

Create a Swing application to demonstrate use of TextArea using scrollpane to show content of text file in text area selected using file chooser:

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

public class SimpleTextViewer {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Text Viewer");
        JTextArea textArea = new JTextArea();
        JButton openButton = new JButton("Open");

        openButton.addActionListener(e -> {
            JFileChooser chooser = new JFileChooser();
            if (chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
                try {
                    textArea.read(new BufferedReader(new FileReader(chooser.getSelectedFile())));
                } catch (Exception ignored) {}
            }
        });

        frame.add(new JScrollPane(textArea));
        frame.add(openButton, BorderLayout.SOUTH);
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Write a program to create a class and implement a default, overloaded and copy

Constructor:

```
class Arithmetic {
    int a, b;

    // Default Constructor (sets default values)
    Arithmetic() {
        a = 0;
        b = 0;
    }

    // Overloaded Constructor (takes two numbers)
    Arithmetic(int x, int y) {
        a = x;
        b = y;
    }

    // Copy Constructor (copies values from another object)
    Arithmetic(Arithmetic obj) {
        a = obj.a;
        b = obj.b;
    }

    // Method to perform addition
    void add() {
        System.out.println("Sum: " + (a + b));
    }

    public static void main(String[] args) {
        Arithmetic obj1 = new Arithmetic(); // Default constructor
        Arithmetic obj2 = new Arithmetic(5, 10); // Overloaded constructor
        Arithmetic obj3 = new Arithmetic(obj2); // Copy constructor

        obj1.add(); // Output: Sum: 0
        obj2.add(); // Output: Sum: 15
        obj3.add(); // Output: Sum: 15
    }
}
```

Write a program to implement the concepts of Abstract classes and methods:

```
abstract class Operation {
    abstract void calculate(int a, int b); // Abstract method
}

class Addition extends Operation {
    void calculate(int a, int b) {
        System.out.println("Sum: " + (a + b));
    }
}

public class AbstractDemo {
    public static void main(String[] args) {
        Operation obj = new Addition();
        obj.calculate(5, 3); // Output: Sum: 8
    }
}
```

Write programs for Border Layout:

```
import javax.swing.*.*;
import java.awt.*.*;

public class BorderLayoutExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("BorderLayout Example");
        frame.setLayout(new BorderLayout());

        frame.add(new JButton("North"), BorderLayout.NORTH);
        frame.add(new JButton("South"), BorderLayout.SOUTH);
        frame.add(new JButton("East"), BorderLayout.EAST);
        frame.add(new JButton("West"), BorderLayout.WEST);
        frame.add(new JButton("Center"), BorderLayout.CENTER);

        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Write programs to demonstrate the action event:

```
import java.awt.*;
import java.awt.event.*;
class ActEvent extends Frame implements ActionListener
{
    TextField tf;
    ActEvent()
    {
        tf= new TextField();
        tf.setBounds(60,50,170,20);
        Button b=new Button("click me");
        b.setBounds(100,120,80,30);
        b.addActionListener(this);
        add(b); add(tf);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        tf.setText("Welcome");
    }
    public static void main(String args[])
    {
        new ActEvent();
    }
}
```