

Appendix

Code

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 // importing required Java.swing tools for button functionality
7
8 public class StudyTimer extends JFrame {
9
10     //instance variables moved
11     private int studyTime;
12     private int breakTime;
13     private int totalStudyTime;
14     private int totalBreakTime;
15     private int remainingTime;
16     private boolean isStudying;
17     private boolean isBreak;
18     private Timer timer;
19     private JLabel timerLabel;
20     private JButton startButton;
21     private JButton pauseButton;
22     private JLabel studyTimeLabel;
23     private JLabel breakTimeLabel;
24     private JLabel totalStudyLabel;
25     private JLabel totalBreakLabel;
26     private JPanel controlPanel;
27
28
29 public StudyTimer(int studyTime, int breakTime) {
30
31     this.studyTime = studyTime * 60; // convert studyTime from mins. to secs.
32     this.breakTime = breakTime * 60; //convert from mins. to secs.
33
34     this.totalStudyTime = 0; //total time should start at 0
35     this.totalBreakTime = 0;
36
37     this.remainingTime = this.studyTime;
38     // setting remaining time relative to study session length
39
40     this.isStudying = false;
41     this.isBreak = false;
42
43     setTitle("Study Timer");
44     setSize(500, 400);
45     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46     setLayout(new GridLayout(5, 2));
47
48     controlPanel = new JPanel();
49
50     // names for each label
51     studyTimeLabel = new JLabel(" Study Time: " + studyTime + " minute(s)");
52     add(studyTimeLabel);
53
54     breakTimeLabel = new JLabel(" Break Time: " + breakTime + " minute(s)");
55     add(breakTimeLabel);
```

```

56 totalStudyLabel = new JLabel("    Total Study Time: " + totalStudyTime + " seconds");
57 add(totalStudyLabel);
58
59 totalBreakLabel = new JLabel("    Total Break Time: " + totalBreakTime + " seconds");
60
61 add(totalBreakLabel);
62
63
64 // variables for remaining time label
65 timerLabel = new JLabel("Remaining Time: " + studyTime + ":00");
66 timerLabel.setFont(new Font("Arial", Font.BOLD, 20));
67 timerLabel.setHorizontalAlignment(SwingConstants.CENTER);
68 add(timerLabel);
69
70 //start button
71 startButton = new JButton("Start");
72 startButton.addActionListener(new ActionListener() {
73     @Override
74     public void actionPerformed(ActionEvent e) {
75         startSession(); //start timer when button is pressed
76     }
77 });
78 controlPanel.add(startButton); //add start button to the panel
79
80 pauseButton = new JButton("Pause"); //pause button
81 pauseButton.addActionListener(new ActionListener() {
82     @Override
83     public void actionPerformed(ActionEvent e) {
84         pauseSession(); //stop when button is pressed
85     }
86 });
87 controlPanel.add(pauseButton); //add pause to control panel
88
89 add(controlPanel, BorderLayout.SOUTH);
90
91
92 public void startSession() {
93     isStudying = true;
94     startButton.setEnabled(false);
95     pauseButton.setEnabled(true);
96     timer = new Timer(1000, new ActionListener() {
97         @Override
98         public void actionPerformed(ActionEvent e) {
99             remainingTime--;
100             updateTimerLabel();
101             if (remainingTime <= 0) {
102                 if (isStudying) {
103                     totalStudyTime += studyTime;
104                     isStudying = false;
105                     isBreak = true;
106                     remainingTime = breakTime;
107                     updateTimerLabel(); //to make sure timer constantly moves
108                 } else {
109                     totalBreakTime += breakTime;

```

```

        totalBreakTime += breakTime;
        isBreak = false;
        isStudying = true;
        remainingTime = studyTime;
        updateTimerLabel();
    }
}
});
timer.start();
}

• public void pauseSession() { //simple pause method using booleans
    timer.stop();
    startButton.setEnabled(true);
    pauseButton.setEnabled(false);
}

• private void updateTimerLabel() {
    int minutes = remainingTime / 60;
    int seconds = remainingTime % 60;
    timerLabel.setText("Remaining Time: " + String.format("%02d:%02d", minutes, seconds));
}

• public static void main(String[] args) { //OptionPane window for entering study times
    int studyTime = Integer.parseInt(JOptionPane.showInputDialog("Enter study time in minutes: (Recommended 25 mins)"));
    int breakTime = Integer.parseInt(JOptionPane.showInputDialog("Enter break time in minutes: (recommended 5 mins)"));

    StudyTimer timer = new StudyTimer(studyTime, breakTime);
    timer.setVisible(true);
}

```