextArea);

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);

    //initial values
    currentQuestion = 0;
    scores = 0;
    //assign random
    random_ans = new
Random();
    //show questions
    showQuestion();
}
//method to show questions
void showQuestion() {

    questionsTextArea.setText(ch4questions[currentQuestio
n][0]);
    String[] answers =
Arrays.copyOfRange(ch4questions[currentQuestion], 1, 5);
    shuffleAnswers(answers);
    for (int i = 0; i < 4; i++) {
        answerButtons[i].setText(answers[i]);
        answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
    }

}
//method to randomize choices
void shuffleAnswers(String[] answers) {
for (int i = 0; i < answers.length; i++) {

```

```

        int j =
random_ans.nextInt(answers.length);
        String temp = answers[i];
        answers[i] = answers[j];
        answers[j] = temp;
    }
}

// method to check answer
private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch4questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch4questions.length) {

            Result_MICRO.Result_ch4MICRO fCh4MICRO = new
Result_MICRO().new Result_ch4MICRO();
                setVisible(false);
                fCh4MICRO.setVisible(true);
            } else {
                //continue to show question
                showQuestion();
            }
        } else {
            currentQuestion++;
            if(currentQuestion > 9) {

                Result_MICRO.Result_ch4MICRO fCh4MICRO = new
Result_MICRO().new Result_ch4MICRO();
                    setVisible(false);
                    fCh4MICRO.setVisible(true);
                    MICROquiz.clip1.stop();

                    playSoundNotLoop("sounds/resultbgmusic.wav");
                }
            } else {
                showQuestion();
            }
        }
    }
}

```

```

class ch5_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array

```

Chapter 5 Microprocessor quiz

```

String[][] ch5questions = {
    {"Which of the following is not
allowed in the arithmetic?\r\n",
     + "metic
instructions of a microprocessor?",

    {"Memory-to-memory addition",
     "Subtraction", "Addition with carry", "Multiplication"},

    {"Which type of addition is used
when constant or \r\n",
     + "known data
are added?", "Immediate Addition", "Memory-to-Register
Addition", "Increment Instruction", "Array
Addition"},

    {"What instruction adds 1 to
any register or memory \r\n",
     + "location
except a segment register in x86 assembly \r\n",
     + "language?", "INC", "MOV",
     "SUB", "ADD"},

    {"Which processor introduced
the XADD instruction?", "80486", "80386", "8086", "Core2"},

    {"Which instruction subtracts 1
from any register \r\n",
     + "or memory
location?", "Decrement (DEC)", "Subtract-with-Borrow (SBB)",
     "Subtraction (SUB)",

    "Addition (ADD)",

    {"Which instruction functions as a
regular subtract-\r\n",
     + "ion, except
that the carry flag also subtracts from\r\n",
     + "the
difference?", "Subtraction-with-Borrow (SBB)", "Decrement
Subtraction", "Increment Subtraction", "Comparison instruction
(CMP)"},

    {"Which instruction compares the
destination operand \r\n",
     + "with the
accumulator in the 80486–Core2 processors?", "CMPEXCHGH",
     "SUB", "MOV", "ADD"},

    {"What circuitry was introduced
in newer microprocessors?\r\n",
     + "cessors for
faster multiplication?", "Special circuitry in Pentium-Core2",
     "Multiplication through separate circuits", "Immediate
multiplication operations", "Increased clocking periods"},

    {"What is the result of multiplying
two 8-bit numbers?", "16-bit product", "64-bit product", "128-bit
product", "32-bit product"},

    {"What type of multiplication is
allowed in 80386 processor?\r\n",
     + "cessor?", "32-
bit product", "64-bit product", "8-bit product", "16-bit product"},}

```

**CHAPTER 5 : Arithmetic and Logic**

**Quiz: 15 items**

Which of the following is not allowed in the arithmetic instructions of a microprocessor?

**Multiplication**

**Subtraction**

**Memory-to-memory addition**

**Addition with carry**

```

        {"In which register pair does the
result of 64-bit \r\n"
+ "multiplication
appear in Pentium 4?", "RDX: RAX", "FS: GS", "EBX: ECX",
"EAX: ESP"},

        {"In 64-bit mode Pentium 4 &
Core2, what type of \r\n"
+ "errors can
result from dividing a 128-bit number by\r\n"
+ "a 64-bit
number?", "Attempt to divide by zero and divide overflow",
"Overflow error and syntax error", "Attempt to divide by one and
divide overflow", "Syntax error and divide overflow"},

        {"In 16-bit division, what is the
size of the divi-\r\n"
+ "dend?", "32-
bit", "16-bit", "8-bit", "64-bit"},

        {"Which instruction is used to
zero-extend a 16-bit \r\n"
+ "unsigned
number in 80386 and above?", "MOVZX", "PENTIUM
4", "AX", "DX"},

        {"Which registers are used to
hold the dividend in a \r\n"
+ "64-bit
division?", "RDX:RAX", "RCX:RAX", "RBX:RAX", "RDX:RBX"}
};

public ch5_MICRO() {
    // Set the icon image of the frame

    setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
    ImageIcon(Home.class.getResource("/images/MICROch5.png")))
    ;
    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
15 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
}

```

```
quiznolabel.setBounds(10, 191, 132, 29);
contentPane.add(quiznolabel);

JLabel lblNewLabel = new
JLabel("_____"
);

lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
contentPane.add(lblNewLabel);

JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer(( JButton ) e.getSource());
        }
    });
}
contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
contentPane.add(questionsTextArea);

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
```

```

//initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {
        questionsTextArea.setText(ch5questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch5questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
            for (int i = 0; i < answers.length; i++) {
                int j =
random_ans.nextInt(answers.length);
                String temp = answers[i];
                answers[i] = answers[j];
                answers[j] = temp;
            }
        }
        // method to check answer
        private void
checkAnswer(JButton button) {
            if
(button.getText().equals(ch5questions[currentQuestion][1])) {
                currentQuestion++;
                scores++;
                //show the result if the user is done
on answering
                if (currentQuestion >=
ch5questions.length) {
                    Result_MICRO.Result_ch5MICRO fCh5MICRO = new
Result_MICRO().new Result_ch5MICRO();
                    setVisible(false);
                    fCh5MICRO.setVisible(true);
                } else {
                    //continue to show question
                    showQuestion();
                }
            } else {

```

```

        currentQuestion++;
        if(currentQuestion > 14) {

    Result_MICRO.Result_ch5MICRO fCh5MICRO = new
Result_MICRO().new Result_ch5MICRO();
        setVisible(false);
        fCh5MICRO.setVisible(true);
        MICROquiz.clip1.stop();

    playSoundNotLoop("sounds/resultbgmusic.wav");
    }
    else {
        showQuestion();
    }

}
}
}

```

```

class ch6_MICRO extends JFrame {

    private JPanel contentPane;
    private JButton[] answerButtons = new
JButton[4];
    private JTextArea questionsTextArea;
    private int currentQuestion;
    private Random random_ans;
    public static int scores;
    //question and choices in an array
    String[][] ch6questions = {
        {"Which group of program
control instructions includes\r\n"
            + "jumps, calls,
returns, interrupts, and machine \r\n"
            + "control
instructions?", "The Jump Group", "The Call
Group", "The Return Group", "The Loop Group"},

        {"What is the maximum distance
a short jump instruct-\r\n"
            + "ion can allow
a jump to in bytes?", "127", "256",
            "512", "1024"},

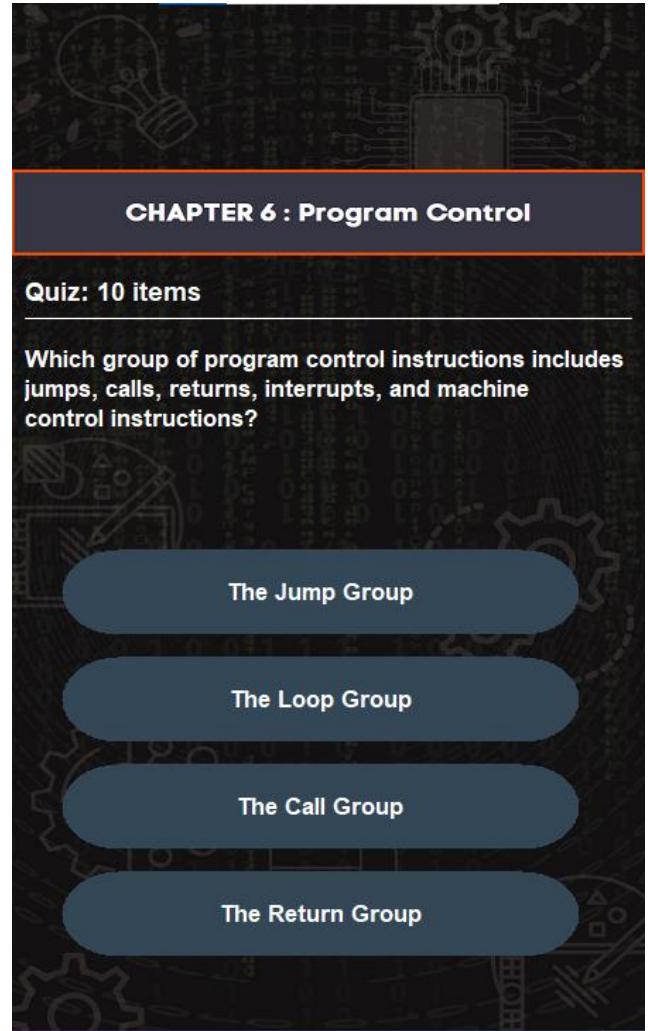
        {"What follows the opcode in a
short jump instruction?", "1-byte signed number representing
distance", "instruction code", "label identifying the
address", "hexadecimal address"},

        {"Which type of jump is
relocatable because it is a \r\n"
            + "relative jump
in Intel microprocessors?", "Near Jump", "Conditional Jump",
"Call Jump", "Far Jump"},

        {"What do bytes 2 and 3 of the 5-
byte far jump \r\n"
            + "instruction
contain?", "The new offset address", "The base address", }
    }
}

```

## Chapter 6 Microprocessor quiz



**CHAPTER 6 : Program Control**

**Quiz: 10 items**

Which group of program control instructions includes jumps, calls, returns, interrupts, and machine control instructions?

The Jump Group

The Loop Group

The Call Group

The Return Group

```

        "The new segment
address", "The current offset address",
        {"When accessing a descriptor in
protected mode, what \r\n"
            + "information is
obtained from the segment address of\r\n"
            + "a far jump
instruction?", "base address of the far jump segment", "current
offset address", "new segment base address", "new offset
address"},

        {"What is the jump table used for
in assembly language?", "Indirect jumps", "Conditional jumps",
"Direct jumps", "Function calls"},

        {"What is the range of short
jumps in 8086-80286?", "+127 and -128 bytes", "+1M", "+2G",
"+32K"},

        {"When comparing signed
numbers, which are the \r\n"
            + "instructions
used for terms greater than and less \r\n"
            + "than?", "JG,
JL, JGE, JLE", "JA, JB, JAB, JBE", "Conditional set
instructions", "JE, JNE"},

        {"What is the alternate name for
LOOPNE instruction?", "ENDIF", "MOV", "ADD", "LOOPNZ"},

};

public ch6_MICRO() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel ch1label = new JLabel("");
    ch1label.setIcon(new
ImageIcon(Home.class.getResource("/images/MICROch6.png")));
;

    ch1label.setBounds(0, 116, 450, 64);
    contentPane.add(ch1label);

    JLabel quiznolabel = new JLabel("Quiz:
10 items");

    quiznolabel.setForeground(Color.WHITE);
    quiznolabel.setFont(new Font("Arial",
Font.BOLD, 18));
    quiznolabel.setBounds(10, 191, 132, 29);
}

```

```

        contentPane.add(quiznolabel);

        JLabel lblNewLabel = new
JLabel("_____");
        "


lblNewLabel.setForeground(Color.WHITE);
lblNewLabel.setFont(new
Font("Copperplate Gothic Bold", Font.BOLD, 17));
lblNewLabel.setBounds(10, 202, 440,
29);
        contentPane.add(lblNewLabel);

        JPanel buttonPanel = new JPanel();
buttonPanel.setOpaque(false);
buttonPanel.setBackground(new
Color(0, 0, 0, 0));
buttonPanel.setBounds(36, 387, 367,
290);
buttonPanel.setLayout(new
GridLayout(4, 1, 6, 15));
//add buttons
answerButtons = new RoundButton[4];
for (int i = 0; i < 4; i++) {
    answerButtons[i] = new RoundButton();
    answerButtons[i].setBackground(new
Color(51,71,86));
    answerButtons[i].setBorder(null);
    answerButtons[i].setForeground(Color.white);
    buttonPanel.add(answerButtons[i]);
    answerButtons[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent e) {
            checkAnswer(( JButton ) e.getSource());
        }
    });
}
        contentPane.add(buttonPanel);
questionsTextArea = new JTextArea();
questionsTextArea.setEditable(false);

questionsTextArea.setForeground(Color.WHITE);
questionsTextArea.setFont(new
Font("Arial", Font.BOLD, 17));
questionsTextArea.setOpaque(false);
questionsTextArea.setBounds(10, 242,
430, 143);
        contentPane.add(questionsTextArea);

        JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);

```

```

//initial values
        currentQuestion = 0;
        scores = 0;
        //assign random
        random_ans = new
Random();
        //show questions
        showQuestion();
    }
    //method to show questions
    void showQuestion() {

        questionsTextArea.setText(ch6questions[currentQuestio
n][0]);
        String[] answers =
        Arrays.copyOfRange(ch6questions[currentQuestion], 1, 5);
        shuffleAnswers(answers);
        for (int i = 0; i < 4; i++) {
            answerButtons[i].setText(answers[i]);
            answerButtons[i].setFont(new
Font("Arial", Font.BOLD, 17));
        }

        }
        //method to randomize choices
        void shuffleAnswers(String[] answers) {
        for (int i = 0; i < answers.length; i++) {
            int j =
random_ans.nextInt(answers.length);
            String temp = answers[i];
            answers[i] = answers[j];
            answers[j] = temp;
        }
    }
    // method to check answer
    private void
checkAnswer(JButton button) {
    if
(button.getText().equals(ch6questions[currentQuestion][1])) {
        currentQuestion++;
        scores++;
        //show the result if the user is done
on answering
        if (currentQuestion >=
ch6questions.length) {

            Result_MICRO.Result_ch6MICRO fCh6MICRO = new
Result_MICRO().new Result_ch6MICRO();
            setVisible(false);
            fCh6MICRO.setVisible(true);
        } else {
            //continue to show question
            showQuestion();
        }
    } else {
        currentQuestion++;
    }
}

```

<pre>         if(currentQuestion &gt; 9) {      Result_MICRO.Result_ch6MICRO fCh6MICRO = new Result_MICRO().new Result_ch6MICRO();         setVisible(false);     fCh6MICRO.setVisible(true);     MICROquiz.clip1.stop();      playSoundNotLoop("sounds/resultbgmusic.wav");     }     else {         showQuestion();     }  } } } </pre>	
<pre> import java.awt.*; import java.awt.event.*; import javax.swing.*;  public class study extends JFrame {      private JPanel contentPane;      /**      * Launch the application.      */     public static void main(String[] args) {         study frame = new study();         frame.setVisible(true);     }      /**      * Create the frame.      */     public study() {         // Set the icon image of the frame  setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResou     setResource("/images/logo.png")));         setUndecorated(true);          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);         setBounds(100, 100, 450, 730);         setLocationRelativeTo(null);         setResizable(false);         contentPane = new JPanel();         setContentPane(contentPane);         contentPane.setLayout(null);          JButton backbtn = new JButton("");         backbtn.setOpaque(false);         backbtn.addMouseListener(new MouseAdapter() {     @Override </pre>	<p>Study Lesson button</p> 

```

        public void mouseEntered(MouseEvent
e) {
            backbtn.setIcon(new
ImageIcon(study.class.getResource("/images/back
buttonhover.png")));
        }
        @Override
        public void mouseExited(MouseEvent e)
{
            backbtn.setIcon(new
ImageIcon(study.class.getResource("/images/back
button.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            Home fhome = new Home();
            setVisible(false);
            fhome.setVisible(true);
        }
    );
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(study.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

    JLabel subjectlabel = new JLabel("");
    subjectlabel.setIcon(new
ImageIcon(study.class.getResource("/images/subject.png")));
    subjectlabel.setBounds(63, 144, 318, 80);
    contentPane.add(subjectlabel);

    JButton SDbtn = new JButton("");
    SDbtn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent
e) {
            SDbtn.setIcon(new
ImageIcon(study.class.getResource('/images/SDhover.png')));
        }
        @Override
        public void mouseExited(MouseEvent e)
{
            SDbtn.setIcon(new
ImageIcon(study.class.getResource("/images/SD.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            studySD fstudySD = new
studySD();
            setVisible(false);
            fstudySD.setVisible(true);
        }
    });

```

```

        }
    });
    SDbtn.setBorder(null);
    SDbtn.setOpaque(false);
    SDbtn.setIIcon(new
ImageIcon(study.class.getResource("/images/SD.png")));
    SDbtn.setBounds(73, 333, 293, 62);
    contentPane.add(SDbtn);

    JButton microbtn = new JButton("");
    microbtn.addMouseListener(new
MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent
e) {
            microbtn.setIIcon(new
ImageIcon(study.class.getResource("/images/microhover.png")));
        }
        public void mouseExited(MouseEvent e)
{
            microbtn.setIIcon(new
ImageIcon(study.class.getResource("/images/micro.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            studyMICRO fMicro = new
studyMICRO();
            setVisible(false);
            fMicro.setVisible(true);
        }
    });
    microbtn.setBorder(null);
    microbtn.setIIcon(new
ImageIcon(study.class.getResource("/images/micro.png")));
    microbtn.setBounds(73, 543, 293, 62);
    contentPane.add(microbtn);

    JButton OSbtn = new JButton("");
    OSbtn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent
e) {
            OSbtn.setIIcon(new
ImageIcon(quizmenu.class.getResource("/images/OShover.png")
));
        }
        public void mouseExited(MouseEvent e)
{
            OSbtn.setIIcon(new
ImageIcon(study.class.getResource("/images/OS.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            studyOS fOs = new studyOS();

```

```

        setVisible(false);
        fOs.setVisible(true);
    }
});

OSbtn.setIcon(new
ImageIcon(study.class.getResource("/images/OS.png")));
OSbtn.setBorder(null);
OSbtn.setBounds(73, 439, 293, 62);
contentPane.add(OSbtn);

JLabel quizmenulabel = new JLabel("");
quizmenulabel.setIcon(new
ImageIcon(study.class.getResource("/images/bg.png")));
quizmenulabel.setBounds(0, 0, 450, 730);
contentPane.add(quizmenulabel);
}
}
}

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.*;

public class studySD extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        studySD frame = new studySD();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public studySD() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

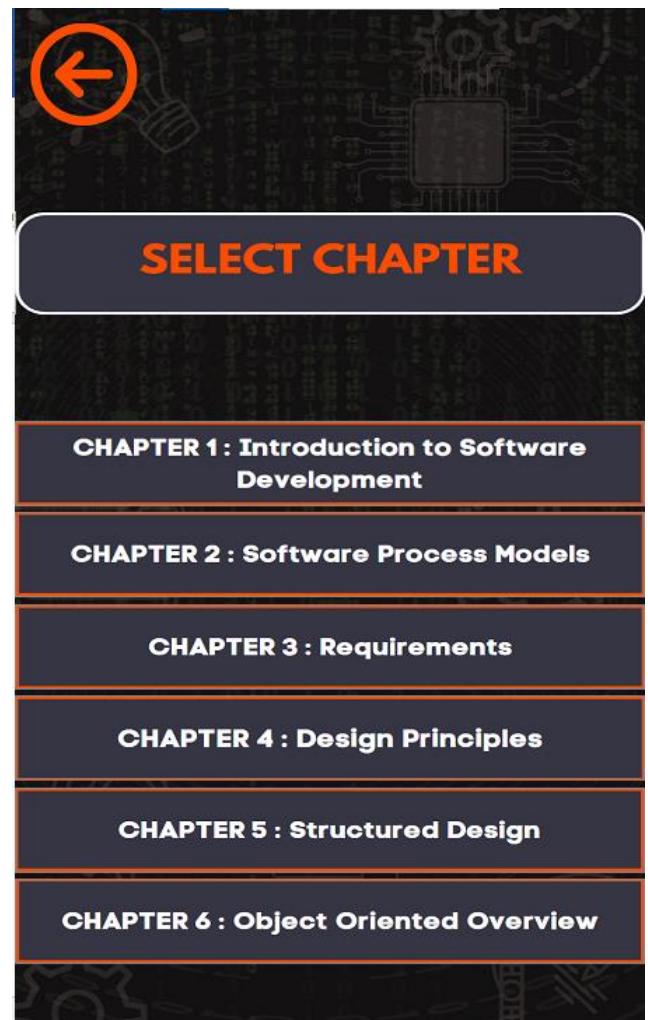
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton backbtn = new JButton("");

```

## Software Design Materials



```

backbtn.setOpaque(false);
backbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        study fStudy = new study();
        setVisible(false);
        fStudy.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

 JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        ch1SDstudy fCh1SDstudy =
new ch1SDstudy();
        setVisible(false);
        fCh1SDstudy.setVisible(true);
    }
});
ch1btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch1.png")));
ch1btn.setBounds(0, 295, 450, 61);
contentPane.add(ch1btn);

```

```

JButton ch2btn = new JButton("");
ch2btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch2SDstudy();
        ch2SDstudy fCh2SDstudy =
            setVisible(false);
            fCh2SDstudy.setVisible(true);
        }
    });
    ch2btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch2.png")));
    ch2btn.setBounds(0, 360, 450, 61);
    contentPane.add(ch2btn);

JButton ch3btn = new JButton("");
ch3btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch3SDstudy();
        ch3SDstudy fCh3SDstudy =
            setVisible(false);
            fCh3SDstudy.setVisible(true);
        }
    });
    ch3btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch3.png")));
    ch3btn.setBounds(0, 426, 450, 61);
    contentPane.add(ch3btn);

JButton ch4btn = new JButton("");
ch4btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch4SDstudy();
        ch4SDstudy fCh4SDstudy =
            setVisible(false);
            fCh4SDstudy.setVisible(true);
        }
    });
    ch4btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch4.png")));
    ch4btn.setBounds(0, 491, 450, 61);
    contentPane.add(ch4btn);

JButton ch5btn = new JButton("");
ch5btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch5SDstudy();
        ch5SDstudy fCh5SDstudy =
            setVisible(false);

```

```

        fCh5SDstudy.setVisible(true);
    }
});

ch5btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch5.png")));
ch5btn.setBounds(0, 557, 450, 61);
contentPane.add(ch5btn);

JButton ch6btn = new JButton("");
ch6btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch6SDstudy fCh6SDstudy =
new ch6SDstudy();
        fCh6SDstudy.setVisible(false);
        fCh6SDstudy.setVisible(true);
    }
});
ch6btn.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/ch6.png")));
ch6btn.setBounds(0, 622, 450, 61);
contentPane.add(ch6btn);

JLabel SDquizlabel = new JLabel("");
SDquizlabel.setIcon(new
ImageIcon(SDquiz.class.getResource("/images/bg.png")));
SDquizlabel.setBounds(0, 0, 450, 730);
contentPane.add(SDquizlabel);
}
}

```

```

public class ch1SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch1SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch1SDstudy.class.getResource("/images/ch1.png")));

```

Chapter 1 Software Design Material

```

chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new

MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studySD fstudySD =
new studySD();
        fstudySD.setVisible(false);

        fstudySD.setVisible(true);
    }
    });
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

    TextArea textArea = new TextArea();
    textArea.setForeground(new Color(0, 0,
0));
    textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
    textArea.setBackground(new Color(255,
255, 255));
    textArea.setBounds(10, 227, 430, 492);
    textArea.setEditable(false);

    contentPane.add(textArea);

    //to display the text from the text file to
the text area
    try {
        BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch1SD.txt"));

```



## CHAPTER 1: Introduction to Software Development

Software engineering also involves a process and includes software development, but it also includes the entire management side of creating a computer program that people will use, including project management, configuration management, scheduling and estimation, baseline building, scheduling, managing people, and several other things.

So, software development is a narrowing of the focus of software engineering to just that part concerned with the creation of the actual software. And it's a broadening of the focus of programming to include analysis, design, and issues.

**How to Develop Software?**

Despite the fact that software development is only part of software engineering, software development is the heart of every software project. After all, at the end of the day what you deliver to the user is working code. A team of developers working in concert usually creates that code.

In order to do software development, you need the following:

- A small, well integrated team: Small teams have fewer lines of communication than larger ones. It's easier to get to know your teammates on a small team and can get to know their strengths and weaknesses, who knows what, and who is the go-to person for particular problems or particular tools.
- Good communication among team members: Constant communication among team members is critical to day to day progress and successful project completion. Teams that are co-located are better at communicating and communicating than teams that are distributed geographically.

```

String line;
StringBuilder sb = new StringBuilder();

while ((line = reader.readLine()) != null) {
    String[] words = line.trim().split("\\s+");
    int lineLength = 0;

    for (String word : words) {
        if (lineLength + word.length() > 80) {
            sb.append("\n");
            lineLength = 0;
        }

        sb.append(word).append(" ");
        lineLength += word.length() + 1;
    }

    sb.append("\n");
}

reader.close();
textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
}

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglab.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}
}

```

```

public class ch2SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch2SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
}

```

Chapter 2 Software Design Material

```

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch2SDstudy.class.getResource("/images/ch2.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studySD fstudySD =
new studySD();
        setVisible(false);

        fstudySD.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);
//to display the text from the text file to
the text area

```

**CHAPTER 2 : Software Process Models**

Every program has a life cycle. It doesn't matter how large or small the project is, or how many people are working on it. All programs go through the same steps:

1. Conception
2. Requirements gathering/exploration/modeling
3. Design
4. Coding and debugging
5. Testing
6. Release
7. Maintenance software evolution
8. Retirement

Your program may compress some of these steps or combine two or more into a single piece of work, but all programs go through all steps of the life cycle. Although every program has a life cycle, many different process variations encompass these steps.

Every development process, however, is a variation on two fundamental types. In the first type, the project team will generally do a complete life cycle at least steps 2 through 7 before they go back and start on the next version of the product. In the second type, which is more prevalent now, the project team will generally do a partial life cycle usually steps 3 through 5 and iterate through those steps several times before proceeding to the release step.

These two general process types can be implemented using two classic management models. These are traditional plan driven models and the iterative development models. In plan driven models, the methodology tends to be stricter in terms of process steps and releases happen. Plan driven models have more clearly defined phases.

```

        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch2SD.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);
}
}

```

```

public class ch3SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch3SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
    }
}

```

### Chapter 3 Software Design Material

```

setContentPane(contentPane);
contentPane.setLayout(null);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch3SDstudy.class.getResource("/images/ch3.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

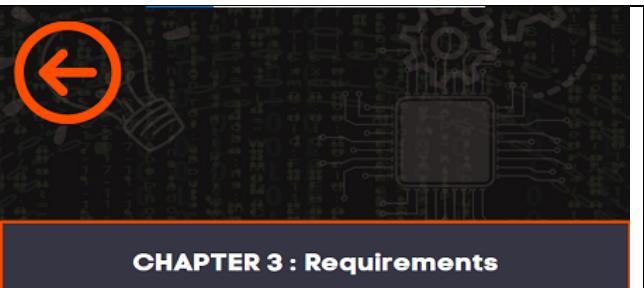
JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studySD fstudySD =
new studySD();
        fstudySD =
setVisible(false);

        fstudySD.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);

```



**CHAPTER 3 : Requirements**

**Types of Requirements**

We're really talking about functional requirements, the list of features the user will see and be able to use when they fire up your program. These "black box" requirements that show the external behavior of your program as the user is concerned these are the only requirements that matter. I plan-driven process the output of this activity of identifying requirements: functional specification of what the software system is supposed to do. agile process the output is a set OF user stories that define the product backlog.

**3.1 Type of Requirements**

We have four type of requirements: user requirements, domain requirements, non-functional requirements, and non-requirements. User requirements are always expressed in natural language. They are the details of what the user expects to see as they use the program. They also include descriptions of layouts, dialog boxes, and menus. Any interaction element in the program should be described in the user requirements. For example:

Logging into the system: When Gloria clicks the Login button on the main menu, a Login dialog box appears in the middle of the screen. The Login dialog box contains two text boxes, labeled "Username" and "Password." There must be two buttons in the dialog box, labeled "Submit" and "Cancel."

In normal usage, she will click in the Username text box and type in her username, and then click in (or tab to) the Password text box and type in her password. The text typed in the Password text box must be hidden. On

```

//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch3SD.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglab.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}
}

```

```

public class ch4SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch4SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
    }
}

```

Chapter 4 Software Design Material

```

        setContentPane(contentPane);
        contentPane.setLayout(null);

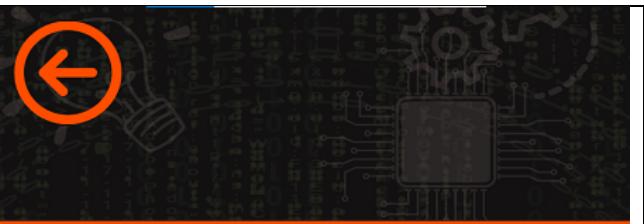
        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch4SDstudy.class.getResource("/images/ch4.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                new studySD();
                studySD fstudySD =
                setVisible(false);

                fstudySD.setVisible(true);
            }
        });
        backbtn.setBackground(Color.WHITE);
        backbtn.setBorder(null);
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
        contentPane.add(backbtn);

        TextArea textArea = new TextArea();
        textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
        textArea.setEditable(false);
    }
}

```



## CHAPTER 4 : Design Principles

One way to look at software problems is with a model that divides them into two different layers:

- "Wicked" problems fall in the upper layer: These are problems that typically come from domains outside of computer science (such as biology, business, meteorology, sociology, political science, and so on). These types of problems tend to be open ended, ill defined, and large in the sense that they require much work. For example, pretty much any kind of a web commerce application is a wicked problem. Horst W. J. Rittel and Melvin M. Webber, in a 1973 paper on social policy,<sup>1</sup> gave a definition for and a set of characteristics used to recognize a wicked problem that we'll look at later in this chapter.
- "Tame" problems fall in the lower layer: These problems tend to cut across other problem domains and tend to be better defined and small. Sorting and searching are great examples of tame problems. Small and well defined, "easy," however. Tame problems can be very complicated and difficult to solve. It's just that they're clearly defined and you know when you have a solution. These are the kinds of problems that provide computer scientists with foundations in terms of data structures and algorithms for the wicked problems we solve from other problem domains. According to Rittel and Webber, a wicked problem is one for which the requirements are completely known only at the end of the problem. A tame problem is one for which the requirements and solution evolve over time. It turns out this describes most of the "interesting" problems in software development. Recently, Jeff Conklin has revised Rittel and Webber's definition of a wicked problem and provided a more succinct list of the characteristics of wicked problems. To paraphrase:
  1. A wicked problem is not understood until after the creation of a solution. Another way of saying this is that the problem is defined and solved at the same time.
  2. Wicked problems have no stopping rule: You can create increments of solutions, but you never get to a final solution.

```

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch4SD.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);
}
}

```

```

public class ch5SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch5SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
    }
}

```

Chapter 5 Software Design Material

```

contentPane = new JPanel();
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch5SDstudy.class.getResource("/images/ch5.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studySD fstudySD =
new studySD();
        setVisible(false);

        fstudySD.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

```

**CHAPTER 5 : Structured Design**

Programming languages created from this point onward, while not eliminating the goto statement (except for Java, which has none), certainly downplayed it. Courses that taught programming encouraged students to avoid it. Problem solving was taught in a top-down structured manner, where one starts with the problem statement and attempts to break the problem down into solvable sub-problems. The process continues until each sub-problem is small enough to be either trivial or very easy to solve. This technique is called structured programming. Before the advent and acceptance of object-oriented programming in the mid 1980s, this was the standard approach to problem solving and programming. It's still one of the best ways to approach a large class of problems.

**Stepwise Refinement**

Niklaus Wirth formalized the structured design technique in his 1971 paper "Program Development by Stepwise Refinement".<sup>3</sup> Stepwise refinement is a process of designing programs that consists of a set of refinement steps. In each given task, the task is broken up into a number of subtasks. Each refinement step must be accompanied by a refinement of the data description and the interface. The degree of modularity obtained will determine the ease or difficulty with which a program can be adapted to changes in requirements or environment. Refinement, you use a notation that's natural to the problem space. As you refine a programming language for description as long as possible. Each refinement step implies a number of design decisions based on a set of design criteria. These criteria include efficiency of time and space, clarity, and regularity of structure (simplicity). Refinement can proceed in two ways: top-down and bottom-up. Top-down refinement is characterized by moving from a general description of the problem to detailed statements of what individual modules do. The guiding principle behind stepwise refinement is that humans can concentrate on only a few things at a time—Miller's famous 7 +/- 2 rule.

```

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch5SD.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

```

```

public class ch6SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch6SDstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
    }
}

```

Chapter 6 Software Design Material

```

        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

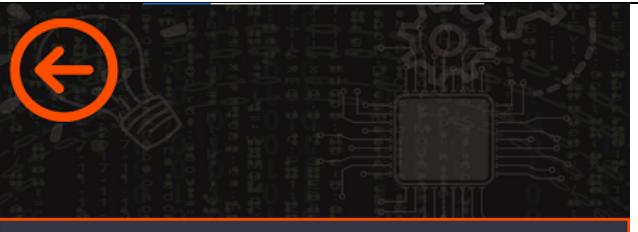
        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch6SDstudy.class.getResource("/images/ch6.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseListener() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                studySD fstudySD =
new studySD();
                setVisible(false);

                fstudySD.setVisible(true);
            }
        });
        backbtn.setBackground(Color.WHITE);
        backbtn.setBorder(null);
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
        contentPane.add(backbtn);

        TextArea textArea = new TextArea();
        textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
    }
}

```



## CHAPTER 6 : Object Oriented Overview

Well, yes, we've all learned about the object-oriented programming (OOP) paradigm before. But it never hurts to go over some basic definitions so all on the same page for our discussion about object oriented analysis design.

First of all, objects are things. They have an identity (a name), a state (of attributes that describes the current data stored inside the object), a defined set of behaviors that operate on that state. A stack is an object (an automobile, a bank account, a window, or a button in a graphical user interface). In an object-oriented program, a set of cooperating objects pass messages among themselves. The messages make requests of the objects to invoke methods that either perform operations on their data (thus changing the state of the object), or to report on the current state of the object. Eventually work gets done. Objects use encapsulation and information hiding (remember, they're different) to isolate data and operations from other objects in the program. Shared data areas are (usually) eliminated. Objects are instances of classes that define attribute types and operations.

Classes are templates for objects. Classes can also be thought of as factories that generate objects. So an Automobile class will generate instances of Automobile objects, a Stack class will create a new stack object, and a Queue class will create a new queue. Classes may inherit attributes and behaviors from other classes. Classes may be arranged in a class hierarchy where one class (a super-class) is a generalization of one or more other classes (sub-classes). A sub-class inherits the attributes and operations from its super class and adds new methods or attributes of its own. In this sense a sub-class is more specific and detailed than its super class; hence, we say that a sub-class extends a super class. For example, a BankAccount object may include a customer's name, address, balance, and a unique BankAccount ID number.

```

        textArea.setEditable(false);

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch6SD.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class studyOS extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        studyOS frame = new studyOS();
    }
}

```

Operating System button

```

        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public studyOS() {
        // Set the icon image of the frame

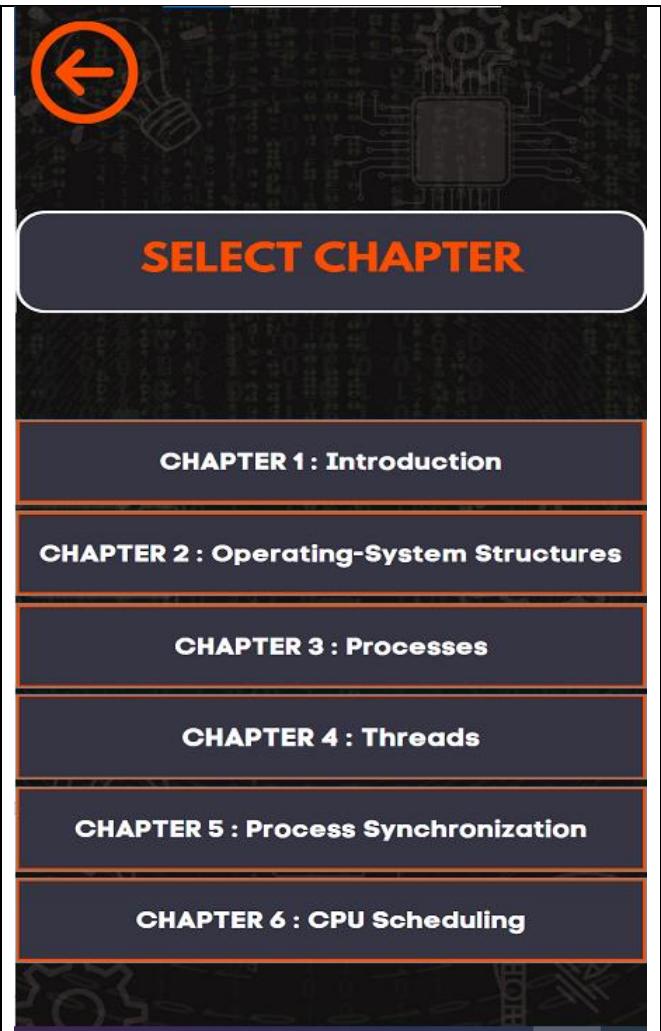
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        study fStudy = new study();
        setVisible(false);
        fStudy.setVisible(true);
    }
});
        backbtn.setBackground(Color.WHITE);
        backbtn.setBorder(null);
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
        contentPane.add(backbtn);
    }
}

```



```

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(studyOS.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        new ch1OSstudy();
        ch1OSstudy fCh1SDstudy =
setVisible(false);
fCh1SDstudy.setVisible(true);

    }
});
ch1btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch1.png")));
ch1btn.setBounds(0, 295, 450, 61);
contentPane.add(ch1btn);

JButton ch2btn = new JButton("");
ch2btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        new ch2OSstudy();
        ch2OSstudy fCh2SDstudy =
setVisible(false);
fCh2SDstudy.setVisible(true);

    }
});
ch2btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch2.png")));
ch2btn.setBounds(0, 360, 450, 61);
contentPane.add(ch2btn);

JButton ch3btn = new JButton("");
ch3btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        new ch3OSstudy();
        ch3OSstudy fCh3SDstudy =
setVisible(false);
fCh3SDstudy.setVisible(true);

    }
});
ch3btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch3.png")));
ch3btn.setBounds(0, 426, 450, 61);
contentPane.add(ch3btn);

JButton ch4btn = new JButton("");

```

```

ch4btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch4OSstudy();
        ch4OSstudy fCh4SDstudy =
            setVisible(false);
            fCh4SDstudy.setVisible(true);
        }
    );
    ch4btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch4.png")));
    ch4btn.setBounds(0, 491, 450, 61);
    contentPane.add(ch4btn);

    JButton ch5btn = new JButton("");
    ch5btn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent
e) {
new ch5OSstudy();
        ch5OSstudy fCh5SDstudy =
            setVisible(false);
            fCh5SDstudy.setVisible(true);
        }
    );
    ch5btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch5.png")));
    ch5btn.setBounds(0, 557, 450, 61);
    contentPane.add(ch5btn);

    JButton ch6btn = new JButton("");
    ch6btn.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent
e) {
new ch6OSstudy();
        ch6OSstudy fCh6SDstudy =
            setVisible(false);
            fCh6SDstudy.setVisible(true);
        }
    );
    ch6btn.setIcon(new
ImageIcon(studyOS.class.getResource("/images/OSch6.png")));
    ch6btn.setBounds(0, 622, 450, 61);
    contentPane.add(ch6btn);

    JLabel studyOSlabel = new JLabel("");
    studyOSlabel.setIcon(new
ImageIcon(studyOS.class.getResource("/images/bg.png")));
    studyOSlabel.setBounds(0, 0, 450, 730);
    contentPane.add(studyOSlabel);
}
public class ch1OSstudy extends JFrame {
    private JPanel contentPane;

```

Chapter 1 Operating System Material

```

    /**
     * Create the frame.
     */
    public ch1OSstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new ImageIcon(ch1OSstudy.class.getResource("/images/OSch1.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                studyOS fstudyOS =
new studyOS();
                setVisible(false);

                fstudyOS.setVisible(true);
            }
        });
        backbtn.setBackground(Color.WHITE);
    }
}

```

**CHAPTER 1: Introduction**

What is an Operating System?  
A program that acts as an intermediary between a user of a computer and computer hardware. Operating system goals:  

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Operating System Definition  

- OS is a resource allocator
- Manages all resources
- Decides between conflicting requests for efficient and fair resource use
- OS is a control program
- Controls execution of programs to prevent errors and improper use of computer
- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is an approximation
- But varies wildly
- "The one program running at all times on the computer" is the kernel.
- Everything else is either
  - a system program (ships with the operating system), or
  - an application program.

Computer Startup  

- bootstrap program is loaded at power-up or reboot
- Typically stored in ROM or EPROM, generally known as firmware
- Initializes all aspects of system
- Loads operating system kernel and starts execution

```

        backbtn.setBorder(null);
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
        contentPane.add(backbtn);

        TextArea textArea = new TextArea();
        textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
        textArea.setEditable(false);

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch1OS.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

public class ch2OSstudy extends JFrame {

```

Chapter 2 Operating System Material

```

private JPanel contentPane;

/**
 * Create the frame.
 */
public ch2OSstudy() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel chapterlabel = new JLabel("");
    chapterlabel.setIcon(new ImageIcon(ch2OSstudy.class.getResource("/images/OSch2.png")));
    chapterlabel.setBounds(0, 145, 450, 74);
    contentPane.add(chapterlabel);

    JButton backbtn = new JButton("");
    backbtn.setOpaque(false);
    backbtn.addMouseListener(new
MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
        }
        @Override
        public void
mouseExited(MouseEvent e) {
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        }
        @Override
        public void
mouseClicked(MouseEvent e) {
            studyOS fstudyOS =
new studyOS();
            setVisible(false);

            fstudyOS.setVisible(true);
        }
    });
}

```

## CHAPTER 2 : Operating-System Structures

**Operating System Services**

- Operating systems provide an environment for execution of programs services to programs and users
- One set of operating-system services provides functions that are help the user:
  - User interface - Almost all operating systems have a user interface (UI)
  - Varies between Command-Line Interpreter(CLI), Graphics User Interface(Batch)
  - Program execution - The system must be able to load a program into memory to run that program, end execution, either normally or abnormally (indicating error)
  - I/O operations - A running program may require I/O, which may involve reading from or writing to a file or an I/O device
- One set of operating-system services provides functions that are help the user (Cont.):
  - File-system manipulation - The file system is of particular interest. Programs need to read and write files and directories, create and delete them, search them, list file information, permission management.
  - Communications – Processes may exchange information, on the same computer or between computers over a network
  - Communications may be via shared memory or through message passing moved by the OS
  - Error detection – OS needs to be constantly aware of possible errors
  - May occur in the CPU and memory hardware, in I/O devices, in user programs
  - For each type of error, OS should take the appropriate action to ensure correct and consistent computing
  - Debugging facilities can greatly enhance the user's and programmer's productivity

```

    });
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

    TextArea textArea = new TextArea();
    textArea.setForeground(new Color(0, 0,
0));
    textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
    textArea.setBackground(new Color(255,
255, 255));
    textArea.setBounds(10, 227, 430, 492);
    textArea.setEditable(false);

    contentPane.add(textArea);
    //to display the text from the text file to
the text area
    try {
        BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch2OS.txt"));
        String line;
        StringBuilder sb = new StringBuilder();

        while ((line = reader.readLine()) != null) {
            String[] words = line.trim().split("\\s+");
            int lineLength = 0;

            for (String word : words) {
                if (lineLength + word.length() > 80) {
                    sb.append("\n");
                    lineLength = 0;
                }

                sb.append(word).append(" ");
                lineLength += word.length() + 1;
            }

            sb.append("\n");
        }

        reader.close();
        textArea.setText(sb.toString());
    } catch (IOException e) {
        e.printStackTrace();
    }
    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);
}

```

```

        }

public class ch3OSstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch3OSstudy() {
        // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new ImageIcon(ch3OSstudy.class.getResource("/images/OSch3.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
Icon(new ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png"))));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
Icon(new ImageIcon(quizmenu.class.getResource("/images/back
button.png"))));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                studyOS fstudyOS =
new studyOS();
                setVisible(false);
            }
        });
    }
}

```

## Chapter 3 Operating System Material

**CHAPTER 3 : Processes**

**Process**  
Fundamental to the structure of operating systems  
A process can be defined as:

- A program in execution
- An instance of a running program
- The entity that can be assigned to, and executed on, a processor
- A unit of activity characterized by a single sequential thread of execution, current state, and an associated set of system resources

**Terminology**  
Application = service = program

- Script
- Process
- Daemon
- Threads
- Job

**Process Concept**  
An operating system executes a variety of programs that run as a process  
Process – a program in execution; process execution must progress sequentially in a sequential fashion. No parallel execution of instructions of a single process  
Multiple parts

- The program code, also called text section
- Current activity including program counter, processor registers
- Stack containing temporary data
- § Function parameters, return addresses, local variables

Data section containing global variables  
Heap containing memory dynamically allocated during run time

```

        fstudyOS.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);
//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch3OS.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));

```

```

        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }

}

public class ch4OSstudy extends JFrame {
    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch4OSstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new ImageIcon(ch4OSstudy.class.getResource("/images/OSch4.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {

```

## Chapter 4 Operating System Material

**CHAPTER 4 : Threads**

**Terminology**

- Multiprogramming
- A computer running more than one program at a time (like running Firefox simultaneously)
- Context switching
- Multiprocessing
- A computer using more than one CPU (processor) or core at a time
- Multitasking
- Multitasking is a logical extension of multi programming (time sharing)
- Tasks sharing a common resource (like 1 CPU)
- Multithreading
- Thread (a code segments)
- is an extension of multitasking

**Processes and Threads**

- These two process characteristics are treated independently by the operating system
- The unit of execution (CPU utilization) is referred to as a thread or lightweight process.
- The unit of resource ownership is referred to as a process or task.
- The unit of dispatching is referred to as a thread or lightweight process.
- The unit of resource ownership is referred to as a process or task.
- Multithreading - The ability of an OS to support multiple, concurrent processes execution within a single process

**Motivation**

- Most modern applications are multithreaded
- Example:
- Web browser: one thread displays images or text while another thread

```

new studyOS();
studyOS fstudyOS =
    setVisible(false);

    fstudyOS.setVisible(true);
}
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);
//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch4OS.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
}

```

```

        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

public class ch5OSstudy extends JFrame {
    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch5OSstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch5OSstudy.class.getResource("/images/OSch5.png"
)));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override

```

## Chapter 5 Operating System Material

The screenshot shows a presentation slide with a dark background featuring a circuit board pattern. At the top left is a large orange circular arrow icon with a white arrow pointing left. The title 'CHAPTER 5 : Process Synchronization' is centered in a white box with a black border. Below the title, there is a section titled 'Background' followed by a bulleted list of points. Further down, there are two code snippets: 'Producer' and 'Consumer', both enclosed in white boxes with black borders.

**CHAPTER 5 : Process Synchronization**

**Background**

- Concurrent access to shared data may result in data inconsistency
- Maintaining data consistency requires mechanisms to ensure the ordered execution of cooperating processes
- Suppose that we wanted to provide a solution to the consumer-producer problem that fills all the buffers. We can do so by having an integer count that keeps track of the number of full buffers. Initially, count is set to 0. It is incremented by the producer after it produces a new buffer and is decremented by the consumer after it consumes a buffer.

**Producer**

```

while (true) {
    /* produce an item and put in nextProduced */
    while (counter == BUFFER_SIZE)
        ; // do nothing
    buffer [in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    counter++;
}

```

**Consumer**

```

while (true) {
    while (counter == 0)
        ; // do nothing
    nextConsumed = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    counter--;
    /* consume the item in nextConsumed */
}

```

```

        public void
mouseClicked(MouseEvent e) {
    studyOS fstudyOS =
new studyOS();
    setVisible(false);

    fstudyOS.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);
//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch5OS.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
}

```

```

        e.printStackTrace();
    }

    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);
}

}

public class ch6OSstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch6OSstudy() {
        // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch6OSstudy.class.getResource("/images/OSch6.png"
)));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
        });
    }
}

```

## Chapter 6 Operating System Material



**CHAPTER 6 : CPU Scheduling**

**Basic Concepts**

- Maximum CPU utilization obtained with multiprogramming
- CPU-I/O Burst Cycle – Process execution consists of a cycle of CPU and I/O wait
- CPU burst followed by I/O burst
- CPU burst distribution is of main concern

**CPU Scheduler**

- Short-term scheduler selects from among the processes in ready queue allocates the CPU to one of them Queue may be ordered in various ways
- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- Scheduling under 1 and 4 is nonpreemptive
- All other scheduling is preemptive
- Consider access to shared data
- Consider preemption while in kernel mode
- Consider interrupts occurring during crucial OS activities

**Dispatcher**

- Dispatcher module gives control of the CPU to the process selected by short-term scheduler; this involves:
  - switching context
  - switching to user mode
  - jumping to the proper location in the user program to restart that program
- Dispatch latency – time it takes for the dispatcher to stop one process and start another

```

        }
        @Override
        public void
mouseClicked(MouseEvent e) {
            studyOS fstudyOS =
new studyOS();
            setVisible(false);

            fstudyOS.setVisible(true);
        }
    });
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

    TextArea textArea = new TextArea();
    textArea.setForeground(new Color(0, 0,
0));
    textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
    textArea.setBackground(new Color(255,
255, 255));
    textArea.setBounds(10, 227, 430, 492);
    textArea.setEditable(false);

    contentPane.add(textArea);
    //to display the text from the text file to
the text area
    try {
        BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch6OS.txt"));
        String line;
        StringBuilder sb = new StringBuilder();

        while ((line = reader.readLine()) != null) {
            String[] words = line.trim().split("\s+");
            int lineLength = 0;

            for (String word : words) {
                if (lineLength + word.length() > 80) {
                    sb.append("\n");
                    lineLength = 0;
                }

                sb.append(word).append(" ");
                lineLength += word.length() + 1;
            }

            sb.append("\n");
        }

        reader.close();
    }
}

```

```

        textArea.setText(sb.toString());
    } catch (IOException e) {
        e.printStackTrace();
    }
    JLabel bglabel = new JLabel("");
    bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450, 730);
    contentPane.add(bglabel);
}
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class studyMICRO extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        studyMICRO frame = new studyMICRO();
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public studyMICRO() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent
e) {

```

## Microprocessor Materials

The interface features a dark-themed header with a back arrow icon. Below it is a large orange button labeled "SELECT CHAPTER". Underneath are six dark rectangular boxes, each containing a chapter title. The background has a subtle circuit board pattern.

- CHAPTER 1 : Introduction**
- CHAPTER 2 : Microprocessor Architecture**
- CHAPTER 3 : Addressing Modes**
- CHAPTER 4 : Data Movement**
- CHAPTER 5 : Arithmetic and Logic**
- CHAPTER 6 : Program Control**

```

        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
{
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
}
    @Override
    public void mouseClicked(MouseEvent
e) {
        study fStudy = new study();
        setVisible(false);
        fStudy.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/ch.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

 JButton ch1btn = new JButton("");
ch1btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
        ch1MICROstudy fCh1SDstudy =
new ch1MICROstudy();
        setVisible(false);
        fCh1SDstudy.setVisible(true);
    }
});
ch1btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/OSch1.png
")));
ch1btn.setBounds(0, 295, 450, 61);
contentPane.add(ch1btn);

 JButton ch2btn = new JButton("");
ch2btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {

```

```

new ch2MICROstudy();
    setVisible(false);
    fCh2SDstudy.setVisible(true);
}
});
ch2btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/MICROch2.
png")));
ch2btn.setBounds(0, 360, 450, 61);
contentPane.add(ch2btn);

JButton ch3btn = new JButton("");
ch3btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch3MICROstudy();
    setVisible(false);
    fCh3SDstudy.setVisible(true);
}
});
ch3btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/MICROch3.
png")));
ch3btn.setBounds(0, 426, 450, 61);
contentPane.add(ch3btn);

JButton ch4btn = new JButton("");
ch4btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch4MICROstudy();
    setVisible(false);
    fCh4SDstudy.setVisible(true);
}
});
ch4btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/MICROch4.
png")));
ch4btn.setBounds(0, 491, 450, 61);
contentPane.add(ch4btn);

JButton ch5btn = new JButton("");
ch5btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent
e) {
new ch5MICROstudy();
    setVisible(false);
    fCh5SDstudy.setVisible(true);
}
}}
);

```

```

        });
        ch5btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/MICROch5.
png")));
        ch5btn.setBounds(0, 557, 450, 61);
        contentPane.add(ch5btn);

        JButton ch6btn = new JButton("");
        ch6btn.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent
e) {
new ch6SDstudy();
                ch6SDstudy fCh6SDstudy =
                    setVisible(false);
                    fCh6SDstudy.setVisible(true);
            }
        });
        ch6btn.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/MICROch6.
png")));
        ch6btn.setBounds(0, 622, 450, 61);
        contentPane.add(ch6btn);

        JLabel studyMICROlabel = new JLabel("");
        studyMICROlabel.setIcon(new
ImageIcon(studyMICRO.class.getResource("/images/bg.png")));
        studyMICROlabel.setBounds(0, 0, 450, 730);
        contentPane.add(studyMICROlabel);
    }
}

```

```

public class ch1MICROstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch1MICROstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");

```

Chapter 1 Microprocessor Material

```

        chapterlabel.setIcon(new
ImageIcon(ch1MICROstudy.class.getResource("/images/OSch1.
png")));
        chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studyMICRO
fstudyMICRO = new studyMICRO();
        setVisible(false);

        fstudyMICRO.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

        contentPane.add(textArea);
//to display the text from the text file to
the text area
try {

```

**CHAPTER 1: Introduction**

**1-1 A HISTORICAL BACKGROUND**

- Events leading to development of the microprocessor.
- 80X86, Pentium, Pentium Pro, Pentium III, Pentium 4, and Core2 microprocessors.
- While not essential to understand the microprocessor, furnishes:
  - interesting reading
  - historical perspective of fast-paced evolution

**The Mechanical Age**

- Idea of computing system not new.
- Calculating with a machine dates to 500 BC.
- Ancient people invented the abacus.
- first mechanical calculator
- strings of beads perform calculations
- Used by ancient priests to keep track of storehouses of grain.
- still in use today
- In 1642 mathematician Blaise Pascal invented a calculator constructed of wheels.
- each gear contained 10 teeth
- When moved one complete revolution, a second gear advances one position
- same principle used in automobile odometer
- Basis of all mechanical calculators.
- PASCAL programming language is named in honor of Blaise Pascal.
- First practical geared mechanical machines to compute information developed in early 1800s.
- humans dreamed of mechanical machines that could compute with a minimum of effort
- One early pioneer of mechanical computing machinery was Charles Babbage
- aided by Ada Byron, Countess of Lovelace
- Commissioned in 1823 by Royal Astronomical Society to build program-controlled computing machine

```

BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch1MICRO.txt"));
String line;
StringBuilder sb = new StringBuilder();

while ((line = reader.readLine()) != null) {
    String[] words = line.trim().split("\\s+");
    int lineLength = 0;

    for (String word : words) {
        if (lineLength + word.length() > 80) {
            sb.append("\n");
            lineLength = 0;
        }

        sb.append(word).append(" ");
        lineLength += word.length() + 1;
    }

    sb.append("\n");
}

reader.close();
textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
}

JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}
}

```

```

public class ch2MICROstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch2MICROstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
    }
}

```

## Chapter 2 Microprocessor Material

```

contentPane.setLayout(null);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch2MICROstudy.class.getResource("/images/MICRO
ch2.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

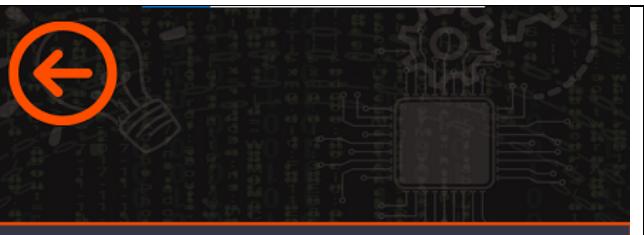
JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studyMICRO
fstudyMICRO = new studyMICRO();
        setVisible(false);

        fstudyMICRO.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);

```



## CHAPTER 2 : Microprocessor Architecture

**INTERNAL MICROPROCESSOR ARCHITECTURE**

- Before a program is written or instruction investigated, internal configuration of the microprocessor must be known.
- In a multiple core microprocessor each core contains the same program model.
- Each core runs a separate task or thread simultaneously. A thread contains a program counter, a register set, and a stack space. A task shares with other threads its code section, data section, and operating system.
- The Programming Model
  - 8086 through Core2 considered program visible.
  - registers are used during programming and are specified by the instructions
  - Other registers considered to be program invisible.
  - not addressable directly during applications
- Programming
  - 80286 and above contain program-invisible registers to control and operate protected memory.
  - and other features of the microprocessor
  - 80386 through Core2 microprocessors contain full 32-bit internal architectures.
  - 8086 through the 80286 are fully upward compatible to the 80386 through Core2 microprocessor.
  - including the 64-bit extensions
- Multipurpose Registers
  - RAX - a 64-bit register (RAX), a 32-bit register(accumulator) (EAX), a 16-bit register (AX), or as either of two 8-bit registers (AH and AL).
  - The accumulator is used for instructions such as multiplication, division, and some of the adjustment instructions.
  - Intel plans to expand the address bus to 52 bits to address 4P (252-peta) bytes of memory.

```

//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch2MICRO.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglab.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}
}

```

```

public class ch3MICROstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch3MICROstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
    }
}

```

Chapter 3 Microprocessor Material

```

contentPane = new JPanel();
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch3MICROstudy.class.getResource("/images/MICRO
ch3.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studyMICRO
fstudyMICRO = new studyMICRO();
        setVisible(false);

        fstudyMICRO.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));
textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);

```

**CHAPTER 3 : Addressing Modes**

**Assembly Language**

Each statement in an assembly language program consists of four parts. The leftmost field is called the label used to identify the name of a memory location used for storing data and for other purposes. All labels must begin with one of the following special characters: @, \$, -, or ?. A label may have a length from 1 to 35 characters. The next field to the right is the opcode or operation code designed to hold the instruction, or opcode the MOV instruction. The move data instruction is an example of an opcode.

Right of the opcode field is the operand field contains information used by the opcode. The MOV AL,BL instruction has the opcode MOV and operands AL and BL.

The comment field, the final field, contains a comment about the instruction(s). Comments always begin with a semicolon(;).

**3-1DATA ADDRESSING MODES**

MOV instruction is a common and flexible instruction provides a basis for explanation of data-addressing modes. Source is to the right and destination is to the left, next to the opcode. An opcode, or operation code, tells the microprocessor which operation to perform. MOV really moves nothing, it moves the source into the destination. It probably should be named COP for copy, it is not.

These data-addressing modes are found with all versions of the Intel microprocessor except for the scaled-index-addressing mode, found only through Core2.

**I-Register Addressing**

The most common form of data addressing once register names learned apply. The microprocessor contains these 8-bit register names used with addressing: AH, AL, BH, BL, CH, CL, DH, and DL. 16-bit register names are CX, DX, SP, BP, SI, and DI. In 80386 & above, extended 32-bit registers are: EAX, EBX, ECX, EDX, ESP, EBP, EDI, and ESI.

Important for instructions to use registers that are the same size. Never use AH with CL, or DI with SI.

```

        textArea.setEditable(false);

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch3MICRO.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

```

```

public class ch4MICROstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch4MICROstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
    }
}

```

Chapter 4 Microprocessor Material

```

        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel chapterlabel = new JLabel("");
        chapterlabel.setIcon(new
ImageIcon(ch4MICROstudy.class.getResource("/images/MICRO
ch4.png")));
        chapterlabel.setBounds(0, 145, 450, 74);
        contentPane.add(chapterlabel);

        JButton backbtn = new JButton("");
        backbtn.setOpaque(false);
        backbtn.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                studyMICRO
fstudyMICRO = new studyMICRO();
                setVisible(false);

                fstudyMICRO.setVisible(true);
            }
        });
        backbtn.setBackground(Color.WHITE);
        backbtn.setBorder(null);
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        backbtn.setBounds(10, 11, 78, 74);
        contentPane.add(backbtn);

        TextArea textArea = new TextArea();
        textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
    }
}

```

## CHAPTER 4 : Data Movement

**Introduction**

- This chapter concentrates on the data movement instructions.
- The data movement instructions include MOV, MOVSX, MOVZX, PU XCHG, XLAT, IN, OUT, LEA, LDS, LES, LFS, LGS, LSS, LAHF, SAHF
- String instructions: MOVS, LODS, STOS, INS, and OUTS.
- Explain the operation of each data movement instruction with application addressing modes.
- Explain the purposes of the assembly language pseudo-operations such as ALIGN, ASSUME, DB, DD, DW, END, ENDS, ENDP, EQU, .PROC, PTR, SEGMENT, USEI6, USE32, and USES.

(cont.)

- Select the appropriate assembly language instruction to accomplish data movement task.
- Determine the symbolic opcode, source, destination, and addressing hexdecimal machine language instruction.
- Use the assembler to set up a data segment, stack segment, and code segment.
- Show how to set up a procedure using PROC and ENDP.
- Explain the difference between memory models and full-segment definition in the MASM assembler.
- Use the Visual online assembler to perform data movement tasks.

4-1 MOV Revisited

- In this chapter, the MOV instruction introduces machine language instructions available with various addressing modes and instructions.
- It may be necessary to interpret machine language programs generated by the assembler.
- Occasionally, machine language patches are made by using the DEBUG command available with DOS and Visual for Windows.

```

textArea.setBackground(new Color(255,
255, 255));
textArea.setBounds(10, 227, 430, 492);
textArea.setEditable(false);

contentPane.add(textArea);
//to display the text from the text file to
the text area
try {
    BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch4MICRO.txt"));
    String line;
    StringBuilder sb = new StringBuilder();

    while ((line = reader.readLine()) != null) {
        String[] words = line.trim().split("\\s+");
        int lineLength = 0;

        for (String word : words) {
            if (lineLength + word.length() > 80) {
                sb.append("\n");
                lineLength = 0;
            }

            sb.append(word).append(" ");
            lineLength += word.length() + 1;
        }

        sb.append("\n");
    }

    reader.close();
    textArea.setText(sb.toString());
} catch (IOException e) {
    e.printStackTrace();
}
JLabel bglabel = new JLabel("");
bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}
}

```

```

public class ch5MICROstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch5MICROstudy() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);
    }
}

```

Chapter 5 Microprocessor Material

```

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 450, 730);
 setLocationRelativeTo(null);
 setResizable(false);
contentPane = new JPanel();

setContentPane(contentPane);
contentPane.setLayout(null);

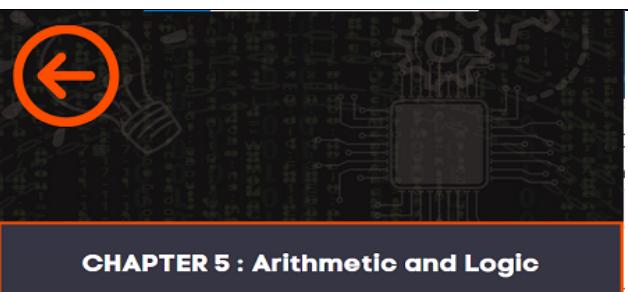
JLabel chapterlabel = new JLabel("");
chapterlabel.setIcon(new
ImageIcon(ch5MICROstudy.class.getResource("/images/MICRO
ch5.png")));
chapterlabel.setBounds(0, 145, 450, 74);
contentPane.add(chapterlabel);

JButton backbtn = new JButton("");
backbtn.setOpaque(false);
backbtn.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        studyMICRO
fstudyMICRO = new studyMICRO();
        setVisible(false);

        fstudyMICRO.setVisible(true);
    }
});
backbtn.setBackground(Color.WHITE);
backbtn.setBorder(null);
backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
backbtn.setBounds(10, 11, 78, 74);
contentPane.add(backbtn);

TextArea textArea = new TextArea();
textArea.setForeground(new Color(0, 0,
0));

```



## CHAPTER 5 : Arithmetic and Logic

**Introduction**

- We examine the arithmetic and logic instructions. The arithmetic instructions include addition, subtraction, multiplication, division, comparison, negative increment, and decrement.
- The logic instructions include AND, OR, Exclusive-OR, NOT, shifts, rotations, and the logical compare (TEST).

**5-1 ADDITION, SUBTRACTION AND COMPARISON**

- The bulk of the arithmetic instructions found in any microprocessor involve addition, subtraction, and comparison.
- Addition, subtraction, and comparison instructions are illustrated.
- Also shown are their uses in manipulating register and memory data.

**Addition**

- Addition (ADD) appears in many forms in the microprocessor.
- A second form of addition, called add-with-carry, is introduced with the ADC instruction.
- The only types of addition not allowed are memory-to-memory and self register.
- Segment registers can only be moved, pushed, or popped.
- Increment instruction (INC) is a special type of addition that adds 1 to a number.

**Register Addition**

- When arithmetic and logic instructions execute, contents of the flag registers change.
- Interrupt, trap, and other flags do not change.
- Any ADD instruction modifies the contents of the sign, zero, carry, auxiliary carry, parity, and overflow flags.

```

        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
        textArea.setEditable(false);

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch5MICRO.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

```

```

public class ch6SDstudy extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ch6SDstudy() {
        // Set the icon image of the frame
    }
}

```

Chapter 6 Microprocessor Material

```

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel chapterlabel = new JLabel("");
    chapterlabel.setIcon(new ImageIcon(ch6SDstudy.class.getResource("/images/MICROch6.png")));
    chapterlabel.setBounds(0, 145, 450, 74);
    contentPane.add(chapterlabel);

    JButton backbtn = new JButton("");
    backbtn.setOpaque(false);
    backbtn.addMouseListener(new
MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
buttonhover.png")));
        }
        @Override
        public void
mouseExited(MouseEvent e) {
            backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
        }
        @Override
        public void
mouseClicked(MouseEvent e) {
            studyMICRO
fstudyMICRO = new studyMICRO();
            setVisible(false);

            fstudyMICRO.setVisible(true);
        }
    });
    backbtn.setBackground(Color.WHITE);
    backbtn.setBorder(null);
    backbtn.setIcon(new
ImageIcon(quizmenu.class.getResource("/images/back
button.png")));
    backbtn.setBounds(10, 11, 78, 74);
    contentPane.add(backbtn);

```

## CHAPTER 6 : Program Control

**Introduction**

- This chapter explains the program control instructions, including the `calls`, `returns`, `interrupts`, and `machine control instructions`.
- This chapter also presents the relational assembly language statements (`.ELSE`, `.ELSEIF`, `.ENDIF`, `.WHILE`, `.ENDW`, `.REPEAT`, and `.UNTIL`) that version 6.xx and above of MASM or TASM, with version 5.xx set for MASM compatibility.

**6-1 THE JUMP GROUP**

- Allows programmer to skip program sections and branch to any part of the next instruction.
- A conditional jump instruction allows decisions based upon numerical results are held in the flag bits, then tested by conditional jump instructions
- LOOP and conditional LOOP are also forms of the jump instruction.

**Unconditional Jump (JMP)**

- Three types: short jump, near jump, far jump.
- Short jump is a 2-byte instruction that allows jumps or branches to memory locations within +127 and -128 bytes.
- from the address following the jump
- 3-byte near jump allows a branch or jump within ±32K bytes from the instruction in the current code segment.
- 5-byte far jump allows a jump to any memory location within the real system.
- The short and near jumps are often called intrasegment jumps.
- Far jumps are called intersegment jumps.

**Short Jump**

```

        TextArea textArea = new TextArea();
        textArea.setForeground(new Color(0, 0,
0));
        textArea.setFont(new Font("Arial",
Font.PLAIN, 13));
        textArea.setBackground(new Color(255,
255, 255));
        textArea.setBounds(10, 227, 430, 492);
        textArea.setEditable(false);

        contentPane.add(textArea);
        //to display the text from the text file to
the text area
        try {
            BufferedReader reader = new
BufferedReader(new FileReader("studyfiles/ch6MICRO.txt"));
            String line;
            StringBuilder sb = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] words = line.trim().split("\\s+");
                int lineLength = 0;

                for (String word : words) {
                    if (lineLength + word.length() > 80) {
                        sb.append("\n");
                        lineLength = 0;
                    }

                    sb.append(word).append(" ");
                    lineLength += word.length() + 1;
                }

                sb.append("\n");
            }

            reader.close();
            textArea.setText(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
        JLabel bglabel = new JLabel("");
        bglabel.setIcon(new
ImageIcon(Home.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;

```

Result for Software Design quiz

```

public class Result_SD extends JFrame {

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        Result_SD frame = new Result_SD();
        frame.setVisible(true);
    }

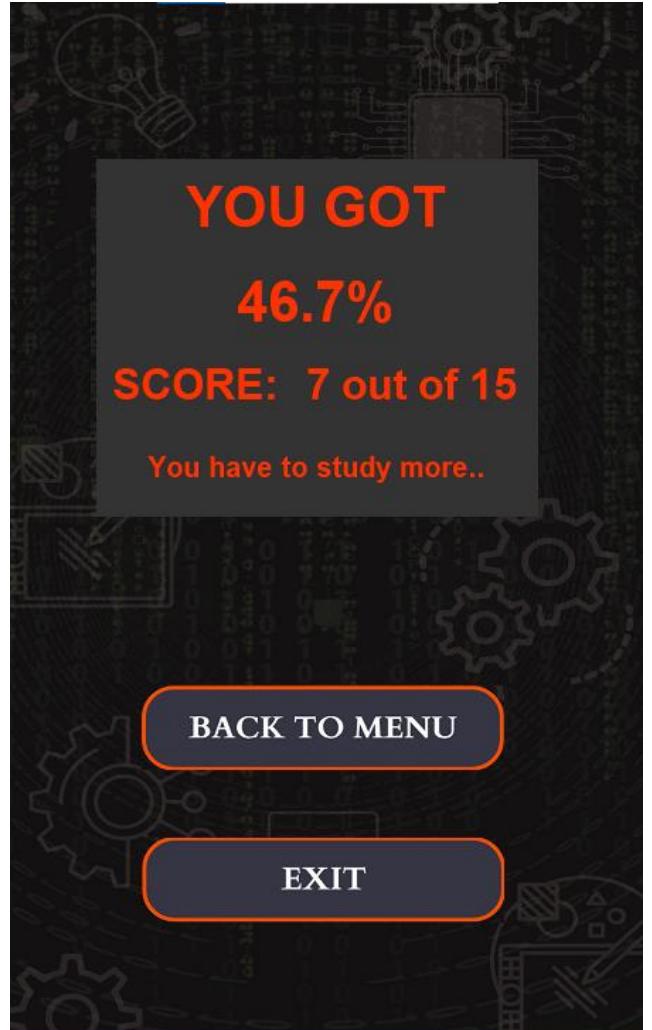
    public Result_SD() {
        // Set the icon image of the frame

        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        JPanel contentPane;
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        //back button
        JButton btnbckmenu = new JButton("");
        btnbckmenu.setBorder(null);
        btnbckmenu.addMouseListener(new
        MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent
e) {
                btnbckmenu.setIcon(new
                ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
            }
            @Override
            public void mouseExited(MouseEvent e)
{
                btnbckmenu.setIcon(new
                ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
            }
            @Override
            public void mouseClicked(MouseEvent
e) {
                Home fhome = new Home();
                setVisible(false);
                fhome.setVisible(true);
                Home.clip.setFramePosition(0);
                Home.clip.start();
            }
        })
    }
}

```



```

        });
        btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
        btnbckmenu.setBounds(94, 483, 256, 60);
        contentPane.add(btnbckmenu);

        //exit button
        JButton btnexit = new JButton("");
        btnexit.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent
e) {
                btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
            }
            @Override
            public void mouseExited(MouseEvent e)
{
                btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
            }
            @Override
            public void mouseClicked(MouseEvent
e) {
                System.exit(0);
            }
        });
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256, 60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }

    class Result_ch1SD extends JFrame {

        private JPanel contentPane;
        private JTextField txtYouGot;
        private JTextField accuracytxt;
        private JTextField txtScore;
        private JTextField scoretxt;
        private JTextField commenttxt;

        /**
         * Create the frame.
         */

```

```
public Result_ch1SD() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);
    setVisible(true);

    txtYouGot = new JTextField();
    txtYouGot.setBorder(null);
    txtYouGot.setEditable(false);
    txtYouGot.setFont(new Font("Arial",
Font.BOLD, 40));
    txtYouGot.setForeground(new
Color(255, 51, 0));

    txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
    txtYouGot.setText("YOU GOT");
    txtYouGot.setBackground(new Color(51,
51, 51));
    txtYouGot.setBounds(62, 111, 312, 69);
    contentPane.add(txtYouGot);
    txtYouGot.setColumns(10);

    txtScore = new JTextField();
    txtScore.setText("SCORE:");
    txtScore.setBorder(null);
    txtScore.setEditable(false);

    txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
    txtScore.setForeground(new Color(255,
51, 0));
    txtScore.setFont(new Font("Arial",
Font.BOLD, 30));
    txtScore.setColumns(10);
    txtScore.setBackground(new Color(51,
51, 51));
    txtScore.setBounds(62, 245, 141, 53);
    contentPane.add(txtScore);

    scoretxt = new JTextField();
    scoretxt.setBorder(null);
    scoretxt.setEditable(false);

    scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);
```

```
scoretxt.setForeground(new Color(255,
51, 0));
scoretxt.setFont(new Font("Arial",
Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51,
51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CE
NTER);
commenttxt.setForeground(new
Color(255, 51, 0));
commenttxt.setFont(new Font("Arial",
Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new
Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312,
69);
contentPane.add(commenttxt);

//call the score for java
double score = SDquiz.ch1_SD.scores;
int whole_score = (int) score ;

String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + " out of
10");
//to get the percentage
double accuracy = ((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {

commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){
commenttxt.setText("Keep up
the good work!");
}
else if(whole_score<=5 &&
whole_score>=3){
commenttxt.setText("You have
to study more..");
}
```

```

        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);

        accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
        accuracytxt.setForeground(new
Color(255, 51, 0));
        accuracytxt.setFont(new Font("Arial",
Font.BOLD, 40));
        accuracytxt.setColumns(10);
        accuracytxt.setBackground(new
Color(51, 51, 51));
        accuracytxt.setBounds(62, 178, 312, 69);
        String accuracytext =
df.format(accuracy);
        accuracytxt.setText(accuracytext + "%");
        contentPane.add(accuracytxt);

        JButton bnbckmenu = new JButton("");
        bnbckmenu.setBorder(null);
        bnbckmenu.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {

                bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {

                bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                Home fhome = new
Home();
                fhome.setVisible(false);
                fhome.setVisible(true);

                Home.clip setFramePosition(0);
                Home.clip.start();
            }
        })
    }
}

```

```

        });
        btbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
        btbckmenu.setBounds(94, 483, 256,
60);
        contentPane.add(btbckmenu);

        JButton btnexit = new JButton("");
        btnexit.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256, 60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

class Result_ch2SD extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
}

```

```
public Result_ch2SD() {
    // Set the icon image of the
frame
    setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);
    setVisible(true);

    txtYouGot = new JTextField();
    txtYouGot.setBorder(null);
    txtYouGot.setEditable(false);
    txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
    txtYouGot.setForeground(new
Color(255, 51, 0));

    txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
    txtYouGot.setText("YOU GOT");
    txtYouGot.setBackground(new
Color(51, 51, 51));
    txtYouGot.setBounds(62, 111,
312, 69);
    contentPane.add(txtYouGot);
    txtYouGot.setColumns(10);

    txtScore = new JTextField();
    txtScore.setText("SCORE:");
    txtScore.setBorder(null);
    txtScore.setEditable(false);

    txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
    txtScore.setForeground(new
Color(255, 51, 0));
    txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
    txtScore.setBackground(new
Color(51, 51, 51));
    txtScore.setBounds(62, 245,
141, 53);
    contentPane.add(txtScore);

    scoretxt = new JTextField();
    scoretxt.setBorder(null);
    scoretxt.setEditable(false);
```

```

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
SDquiz.ch2_SD.scores;
int whole_score = (int) score ;

String final_score =
Integer.toString(whole_score);

out of 10");
scoretxt.setText(final_score + " "
//to get the percentage
double accuracy =
((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {

commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){

commenttxt.setText("Keep up the good work!");
}

```

```

        }
        else if(whole_score<=5 &&
whole_score>=3){

            commenttxt.setText("You have to study more..");
        }
        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);

        accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
        accuracytxt.setForeground(new
Color(255, 51, 0));
        accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
        accuracytxt.setColumns(10);
        accuracytxt.setBackground(new
Color(51, 51, 51));
        accuracytxt.setBounds(62, 178,
312, 69);
        String accuracytext =
df.format(accuracy);
        accuracytxt.setText(accuracytext
+ "%");
        contentPane.add(accuracytxt);

        JButton btnbckmenu = new
JButton("");
        btnbckmenu.setBorder(null);

        btnbckmenu.addMouseListener(new MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {

                btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {

                btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
            }
            @Override

```

```

        public void
mouseClicked(MouseEvent e) {
new Home();
                Home fhome =
setVisible(false);

fhome.setVisible(true);

Home.clip setFramePosition(0);

Home.clip.start();
}
});
btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btnbckmenu);

JButton btnexit = new
 JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
@Override
public void
mouseEntered(MouseEvent e) {

btnclick.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
}
@Override
public void
mouseExited(MouseEvent e) {

btnclick.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
}
@Override
public void
mouseClicked(MouseEvent e) {
System.exit(0);
}
});
btnclick.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
btnclick.setBorder(null);
btnclick.setBounds(94, 591, 256,
60);
contentPane.add(btnclick);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));

```

```
730);
        bglabel.setBounds(0, 0, 450,
    contentPane.add(bglabel);
    }
}
class Result_ch3SD extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch3SD() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");
        txtScore.setBorder(null);
        txtScore.setEditable(false);
```

```
txtScore.setHorizontalAlignment(SwingConstants.CENTER);
txtScore.setForeground(new Color(255, 51, 0));
txtScore.setFont(new Font("Arial", Font.BOLD, 30));
txtScore.setColumns(10);
txtScore.setBackground(new Color(51, 51, 51));
txtScore.setBounds(62, 245, 141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
int whole_score = (int) score ;
String final_score =
Integer.toString(whole_score);
```

```

        scoretxt.setText(final_score + "
out of 15");

        //to get the percentage
        double accuracy =
((score/15)*100);
        DecimalFormat df = new
DecimalFormat("#.#");
        //for rating
        if (whole_score>=13) {

            commenttxt.setText("Excellent!");

        } else if(whole_score<=12 &&
whole_score>=9){

            commenttxt.setText("Keep up the good work!");
        }
        else if(whole_score<=8 &&
whole_score>=6){

            commenttxt.setText("You have to study more.. ");
        }
        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);

        accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
        accuracytxt.setForeground(new
Color(255, 51, 0));
        accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
        accuracytxt.setColumns(10);
        accuracytxt.setBackground(new
Color(51, 51, 51));
        accuracytxt.setBounds(62, 178,
312, 69);
        String accuracytext =
df.format(accuracy);
        accuracytext += "%");
        accuracytxt.setText(accuracytext);
        contentPane.add(accuracytxt);

        JButton btnbckmenu = new
JButton("");
        btnbckmenu.setBorder(null);

        btnbckmenu.addMouseListener(new MouseAdapter() {
            @Override

```

```

public void
mouseEntered(MouseEvent e) {
    bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
}
@Override
public void
mouseExited(MouseEvent e) {
    bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
}
@Override
public void
mouseClicked(MouseEvent e) {
    Home fhome =
new Home();
    setVisible(false);

    fhome.setVisible(true);
    Home.clip setFramePosition(0);
    Home.clip.start();
}
});
bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
bnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(bnbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
    }
}
@Override

```

```

        public void
mouseClicked(MouseEvent e) {
            System.exit(0);
        }
    });
    btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
    btnexit.setBorder(null);
    btnexit.setBounds(94, 591, 256,
60);
    contentPane.add(btnexit);

    JLabel bglabel = new JLabel();
    bglabel.setOpaque(true);
    bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
    bglabel.setBounds(0, 0, 450,
730);
    contentPane.add(bglabel);
}
}

class Result_ch4SD extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch4SD() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));

```

```

txtYouGot.setForeground(new
Color(255, 51, 0));

txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);

Color(51, 51, 51));
312, 69);
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);

Color(255, 51, 0));
Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);

Color(255, 51, 0));
Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CE
NTER);

Color(255, 51, 0));
commenttxt.setForeground(new

```

```

commenttxt.setFont(new
Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new
Color(51, 51, 51));
commenttxt.setBounds(62, 295,
312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
int whole_score = (int) score ;
String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + "
out of 10");
//to get the percentage
double accuracy =
((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {

    commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){

    commenttxt.setText("Keep up the good work!");
}
else if(whole_score<=5 &&
whole_score>=3){

    commenttxt.setText("You have to study more..");
}
else {
    commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
    commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));

```

```

accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));

Color(51, 51, 51));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
accuracytxt.setBounds(62, 178,
312, 69);
String accuracytext =
df.format(accuracy);
accuracytxt.setText(accuracytext
+ "%");
contentPane.add(accuracytxt);

JButton btnbckmenu = new
 JButton("");
btnbckmenu.setBorder(null);

btnbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

    btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
}
@Override
public void
mouseExited(MouseEvent e) {

    btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
}
@Override
public void
mouseClicked(MouseEvent e) {
    Home fhome =
new Home();
    fhome.setVisible(false);

    fhome.setVisible(true);

    Home.clip setFramePosition(0);

    Home.clip.start();
}
});
btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btnbckmenu);

```

```

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void
mouseExited(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}
class Result_ch5SD extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch5SD() {
        // Set the icon image of the
frame

```

```
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");
        txtScore.setBorder(null);
        txtScore.setEditable(false);

        txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
        txtScore.setForeground(new
Color(255, 51, 0));
        txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
        txtScore.setBackground(new
Color(51, 51, 51));
        txtScore.setBounds(62, 245,
141, 53);
        contentPane.add(txtScore);

        scoretxt = new JTextField();
        scoretxt.setBorder(null);
        scoretxt.setEditable(false);

        scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);
```

```

Color(255, 51, 0));
Font("Arial", Font.BOLD, 30);

Color(51, 51, 51));
176, 53);
scoretxt.setForeground(new
scoretxt.setFont(new
scoretxt.setColumns(10);
scoretxt.setBackground(new
scoretxt.setBounds(198, 245,
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new
Color(255, 51, 0));
commenttxt.setFont(new
Font("Arial", Font.BOLD, 20));

Color(51, 51, 51));
312, 69);
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new
commenttxt.setBounds(62, 295,
contentPane.add(commenttxt);

//call the score for java
double score =
SDquiz.ch5_SD.scores;
int whole_score = (int) score ;
String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + "
out of 10");
//to get the percentage
double accuracy =
((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {

commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){

commenttxt.setText("Keep up the good work!");
}
else if(whole_score<=5 &&
whole_score>=3){
```

```

        commenttxt.setText("You have to study more..");
    }
    else {
        commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
        commenttxt.setText("It's
okay...Study more,\nyou can do it!");
    }
    accuracytxt = new JTextField();
    accuracytxt.setBorder(null);
    accuracytxt.setEditable(false);

    accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
    accuracytxt.setForeground(new
Color(255, 51, 0));
    accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
    accuracytxt.setColumns(10);
    accuracytxt.setBackground(new
Color(51, 51, 51));
    accuracytxt.setBounds(62, 178,
312, 69);
    String accuracytext =
df.format(accuracy);
    + "%");
    accuracytxt.setText(accuracytext
+ contentPane.add(accuracytxt);

    JButton btbckmenu = new
JButton("");
    btbckmenu.setBorder(null);

    btbckmenu.addMouseListener(new MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

            btbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
        }
        @Override
        public void
mouseExited(MouseEvent e) {

            btbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.bn
g")));
        }
        @Override
        public void
mouseClicked(MouseEvent e) {
            Home fhome =
new Home();

```

```

        setVisible(false);

fhome.setVisible(true);

Home.clip.setFramePosition(0);

Home.clip.start();
    }
});

btnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btnbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

    btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void
mouseExited(MouseEvent e) {

    btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}

```

```

        }

class Result_ch6SD extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoreetxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch6SD() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");
        txtScore.setBorder(null);
        txtScore.setEditable(false);

        txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);

```

```
Color(255, 51, 0));  
Font("Arial", Font.BOLD, 30));  
  
Color(51, 51, 51));  
141, 53);  
  
txtScore.setForeground(new  
txtScore.setFont(new  
txtScore.setColumns(10);  
txtScore.setBackground(new  
txtScore.setBounds(62, 245,  
contentPane.add(txtScore);  
  
scoretxt = new JTextField();  
scoretxt.setBorder(null);  
scoretxt.setEditable(false);  
  
scoretxt.setHorizontalAlignment(SwingConstants.CENTER);  
scoretxt.setForeground(new  
scoretxt.setFont(new  
scoretxt.setColumns(10);  
scoretxt.setBackground(new  
scoretxt.setBounds(198, 245,  
contentPane.add(scoretxt);  
  
commenttxt = new JTextField();  
commenttxt.setBorder(null);  
  
commenttxt.setHorizontalAlignment(SwingConstants.CENTER);  
commenttxt.setForeground(new  
commenttxt.setFont(new  
commenttxt.setEditable(false);  
commenttxt.setColumns(10);  
commenttxt.setBackground(new  
commenttxt.setBounds(62, 295,  
contentPane.add(commenttxt);  
  
//call the score for java  
double score =  
int whole_score = (int) score ;  
  
String final_score =  
Integer.toString(whole_score);  
  
scoretxt.setText(final_score + "  
out of 10");
```

```

        //to get the percentage
        double accuracy =
((score/10)*100);

        DecimalFormat df = new
DecimalFormat("#.#");
        //for rating
        if (whole_score>=8) {

            commenttxt.setText("Excellent!");

        } else if(whole_score<=7 &&
whole_score>=6){

            commenttxt.setText("Keep up the good work!");
        }
        else if(whole_score<=5 &&
whole_score>=3){

            commenttxt.setText("You have to study more.. ");
        }
        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorderStyle(null);
        accuracytxt.setEditable(false);

        accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
        accuracytxt.setForeground(new
Color(255, 51, 0));
        accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));

        Color(51, 51, 51));
        accuracytxt.setColumns(10);
        accuracytxt.setBackground(new
Color(51, 51, 51));
        accuracytxt.setBounds(62, 178,
312, 69);
        String accuracytext =
df.format(accuracy);
        accuracytxt.setText(accuracytext
+ "%");
        contentPane.add(accuracytxt);

        JButton btnbckmenu = new
JButton("");
        btnbckmenu.setBorder(null);

        btnbckmenu.addMouseListener(new MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {

```

```
        bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenuho
ver.png")));
    }
    @Override
    public void

mouseExited(MouseEvent e) {

    bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
    }
    @Override
    public void

mouseClicked(MouseEvent e) {
    Home fhome =
new Home();
    setVisible(false);

    fhome.setVisible(true);

    Home.clip setFramePosition(0);

    Home.clip.start();
}
});;
    bnbckmenu.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bcktomenu.pn
g")));
    bnbckmenu.setBounds(94, 483,
256, 60);
    contentPane.add(bnbckmenu);

    JButton btnexit = new
JButton("");
    btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void

mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void

mouseExited(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
    }
    @Override
    public void

mouseClicked(MouseEvent e) {
```

```

        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/ext.png")));
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_SD.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}
}

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;

public class Result_OS extends JFrame {

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        Result_OS frame = new Result_OS();
        frame.setVisible(true);

    }

    public Result_OS() {
        // Set the icon image of the frame

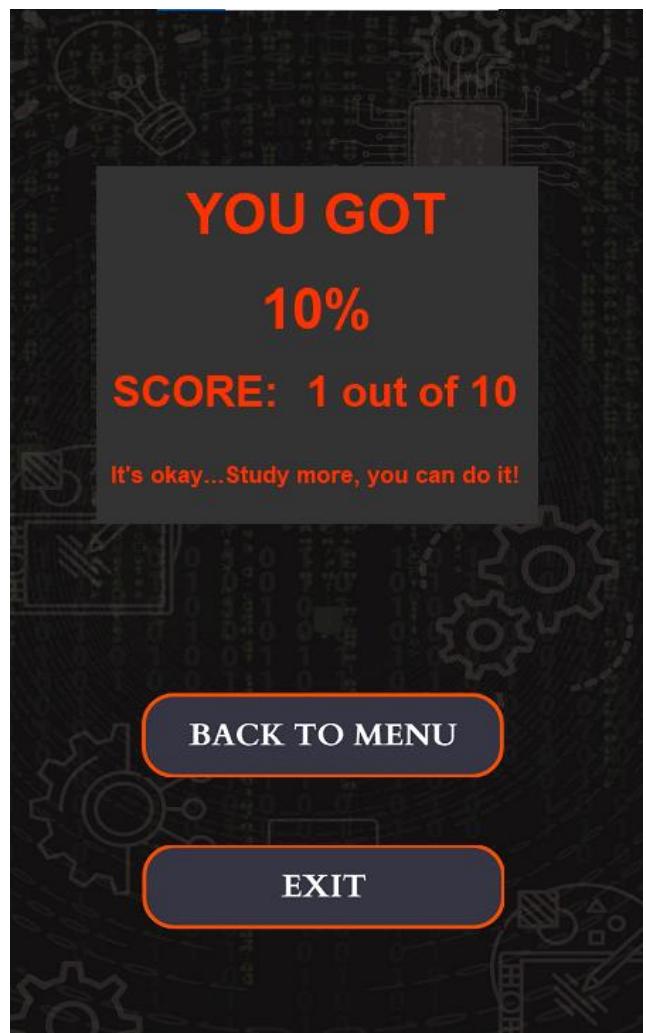
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        JPanel contentPane;
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton btnbckmenu = new JButton("");
        btnbckmenu.setBorder(null);

```

Result for Operating System quiz



```

        btnbckmenu.addMouseListener(new
MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent
e) {
                btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenuh
over.png")));
            }
            @Override
            public void mouseExited(MouseEvent e)
{
                btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
            }
            @Override
            public void mouseClicked(MouseEvent
e) {
                Home fhome = new Home();
                setVisible(false);
                fhome.setVisible(true);
                Home.clip setFramePosition(0);
                Home.clip.start();
            }
        });
        btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
        btnbckmenu.setBounds(94, 483, 256, 60);
        contentPane.add(btnbckmenu);

        JButton btnexit = new JButton("");
        btnexit.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent
e) {
                btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
            }
            @Override
            public void mouseExited(MouseEvent e)
{
                btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
            }
            @Override
            public void mouseClicked(MouseEvent
e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
        btnexit.setBorder(null);
    }
}

```

```

btnexit.setBounds(94, 591, 256, 60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}

class Result_ch1OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch1OS() {
        // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new Font("Arial",
Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new Color(51,
51, 51));
        txtYouGot.setBounds(62, 111, 312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);
    }
}

```

```
txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENTER);
txtScore.setForeground(new Color(255,
51, 0));
txtScore.setFont(new Font("Arial",
Font.BOLD, 30));
txtScore.setColumns(10);
txtScore.setBackground(new Color(51,
51, 51));
txtScore.setBounds(62, 245, 141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255,
51, 0));
scoretxt.setFont(new Font("Arial",
Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51,
51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new
Color(255, 51, 0));
commenttxt.setFont(new Font("Arial",
Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new
Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312,
69);
contentPane.add(commenttxt);

//call the score for java
double score = OSquiz.ch1_OS.scores;
int whole_score = (int) score ;
```

```

        String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + " out of
10");
//to get the percentage
double accuracy = ((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {

commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){
commenttxt.setText("Keep up
the good work!");
}
else if(whole_score<=5 &&
whole_score>=3){
commenttxt.setText("You have
to study more..");
}
else {
commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));
accuracytxt.setFont(new Font("Arial",
Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
Color(51, 51, 51));
accuracytxt.setBounds(62, 178, 312, 69);
String accuracytext =
df.format(accuracy);
accuracytxt.setText(accuracytext + "%");
contentPane.add(accuracytxt);

JButton btnbckmenu = new JButton("");
btnbckmenu.setBorder(null);
btnbckmenu.addMouseListener(new
MouseAdapter() {
@Override
public void
mouseEntered(MouseEvent e) {

```

```
btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenuho
ver.png")));
}
@Override
public void
mouseExited(MouseEvent e) {
    btbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
}
@Override
public void
mouseClicked(MouseEvent e) {
    Home fhome = new
Home();
    setVisible(false);
    fhome.setVisible(true);

    Home.clip setFramePosition(0);
    Home.clip.start();
}
});
btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
btnbckmenu.setBounds(94, 483, 256,
60);
contentPane.add(btbckmenu);

JButton btnexit = new JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {
        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void
mouseExited(MouseEvent e) {
        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
}
```

```

        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256, 60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

class Result_ch2OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoreetxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch2OS() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
    }
}

```

```
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENTER);
txtScore.setForeground(new Color(255, 51, 0));
txtScore.setFont(new Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
txtScore.setColumns(10);
txtScore.setBackground(new Color(51, 51, 51));
txtScore.setBounds(62, 245, 141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));

Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);
```

```

OSquiz.ch2_OS.scores;

//call the score for java
double score =
int whole_score = (int) score ;

String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + "
out of 15");

//to get the percentage
double accuracy =
((score/15)*100);

DecimalFormat df = new
DecimalFormat("#.#");

//for rating
if (whole_score>=13) {

    commenttxt.setText("Excellent!");

} else if(whole_score<=12 &&
whole_score>=9){

    commenttxt.setText("Keep up the good work!");
}
else if(whole_score<=8 &&
whole_score>=6){

    commenttxt.setText("You have to study more..");
}
else {
    commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
    commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));
accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
Color(51, 51, 51));
accuracytxt.setBounds(62, 178,
312, 69);
String accuracytext =
df.format(accuracy);
accuracytext.setText(accuracytext +
"%");
contentPane.add(accuracytxt);

```

```

JButton btnbckmenu = new
JButton("");
btnbckmenu.setBorder(null);

btnbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

    btbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenuho
ver.png")));
}

@Override
public void
mouseExited(MouseEvent e) {

    btbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
}

@Override
public void
mouseClicked(MouseEvent e) {
    Home fhome =
new Home();
    setVisible(false);

    fhome.setVisible(true);

    Home.clip setFramePosition(0);

    Home.clip.start();
}
});

btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
}
});

```

```

        public void
mouseExited(MouseEvent e) {
    btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
}
@Override
public void
mouseClicked(MouseEvent e) {
    System.exit(0);
}
});
btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}
class Result_ch3OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch3OS() {
        // Set the icon image of the
frame
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);
    }
}

```

```
txtYouGot = new JTextField();
txtYouGot.setBorder(null);
txtYouGot.setEditable(false);
txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
Color(255, 51, 0));

txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
Color(51, 51, 51));
312, 69);
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENTER);
Color(255, 51, 0));
Font("Arial", Font.BOLD, 30));
Color(51, 51, 51));
141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
Color(255, 51, 0));
Font("Arial", Font.BOLD, 30));
Color(51, 51, 51));
176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);
```

```

        commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
        commenttxt.setForeground(new Color(255, 51, 0));
        commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
        commenttxt.setEditable(false);
        commenttxt.setColumns(10);
        commenttxt.setBackground(new Color(51, 51, 51));
        commenttxt.setBounds(62, 295, 312, 69);
        contentPane.add(commenttxt);

        //call the score for java
        double score =
        int whole_score = (int) score ;
        String final_score =
        Integer.toString(whole_score);
        scoretxt.setText(final_score + " out of 10");
        //to get the percentage
        double accuracy =
        ((score/10)*100);
        DecimalFormat df = new DecimalFormat("#.#");
        //for rating
        if (whole_score>=8) {
            commenttxt.setText("Excellent!");
        } else if(whole_score<=7 && whole_score>=6){
            commenttxt.setText("Keep up the good work!");
        } else if(whole_score<=5 && whole_score>=3){
            commenttxt.setText("You have to study more.. ");
        } else {
            commenttxt.setFont(new Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);
    }
}

```

```
accuracytxt.setHorizontalAlignment(SwingConstants.CENTER);
accuracytxt.setForeground(new Color(255, 51, 0));
accuracytxt.setFont(new Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new Color(51, 51, 51));
accuracytxt.setBounds(62, 178, 312, 69);
String accuracytext =
df.format(accuracy);
accuracytext += "%");
contentPane.add(accuracytxt);

JButton btbckmenu = new JButton("");
btbckmenu.setBorder(null);

btbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenuhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenu.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        new Home();
        Home fhome =
        fhome.setVisible(false);
        fhome.setVisible(true);
        Home.clip.setFramePosition(0);
        Home.clip.start();
    }
});
btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenu.png")));

```

```

256, 60);
btnbckmenu.setBounds(94, 483,
contentPane.add(btnbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
    }
    @Override
    public void
mouseExited(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}
class Result_ch4OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.

```

```
/*
public Result_ch4OS() {
    // Set the icon image of the
frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
    setUndecorated(true);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);
    setVisible(true);

txtYouGot = new JTextField();
txtYouGot.setBorder(null);
txtYouGot.setEditable(false);
txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
    txtYouGot.setForeground(new
Color(255, 51, 0));

    txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
    txtYouGot.setText("YOU GOT");
    txtYouGot.setBackground(new
Color(51, 51, 51));
    txtYouGot.setBounds(62, 111,
312, 69);
    contentPane.add(txtYouGot);
    txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

    txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
    txtScore.setForeground(new
Color(255, 51, 0));
    txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
    txtScore.setColumns(10);
    txtScore.setBackground(new
Color(51, 51, 51));
    txtScore.setBounds(62, 245,
141, 53);
    contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
```

```

        scoretxt.setEditable(false);
        scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
        scoretxt.setForeground(new Color(255, 51, 0));
        scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
        scoretxt.setColumns(10);
        scoretxt.setBackground(new Color(51, 51, 51));
        scoretxt.setBounds(198, 245, 176, 53);
        contentPane.add(scoretxt);

        commenttxt = new JTextField();
        commenttxt.setBorder(null);

        commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
        commenttxt.setForeground(new Color(255, 51, 0));
        commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
        commenttxt.setEditable(false);
        commenttxt.setColumns(10);
        commenttxt.setBackground(new Color(51, 51, 51));
        commenttxt.setBounds(62, 295, 312, 69);
        contentPane.add(commenttxt);

        //call the score for java
        double score =
        int whole_score = (int) score ;
        String final_score =
        Integer.toString(whole_score);

        scoretxt.setText(final_score + " out of 10");
        //to get the percentage
        double accuracy =
        ((score/10)*100);
        DecimalFormat df = new
        DecimalFormat("#.#");
        //for rating
        if (whole_score>=8) {
            commenttxt.setText("Excellent!");
        } else if(whole_score<=7 &&
        whole_score>=6){
    
```

```

        commenttxt.setText("Keep up the good work!");
    }
    else if(whole_score<=5 &&
whole_score>=3){

        commenttxt.setText("You have to study more..");
    }
    else {
        commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
        commenttxt.setText("It's
okay...Study more,\nyou can do it!");
    }
    accuracytxt = new JTextField();
    accuracytxt.setBorder(null);
    accuracytxt.setEditable(false);

    accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
    accuracytxt.setForeground(new
Color(255, 51, 0));
    accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
    accuracytxt.setColumns(10);
    accuracytxt.setBackground(new
Color(51, 51, 51));
    accuracytxt.setBounds(62, 178,
312, 69);
    String accuracytext =
df.format(accuracy);
    accuracytxt.setText(accuracytext
+ "%");
    contentPane.add(accuracytxt);

    JButton btnbckmenu = new
JButton("");
    btnbckmenu.setBorder(null);

    btnbckmenu.addMouseListener(new MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

            btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenuho
ver.png")));
        }
        @Override
        public void
mouseExited(MouseEvent e) {

            btnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
        }
    })

```

```

        @Override
        public void
mouseClicked(MouseEvent e) {
            Home fhome =
new Home();
            setVisible(false);

            fhome.setVisible(true);

            Home.clip setFramePosition(0);

            Home.clip.start();
        }
    });
    bnbckmenu.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p
ng")));
    bnbckmenu.setBounds(94, 483,
256, 60);
    contentPane.add(bnbckmenu);

    JButton btnexit = new
JButton("");
    btnexit.addMouseListener(new
MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

            btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
        }
        @Override
        public void
mouseExited(MouseEvent e) {

            btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
        }
        @Override
        public void
mouseClicked(MouseEvent e) {
            System.exit(0);
        }
    });
    btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
    btnexit.setBorder(null);
    btnexit.setBounds(94, 591, 256,
60);
    contentPane.add(btnexit);

    JLabel bglabel = new JLabel();
    bglabel.setOpaque(true);

```

```

        bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450,
730);
        contentPane.add(bglabel);
    }
}

class Result_ch5OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch5OS() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");

```

```
        txtScore.setBorder(null);
        txtScore.setEditable(false);

        txtScore.setHorizontalAlignment(SwingConstants.CENTER);
        txtScore.setForeground(new Color(255, 51, 0));
        txtScore.setFont(new Font("Arial", Font.BOLD, 30));
        txtScore.setColumns(10);
        txtScore.setBackground(new Color(51, 51, 51));
        txtScore.setBounds(62, 245, 141, 53);
        contentPane.add(txtScore);

        scoretxt = new JTextField();
        scoretxt.setBorder(null);
        scoretxt.setEditable(false);

        scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
        scoretxt.setForeground(new Color(255, 51, 0));
        scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
        scoretxt.setColumns(10);
        scoretxt.setBackground(new Color(51, 51, 51));
        scoretxt.setBounds(198, 245, 176, 53);
        contentPane.add(scoretxt);

        commenttxt = new JTextField();
        commenttxt.setBorder(null);

        commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
        commenttxt.setForeground(new Color(255, 51, 0));
        commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
        commenttxt.setEditable(false);
        commenttxt.setColumns(10);
        commenttxt.setBackground(new Color(51, 51, 51));
        commenttxt.setBounds(62, 295, 312, 69);
        contentPane.add(commenttxt);

        //call the score for java
        double score =
        int whole_score = (int) score ;

OSquiz.ch5_OS.scores;
```

```

String final_score =
Integer.toString(whole_score);

out of 10");
scoretxt.setText(final_score + "
//to get the percentage
double accuracy =
((score/10)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=8) {
commenttxt.setText("Excellent!");

} else if(whole_score<=7 &&
whole_score>=6){

commenttxt.setText("Keep up the good work!");
}
else if(whole_score<=5 &&
whole_score>=3){

commenttxt.setText("You have to study more..");
}
else {
commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));
accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
Color(51, 51, 51));
accuracytxt.setBounds(62, 178,
312, 69);
String accuracytext =
df.format(accuracy);
accuracytxt.setText(accuracytext
+ "%");
contentPane.add(accuracytxt);

JButton btnbckmenu = new
JButton("");
btnbckmenu.setBorder(null);

```

```
btnbckmenu.addMouseListener(new MouseAdapter() {  
    @Override  
    public void  
mouseEntered(MouseEvent e) {  
  
    btnbckmenu.setIcon(new  
ImageIcon(Result_OS.class.getResource("/images/bcktomenuho  
ver.png")));  
    }  
    @Override  
    public void  
mouseExited(MouseEvent e) {  
  
    btnbckmenu.setIcon(new  
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p  
ng")));  
    }  
    @Override  
    public void  
mouseClicked(MouseEvent e) {  
        Home fhome =  
new Home();  
        setVisible(false);  
  
        fhome.setVisible(true);  
  
        Home.clip setFramePosition(0);  
  
        Home.clip.start();  
    }  
});  
btnbckmenu.setIcon(new  
ImageIcon(Result_OS.class.getResource("/images/bcktomenu.p  
ng")));  
btnbckmenu.setBounds(94, 483,  
256, 60);  
contentPane.add(btnbckmenu);  
  
JButton btnexit = new  
JButton("");  
btnexit.addMouseListener(new  
MouseAdapter() {  
    @Override  
    public void  
mouseEntered(MouseEvent e) {  
  
    btnexit.setIcon(new  
ImageIcon(Result_OS.class.getResource("/images/exthover.png"  
)));  
    }  
    @Override  
    public void  
mouseExited(MouseEvent e) {  
  
    btnexit.setIcon(new  
ImageIcon(Result_OS.class.getResource("/images/exthover.png")));  
    }  
    @Override  
    public void
```

```

        btnclose.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnclose.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
btnclose.setBorder(null);
btnclose.setBounds(94, 591, 256,
60);
contentPane.add(btnclose);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}

class Result_ch6OS extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch6OS() {
        // Set the icon image of the
frame
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();

```

```
txtYouGot.setBorder(null);
txtYouGot.setEditable(false);
txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
txtYouGot.setForeground(new
Color(255, 51, 0));

txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
txtYouGot.setText("YOU GOT");
txtYouGot.setBackground(new
Color(51, 51, 51));
txtYouGot.setBounds(62, 111,
312, 69);
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
txtScore.setForeground(new
Color(255, 51, 0));
txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
txtScore.setBackground(new
Color(51, 51, 51));
txtScore.setBounds(62, 245,
141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);
scoretxt.setForeground(new
Color(255, 51, 0));
scoretxt.setFont(new
Font("Arial", Font.BOLD, 30));
scoretxt.setBackground(new
Color(51, 51, 51));
scoretxt.setBounds(198, 245,
176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);
```

```

        commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
        commenttxt.setForeground(new Color(255, 51, 0));
        commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
        commenttxt.setEditable(false);
        commenttxt.setColumns(10);
        commenttxt.setBackground(new Color(51, 51, 51));
        commenttxt.setBounds(62, 295, 312, 69);
        contentPane.add(commenttxt);

        //call the score for java
        double score =
        int whole_score = (int) score ;
        String final_score =
        Integer.toString(whole_score);
        scoretxt.setText(final_score + " out of 10");
        //to get the percentage
        double accuracy =
        ((score/10)*100);
        DecimalFormat df = new DecimalFormat("#.#");
        //for rating
        if (whole_score>=8) {
            commenttxt.setText("Excellent!");
        } else if(whole_score<=7 && whole_score>=6){
            commenttxt.setText("Keep up the good work!");
        } else if(whole_score<=5 && whole_score>=3){
            commenttxt.setText("You have to study more..");
        } else {
            commenttxt.setFont(new Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);
    }
}

```

```
accuracytxt.setHorizontalAlignment(SwingConstants.CENTER);
accuracytxt.setForeground(new Color(255, 51, 0));
accuracytxt.setFont(new Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new Color(51, 51, 51));
accuracytxt.setBounds(62, 178, 312, 69);
String accuracytext =
df.format(accuracy);
accuracytext += "%");
contentPane.add(accuracytxt);

JButton btbckmenu = new JButton("");
btbckmenu.setBorder(null);

btbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenuhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenu.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        new Home();
        Home fhome =
        fhome.setVisible(false);
        fhome.setVisible(true);
        Home.clip.setFramePosition(0);
        Home.clip.start();
    }
});
btbckmenu.setIcon(new ImageIcon(Result_OS.class.getResource("/images/bcktomenu.png")));

```

```

256, 60);
256, 60);

        btnbckmenu.setBounds(94, 483,
        contentPane.add(btnbckmenu);

        JButton btnexit = new
        JButton("");
        btnexit.addMouseListener(new
        MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {

                btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/exthover.png"
)));
            }
            @Override
            public void
mouseExited(MouseEvent e) {

                btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/ext.png")));
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256,
60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_OS.class.getResource("/images/bg.png")));
        bglabel.setBounds(0, 0, 450,
730);
        contentPane.add(bglabel);
    }
}
}

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;

public class Result_MICRO extends JFrame {

    /**
     * Launch the application.
     */

```

Result for Microprocessor quiz

```

public static void main(String[] args) {
    Result_MICRO frame = new Result_MICRO();
    frame.setVisible(true);

}

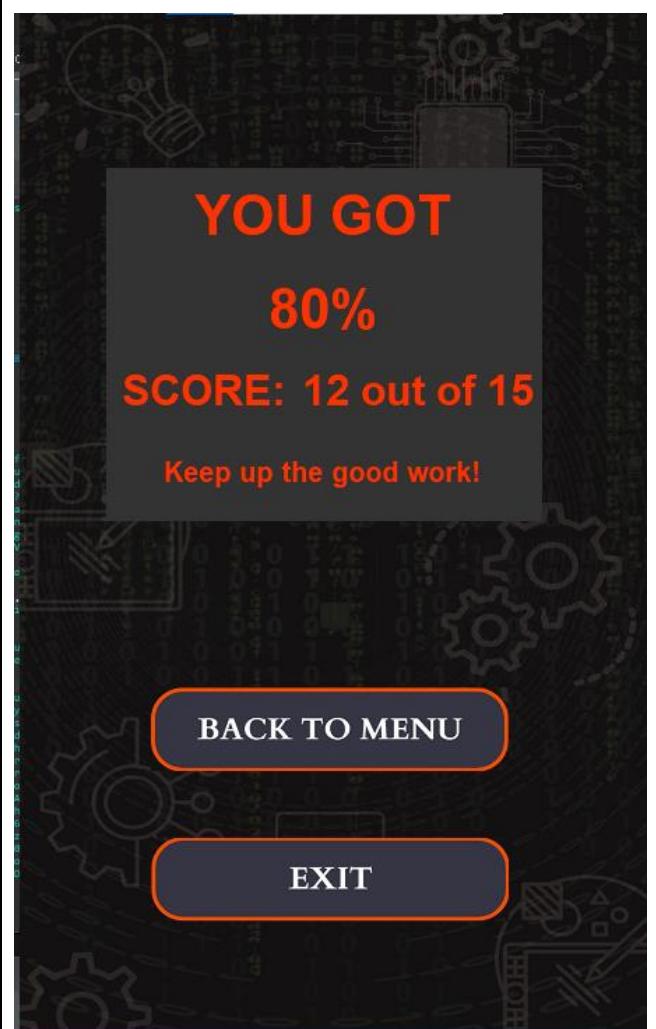
public Result_MICRO() {
    // Set the icon image of the frame

setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
    setUndecorated(true);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 730);
    setLocationRelativeTo(null);
    setResizable(false);
    JPanel contentPane;
    contentPane = new JPanel();
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JButton btnbckmenu = new JButton("");
    btnbckmenu.setBorder(null);
    btnbckmenu.addMouseListener(new
MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent
e) {
            btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
        }
        @Override
        public void mouseExited(MouseEvent e)
{
            btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
        }
        @Override
        public void mouseClicked(MouseEvent
e) {
            Home fhome = new Home();
            setVisible(false);
            fhome.setVisible(true);
            Home.clip setFramePosition(0);
            Home.clip.start();
        }
    });
    btnbckmenu.setIIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
    btnbckmenu.setBounds(94, 483, 256, 60);
    contentPane.add(btnbckmenu);
}

```



```

JButton btnexit = new JButton("");
btnexit.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent
e) {
        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
    }
    @Override
    public void mouseExited(MouseEvent e)
{
        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
    }
    @Override
    public void mouseClicked(MouseEvent
e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256, 60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bg.png"))
);
bglabel.setBounds(0, 0, 450, 730);
contentPane.add(bglabel);
}

class Result_ch1MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch1MICRO() {
        // Set the icon image of the frame

```

```
setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.getResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new Font("Arial",
Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new Color(51,
51, 51));
        txtYouGot.setBounds(62, 111, 312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");
        txtScore.setBorder(null);
        txtScore.setEditable(false);

        txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
        txtScore.setForeground(new Color(255,
51, 0));
        txtScore.setFont(new Font("Arial",
Font.BOLD, 30));
        txtScore.setColumns(10);
        txtScore.setBackground(new Color(51,
51, 51));
        txtScore.setBounds(62, 245, 141, 53);
        contentPane.add(txtScore);

        scoretxt = new JTextField();
        scoretxt.setBorder(null);
        scoretxt.setEditable(false);

        scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);
        scoretxt.setForeground(new Color(255,
51, 0));
```

```

scoretxt.setFont(new Font("Arial",
Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51,
51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new
Color(255, 51, 0));
commenttxt.setFont(new Font("Arial",
Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new
Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312,
69);
contentPane.add(commenttxt);

//call the score for java
double score =
MICROquiz.ch1_MICRO.scores;
int whole_score = (int) score ;

String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + " out of
15");
//to get the percentage
double accuracy = ((score/15)*100);
DecimalFormat df = new
DecimalFormat("#.#");
//for rating
if (whole_score>=13) {

commenttxt.setText("Excellent!");

} else if(whole_score<=12 &&
whole_score>=9){
commenttxt.setText("Keep up
the good work!");
}
else if(whole_score<=8 &&
whole_score>=6){
commenttxt.setText("You have
to study more..");
}
else {
}

```

```

commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));
accuracytxt.setFont(new Font("Arial",
Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
Color(51, 51, 51));
accuracytxt.setBounds(62, 178, 312, 69);
String accuracytext =
df.format(accuracy);
accuracytxt.setText(accuracytext + "%");
contentPane.add(accuracytxt);

JButton btnbckmenu = new JButton("");
btnbckmenu.setBorder(null);
btnbckmenu.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
    }
    @Override
    public void
mouseExited(MouseEvent e) {

        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        Home fhome = new
Home();
        setVisible(false);
        fhome.setVisible(true);

        Home.clip setFramePosition(0);
        Home.clip.start();
    }
});
```

```

        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
        btnbckmenu.setBounds(94, 483, 256,
60);
        contentPane.add(btnbckmenu);

        JButton btnexit = new JButton("");
        btnexit.addMouseListener(new
MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {
                btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {
                btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256, 60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bg.png"))
);
        bglabel.setBounds(0, 0, 450, 730);
        contentPane.add(bglabel);
    }
}

class Result_ch2MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**

```

```

        * Create the frame.
        */
        public Result_ch2MICRO() {
            // Set the icon image of the
frame
            setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
            setUndecorated(true);

            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(100, 100, 450, 730);
            setLocationRelativeTo(null);
            setResizable(false);
            contentPane = new JPanel();
            setContentPane(contentPane);
            contentPane.setLayout(null);
            setVisible(true);

            txtYouGot = new JTextField();
            txtYouGot.setBorder(null);
            txtYouGot.setEditable(false);
            txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
            txtYouGot.setForeground(new
Color(255, 51, 0));

            txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
            txtYouGot.setText("YOU GOT");
            txtYouGot.setBackground(new
Color(51, 51, 51));
            txtYouGot.setBounds(62, 111,
312, 69);
            contentPane.add(txtYouGot);
            txtYouGot.setColumns(10);

            txtScore = new JTextField();
            txtScore.setText("SCORE:");
            txtScore.setBorder(null);
            txtScore.setEditable(false);

            txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
            txtScore.setForeground(new
Color(255, 51, 0));
            txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
            txtScore.setColumns(10);
            txtScore.setBackground(new
Color(51, 51, 51));
            txtScore.setBounds(62, 245,
141, 53);
            contentPane.add(txtScore);

            scoretxt = new JTextField();

```

```

        scoretxt.setBorder(null);
        scoretxt.setEditable(false);

    scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
    scoretxt.setForeground(new Color(255, 51, 0));
    scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
    scoretxt.setColumns(10);
    scoretxt.setBackground(new Color(51, 51, 51));
    scoretxt.setBounds(198, 245, 176, 53);
    contentPane.add(scoretxt);

    commenttxt = new JTextField();
    commenttxt.setBorder(null);

    commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
    commenttxt.setForeground(new Color(255, 51, 0));
    commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
    commenttxt.setEditable(false);
    commenttxt.setColumns(10);
    commenttxt.setBackground(new Color(51, 51, 51));
    commenttxt.setBounds(62, 295, 312, 69);
    contentPane.add(commenttxt);

    //call the score for java
    double score =
MICROquiz.ch2_MICRO.scores;
    int whole_score = (int) score ;

    String final_score =
Integer.toString(whole_score);

    out of 10");
    //to get the percentage
    double accuracy =
((score/10)*100);

    DecimalFormat df = new
DecimalFormat("#.#");
    //for rating
    if (whole_score>=8) {

        commenttxt.setText("Excellent!");

        } else if(whole_score<=7 &&
whole_score>=6){


```

```

        commenttxt.setText("Keep up the good work!");
    }
    else if(whole_score<=5 &&
whole_score>=3){

        commenttxt.setText("You have to study more..");
    }
    else {
        commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
        commenttxt.setText("It's
okay...Study more,\nyou can do it!");
    }
    accuracytxt = new JTextField();
    accuracytxt.setBorder(null);
    accuracytxt.setEditable(false);

    accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
    accuracytxt.setForeground(new
Color(255, 51, 0));
    accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
    accuracytxt.setColumns(10);
    accuracytxt.setBackground(new
Color(51, 51, 51));
    accuracytxt.setBounds(62, 178,
312, 69);
    String accuracytext =
df.format(accuracy);
    accuracytxt.setText(accuracytext
+ "%");
    contentPane.add(accuracytxt);

    JButton btnbckmenu = new
JButton("");
    btnbckmenu.setBorder(null);

    btnbckmenu.addMouseListener(new MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

            btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
        }
        @Override
        public void
mouseExited(MouseEvent e) {

            btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
        }
    })

```

```

        @Override
        public void
mouseClicked(MouseEvent e) {
                    Home fhome =
new Home();
                        setVisible(false);

fhome.setVisible(true);

Home.clip setFramePosition(0);

Home.clip.start();
}
});
btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bctomen
u.png")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btnbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
                    @Override
                    public void
mouseEntered(MouseEvent e) {

btnclick.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
}
                    @Override
                    public void
mouseExited(MouseEvent e) {

btnclick.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png")));
}
                    @Override
                    public void
mouseClicked(MouseEvent e) {
                        System.exit(0);
}
});
btnclick.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png")));
btnclick.setBorder(null);
btnclick.setBounds(94, 591, 256,
60);
contentPane.add(btnclick);

JLabel bglabel = new JLabel();

```

```

        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bg.png"))
);
        bglabel.setBounds(0, 0, 450,
730);
        contentPane.add(bglabel);
    }
}

class Result_ch3MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch3MICRO() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);
    }
}

```

```

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENTER);
txtScore.setForeground(new Color(255, 51, 0));
txtScore.setFont(new Font("Arial", Font.BOLD, 30));
txtScore.setColumns(10);
txtScore.setBackground(new Color(51, 51, 51));
txtScore.setBounds(62, 245, 141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
MICROquiz.ch3_MICRO.scores;

```

```

        int whole_score = (int) score ;

        String final_score =
Integer.toString(whole_score);

        out of 10");
        scoretxt.setText(final_score + " 

//to get the percentage
        double accuracy =
((score/10)*100);

        DecimalFormat df = new
DecimalFormat("#.#");
        //for rating
        if (whole_score>=8) {

            commenttxt.setText("Excellent!");

        } else if(whole_score<=7 &&
whole_score>=6){

            commenttxt.setText("Keep up the good work!");
        }
        else if(whole_score<=5 &&
whole_score>=3){

            commenttxt.setText("You have to study more..");
        }
        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
        }
        accuracytxt = new JTextField();
        accuracytxt.setBorder(null);
        accuracytxt.setEditable(false);

        accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
        accuracytxt.setForeground(new
Color(255, 51, 0));
        accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
        accuracytxt.setColumns(10);
        accuracytxt.setBackground(new
Color(51, 51, 51));
        accuracytxt.setBounds(62, 178,
312, 69);
        String accuracytext =
df.format(accuracy);
        accuracytxt.setText(accuracytext
+ "%");
        contentPane.add(accuracytxt);

        JButton btnbckmenu = new
JButton("");

```

```

        btnbckmenu.setBorder(null);

    btnbckmenu.addMouseListener(new MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
    }
        @Override
        public void
mouseExited(MouseEvent e) {

        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
    }
        @Override
        public void
mouseClicked(MouseEvent e) {
            Home fhome =
new Home();
            setVisible(false);

            fhome.setVisible(true);

            Home.clip setFramePosition(0);

            Home.clip.start();
        }
    });
    btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
    btnbckmenu.setBounds(94, 483,
256, 60);
    contentPane.add(btnbckmenu);

    JButton btnexit = new
JButton("");
    btnexit.addMouseListener(new
MouseAdapter() {
        @Override
        public void
mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
    }
        @Override
        public void
mouseExited(MouseEvent e) {

```

```

        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
    }
    @Override
    public void
mouseClicked(MouseEvent e) {
        System.exit(0);
    }
});
btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
);
btnexit.setBorder(null);
btnexit.setBounds(94, 591, 256,
60);
contentPane.add(btnexit);

JLabel bglabel = new JLabel();
bglabel.setOpaque(true);
bglabel.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bg.png"))
);
bglabel.setBounds(0, 0, 450,
730);
contentPane.add(bglabel);
}
}
class Result_ch4MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch4MICRO() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
}

```

```
setVisible(true);

txtYouGot = new JTextField();
txtYouGot.setBorder(null);
txtYouGot.setEditable(false);
txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
txtYouGot.setForeground(new
Color(255, 51, 0));

txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
txtYouGot.setText("YOU GOT");
txtYouGot.setBackground(new
Color(51, 51, 51));
txtYouGot.setBounds(62, 111,
312, 69);
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
txtScore.setForeground(new
Color(255, 51, 0));
txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
txtScore.setBackground(new
Color(51, 51, 51));
txtScore.setBounds(62, 245,
141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTE
R);
scoretxt.setForeground(new
Color(255, 51, 0));
scoretxt.setFont(new
Font("Arial", Font.BOLD, 30));
scoretxt.setBackground(new
Color(51, 51, 51));
scoretxt.setBounds(198, 245,
176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
```

```
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
MICROquiz.ch4_MICRO.scores;
int whole_score = (int) score;

String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + " out of 10");

//to get the percentage
double accuracy =
((score/10)*100);
DecimalFormat df = new DecimalFormat("#.#");
//for rating
if (whole_score>=8) {
commenttxt.setText("Excellent!");
} else if(whole_score<=7 && whole_score>=6){
commenttxt.setText("Keep up the good work!");
}
else if(whole_score<=5 && whole_score>=3){
commenttxt.setText("You have to study more.. ");
}
else {
commenttxt.setFont(new Font("Arial", Font.BOLD, 17));
commenttxt.setText("It's okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);
```

```
accuracytxt.setHorizontalAlignment(SwingConstants.CENTER);
accuracytxt.setForeground(new Color(255, 51, 0));
accuracytxt.setFont(new Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new Color(51, 51, 51));
accuracytxt.setBounds(62, 178, 312, 69);
String accuracytext =
df.format(accuracy);
accuracytext += "%");
contentPane.add(accuracytxt);

JButton btbckmenu = new JButton("");
btbckmenu.setBorder(null);

btbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_MICRO.class.getResource("/images/bcktomenuhover.png")));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        btbckmenu.setIcon(new ImageIcon(Result_MICRO.class.getResource("/images/bcktomenu.png")));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        Home fhome =
new Home();
        fhome.setVisible(false);
        fhome.setVisible(true);
        Home.clip.setFramePosition(0);
        Home.clip.start();
    }
});
btbckmenu.setIcon(new ImageIcon(Result_MICRO.class.getResource("/images/bcktomenu.png")));
```

```

256, 60);
256, 60);

        btnbckmenu.setBounds(94, 483,
        contentPane.add(btnbckmenu);

        JButton btnexit = new
        JButton("");
        btnexit.addMouseListener(new
        MouseAdapter() {
            @Override
            public void
mouseEntered(MouseEvent e) {

                btnexit.setIcon(new
                ImageIcon(Result_MICRO.class.getResource("/images/exthover.
                png")));
            }
            @Override
            public void
mouseExited(MouseEvent e) {

                btnexit.setIcon(new
                ImageIcon(Result_MICRO.class.getResource("/images/ext.png")))
            );
            }
            @Override
            public void
mouseClicked(MouseEvent e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
        ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))
    );
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256,
60);
        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
        ImageIcon(Result_MICRO.class.getResource(" /images/bg.png"))
    );
        bglabel.setBounds(0, 0, 450,
730);
        contentPane.add(bglabel);
    }

}

class Result_ch5MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;
}

```

```

    /**
     * Create the frame.
     */
    public Result_ch5MICRO() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));

        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
        contentPane.add(txtYouGot);
        txtYouGot.setColumns(10);

        txtScore = new JTextField();
        txtScore.setText("SCORE:");
        txtScore.setBorder(null);
        txtScore.setEditable(false);

        txtScore.setHorizontalAlignment(SwingConstants.CENT
ER);
        txtScore.setForeground(new
Color(255, 51, 0));
        txtScore.setFont(new
Font("Arial", Font.BOLD, 30));
        txtScore.setBackground(new
Color(51, 51, 51));
        txtScore.setBounds(62, 245,
141, 53);
        contentPane.add(txtScore);
    }
}

```

```

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);

//call the score for java
double score =
MICROquiz.ch5_MICRO.scores;
int whole_score = (int) score ;

String final_score =
Integer.toString(whole_score);

scoretxt.setText(final_score + " out of 15");

//to get the percentage
double accuracy =
((score/15)*100);

DecimalFormat df = new DecimalFormat("#.#");

//for rating
if (whole_score>=13) {

commenttxt.setText("Excellent!");
}

```

```

        } else if(whole_score<=12 &&
whole_score>=9){

    commenttxt.setText("Keep up the good work!");
    }
    else if(whole_score<=8 &&
whole_score>=6){

        commenttxt.setText("You have to study more..");
        }
        else {
            commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
            commenttxt.setText("It's
okay...Study more,\nyou can do it!");
            }
            accuracytxt = new JTextField();
            accuracytxt.setBorder(null);
            accuracytxt.setEditable(false);

            accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
            accuracytxt.setForeground(new
Color(255, 51, 0));
            accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
            accuracytxt.setColumns(10);
            accuracytxt.setBackground(new
Color(51, 51, 51));
            accuracytxt.setBounds(62, 178,
312, 69);
            String accuracytext =
df.format(accuracy);
            accuracytxt.setText(accuracytext
+ "%");
            contentPane.add(accuracytxt);

            JButton btnbckmenu = new
JButton("");
            btnbckmenu.setBorder(null);

            btnbckmenu.addMouseListener(new MouseAdapter() {
                @Override
                public void
mouseEntered(MouseEvent e) {

                    btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
                }
                @Override
                public void
mouseExited(MouseEvent e) {

                    btnbckmenu.setIcon(new

```

```
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
        }
    @Override
    public void
mouseClicked(MouseEvent e) {
                Home fhome =
new Home();
                setVisible(false);

        fhome.setVisible(true);

        Home.clip setFramePosition(0);

        Home.clip.start();
            }
        });
        btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
        btnbckmenu.setBounds(94, 483,
256, 60);
        contentPane.add(btnbckmenu);

        JButton btnexit = new
JButton("");
        btnexit.addMouseListener(new
MouseAdapter() {
                @Override
                public void
mouseEntered(MouseEvent e) {

                    btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
                }
            @Override
            public void
mouseExited(MouseEvent e) {

                btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png")));
            }
        });
        btnexit.setI
    @Override
    public void
mouseClicked(MouseEvent e) {
                System.exit(0);
            }
        });
        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/ext.png")));
        btnexit.setBorder(null);
        btnexit.setBounds(94, 591, 256,
60);
```

```

        contentPane.add(btnexit);

        JLabel bglabel = new JLabel();
        bglabel.setOpaque(true);
        bglabel.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bg.png")))
);
        bglab
    730);
        contentPane.add(bglabel);
    }
}

class Result_ch6MICRO extends JFrame {

    private JPanel contentPane;
    private JTextField txtYouGot;
    private JTextField accuracytxt;
    private JTextField txtScore;
    private JTextField scoretxt;
    private JTextField commenttxt;

    /**
     * Create the frame.
     */
    public Result_ch6MICRO() {
        // Set the icon image of the
frame
        setIconImage(Toolkit.getDefaultToolkit().getImage(Home.class.g
etResource("/images/logo.png")));
        setUndecorated(true);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 730);
        setLocationRelativeTo(null);
        setResizable(false);
        contentPane = new JPanel();
        setContentPane(contentPane);
        contentPane.setLayout(null);
        setVisible(true);

        txtYouGot = new JTextField();
        txtYouGot.setBorder(null);
        txtYouGot.setEditable(false);
        txtYouGot.setFont(new
Font("Arial", Font.BOLD, 40));
        txtYouGot.setForeground(new
Color(255, 51, 0));
        txtYouGot.setHorizontalAlignment(SwingConstants.CEN
TER);
        txtYouGot.setText("YOU GOT");
        txtYouGot.setBackground(new
Color(51, 51, 51));
        txtYouGot.setBounds(62, 111,
312, 69);
    }
}

```

```
contentPane.add(txtYouGot);
txtYouGot.setColumns(10);

txtScore = new JTextField();
txtScore.setText("SCORE:");
txtScore.setBorder(null);
txtScore.setEditable(false);

txtScore.setHorizontalAlignment(SwingConstants.CENTER);
txtScore.setForeground(new Color(255, 51, 0));
txtScore.setFont(new Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
txtScore.setColumns(10);
txtScore.setBackground(new Color(51, 51, 51));
txtScore.setBounds(62, 245, 141, 53);
contentPane.add(txtScore);

scoretxt = new JTextField();
scoretxt.setBorder(null);
scoretxt.setEditable(false);

scoretxt.setHorizontalAlignment(SwingConstants.CENTER);
scoretxt.setForeground(new Color(255, 51, 0));
scoretxt.setFont(new Font("Arial", Font.BOLD, 30));

Color(51, 51, 51));
scoretxt.setColumns(10);
scoretxt.setBackground(new Color(51, 51, 51));
scoretxt.setBounds(198, 245, 176, 53);
contentPane.add(scoretxt);

commenttxt = new JTextField();
commenttxt.setBorder(null);

commenttxt.setHorizontalAlignment(SwingConstants.CENTER);
commenttxt.setForeground(new Color(255, 51, 0));
commenttxt.setFont(new Font("Arial", Font.BOLD, 20));

Font("Arial", Font.BOLD, 20));
commenttxt.setEditable(false);
commenttxt.setColumns(10);
commenttxt.setBackground(new Color(51, 51, 51));
commenttxt.setBounds(62, 295, 312, 69);
contentPane.add(commenttxt);
```

```

        //call the score for java
        double score =
MICROquiz.ch6_MICRO.scores;
                int whole_score = (int) score ;

                        String final_score =
Integer.toString(whole_score);

                                scoretxt.setText(final_score + "
out of 10");
                                //to get the percentage
                                double accuracy =
((score/10)*100);
                                DecimalFormat df = new
DecimalFormat("#.##");
                                //for rating
                                if (whole_score>=8) {

commenttxt.setText("Excellent!");

                                } else if(whole_score<=7 &&
whole_score>=6){

commenttxt.setText("Keep up the good work!");
}
                                else if(whole_score<=5 &&
whole_score>=3){

commenttxt.setText("You have to study more..");
}
                                else {
commenttxt.setFont(new
Font("Arial", Font.BOLD, 17));
commenttxt.setText("It's
okay...Study more,\nyou can do it!");
}
accuracytxt = new JTextField();
accuracytxt.setBorder(null);
accuracytxt.setEditable(false);

accuracytxt.setHorizontalAlignment(SwingConstants.CE
NTER);
accuracytxt.setForeground(new
Color(255, 51, 0));
accuracytxt.setFont(new
Font("Arial", Font.BOLD, 40));
accuracytxt.setColumns(10);
accuracytxt.setBackground(new
Color(51, 51, 51));
accuracytxt.setBounds(62, 178,
312, 69);
String accuracytext =
df.format(accuracy);
accuracytext.setText(accuracytext +
"%");
contentPane.add(accuracytxt);

```

```

JButton btnbckmenu = new
JButton("");
btnbckmenu.setBorder(null);

btnbckmenu.addMouseListener(new MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

    btbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
uhover.png")));
}

@Override
public void
mouseExited(MouseEvent e) {

    btbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
}

@Override
public void
mouseClicked(MouseEvent e) {
    Home fhome =
new Home();
    setVisible(false);

    fhome.setVisible(true);

    Home.clip setFramePosition(0);

    Home.clip.start();
}
});

btnbckmenu.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/bcktomen
u.png")));
btnbckmenu.setBounds(94, 483,
256, 60);
contentPane.add(btbckmenu);

JButton btnexit = new
JButton("");
btnexit.addMouseListener(new
MouseAdapter() {
    @Override
    public void
mouseEntered(MouseEvent e) {

        btnexit.setIcon(new
ImageIcon(Result_MICRO.class.getResource("/images/exthover.
png")));
}
});

```

```
        public void  
mouseExited(MouseEvent e) {  
  
    btnexit.setIcon(new  
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))  
);  
    }  
    @Override  
    public void  
mouseClicked(MouseEvent e) {  
    System.exit(0);  
    }  
});  
btnexit.setIcon(new  
ImageIcon(Result_MICRO.class.getResource("/images/ext.png"))  
);  
btnexit.setBorder(null);  
btnexit.setBounds(94, 591, 256,  
60);  
contentPane.add(btnexit);  
  
JLabel bglabel = new JLabel();  
bglabel.setOpaque(true);  
bglabel.setIcon(new  
ImageIcon(Result_MICRO.class.getResource("/images/bg.png"))  
);  
bglabel.setBounds(0, 0, 450,  
730);  
contentPane.add(bglabel);  
}  
}  
}
```