

Radio Button

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RadioButton {

    public static void main(String[] args) {

        // Create a JFrame (window)

        JFrame frame = new JFrame("Radio
        Button Demo");

        frame.setDefaultCloseOperation(JFrame.EXIT_
        ON_CLOSE);

        frame.setSize(500, 500);

        // Create a panel to hold the radio
        buttons

        JPanel panel = new JPanel();

        // Create a label to display the selected
        option

        JLabel label = new JLabel("Selected
        Option: ");

        label.setHorizontalAlignment(JLabel.CENTER);

        // Create radio buttons

        JRadioButton option1 = new
        JRadioButton("Option 1");

        JRadioButton option2 = new
        JRadioButton("Option 2");

        JRadioButton option3 = new
        JRadioButton("Option 3");
```

```
        // Create a button group to ensure
        exclusive selection

        ButtonGroup group = new ButtonGroup();

        group.add(option1);

        group.add(option2);

        group.add(option3);

        // Add action listeners to radio buttons

        option1.addActionListener(new
        ActionListener() {

            @Override

            public void
            actionPerformed(ActionEvent e) {

                label.setText("Selected Option:
                Option 1");

            }

        });

        option2.addActionListener(new
        ActionListener() {

            @Override

            public void
            actionPerformed(ActionEvent e) {

                label.setText("Selected Option:
                Option 2");

            }

        });

        option3.addActionListener(new
        ActionListener() {

            @Override

            public void
            actionPerformed(ActionEvent e) {
```

```

        label.setText("Selected Option:
Option 3");
    }
});

// Add components to the panel
panel.add(option1);
panel.add(option2);
panel.add(option3);

// Create a layout for the panel
panel.setLayout(new GridLayout(3, 1));

// Add the panel and label to the frame
frame.add(panel, BorderLayout.WEST);
frame.add(label, BorderLayout.CENTER);

frame.setVisible(true);
}
}

```

Student.java

```

class Student {
    private String name;
    private int age;

    // Parameterized constructor
    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }
}

```

```

    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }

    public static void main(String[] args) {
        // Creating a Student object using the
        parameterized constructor

        Student student1 = new Student("John",
20);

        // Creating another Student object using
        the parameterized constructor

        Student student2 = new Student("Alice",
22);

        // Display details of the students
        System.out.println("Student 1 details:");
        student1.displayDetails();

        System.out.println("\nStudent 2
details:");
        student2.displayDetails();
    }
}

```

Unchecked Exception

```

public class UncheckedException{
    public static void main(String[] args) {
        // 1. ArithmeticException
        try {

```

```

        int result = 5 / 0; // Division by zero

        System.out.println("Result: " + result);
    } catch (ArithmeticException e) {

System.out.println("ArithmeticException: " +
e.getMessage());

    }

// 2. NullPointerException
try {
    String text = null;

    int length = text.length(); // Attempt to
access a method on a null object

    System.out.println("Text length: " +
length);
} catch (NullPointerException e) {

System.out.println("NullPointerException: " +
e.getMessage());

    }

// 3. ArrayIndexOutOfBoundsException
try {
    int[] numbers = {1, 2, 3};

    int value = numbers[4]; // Accessing an
array element that doesn't exist

    System.out.println("Value: " + value);
} catch (ArrayIndexOutOfBoundsException
e) {

System.out.println("ArrayIndexOutOfBoundsE
xception: " + e.getMessage());

    }

```

```

// 4. NumberFormatException
try {
    String str = "abc";

    int num = Integer.parseInt(str); //
Parsing a non-integer string

    System.out.println("Number: " + num);
} catch (NumberFormatException e) {

System.out.println("NumberFormatException:
" + e.getMessage());

    }
}

```

Abstract Area Java

```

abstract class Figure
{
    double dim1;
    double dim2;

    Figure(double a, double b)
    {
        dim1 = a;
        dim2 = b;
    }

    abstract double area();
}

class Rectangle extends Figure
{
    Rectangle(double a, double b)
    {
        super(a, b);
    }
}

```

```
// override area for rectangle

double area()
{
    System.out.println("Area for Rectangle.");
    return dim1 * dim2;
}

}

class Triangle extends Figure
{
    Triangle(double a, double b)
    {
        super(a, b);
    }

    double area()
    {
        System.out.println("Area for Triangle.");
        return dim1 * dim2 / 2;
    }
}

class abstractArea
{
    public static void main(String args[])
    {
        Rectangle r = new Rectangle(9, 5);
        Triangle t = new Triangle(10, 8);

        Figure figref;

        figref = r;

        System.out.println("Area is " +
figref.area());

```

```
figref = t;

        System.out.println("Area is " +
figref.area());
    }
}

```

Button text java

```
import javax.swing.*;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class button_text {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Java GUI
Example");

        frame.setDefaultCloseOperation(JFrame.EXIT_
ON_CLOSE);

        frame.setSize(400, 300);

        // Create a text field
        JTextField textField = new JTextField();
        textField.setBounds(20, 20, 200, 30);

        // Create a text area
        JTextArea textArea = new JTextArea();
        textArea.setBounds(20, 60, 350, 150);
        textArea.setEditable(false);

        // Create a button
        JButton button = new JButton("Click
Me");

        button.setBounds(230, 20, 120, 30);

```

```

        // Add an action listener to the button

        button.addActionListener(new
        ActionListener() {

            @Override

            public void
            actionPerformed(ActionEvent e) {

                String text = textField.getText(); //
                Get text from the text field

                textArea.append(text + "\n"); //
                Append text to the text area

                textField.setText(""); // Clear the text
                field

            }

        });

```

```

        // Add components to the frame

        frame.add(textField);

        frame.add(textArea);

        frame.add(button);

```

```

        // Set the layout to null (absolute
        positioning)

        frame.setLayout(null);

        frame.setVisible(true);

    }

}

```

Factorial

```

public class factorial

{

    public static void main(String args[])

```

```

    {

        int n;

        int fact = 1;

        n = Integer.parseInt(args[0]);

        for(int i=1;i<=n;i++){

            fact=fact*i;

        }

        System.out.println("Factorial of "+n+"
        is: "+fact);

    }

}

```

FileIO

```

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

```

```

public class fileIO {

    public static void main(String[] args) {

        String filename = "sample.txt"; //
        Replace with the path to your text file

        int vowelCount = 0;

        int wordCount = 0;

        int charACount = 0;

```

```

        try (BufferedReader reader = new
        BufferedReader(new
        FileReader(filename))) {

```

```

        String line;

        while ((line = reader.readLine()) !=
null) {

            String[] words = line.split("\\s+");
// Split the line into words

            wordCount += words.length; //
Increase word count

            for (String word : words) {

                charACount += word.length() -
word.replace("a", "").length(); // Count
occurrences of 'a'

                for (char c :
word.toLowerCase().toCharArray()) {

                    if (c == 'a' || c == 'e' || c ==
'i' || c == 'o' || c == 'u') {

                        vowelCount++; // Increase
vowel count

                    }

                }

            }

        } catch (IOException e) {

            System.err.println("An error
occurred while reading the file: " +
e.getMessage());

        }

        System.out.println("Number of
vowels: " + vowelCount);

        System.out.println("Number of
words: " + wordCount);

        System.out.println("Number of 'a'
occurrences: " + charACount);

    }

```

```

}

```

File Scanner

```

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class file_scanner {

    public static void main(String[] args) {

        // Specify the path to the file you
want to read

        String filePath = "sample.txt";

        try {

            // Create a File object with the
specified file path

            File file = new File(filePath);

            // Create a Scanner to read from
the file

            Scanner scanner = new
Scanner(file);

            // Read and display the contents of
the file line by line

            while (scanner.hasNextLine()) {

                String line = scanner.nextLine();

                System.out.println(line);

            }

            // Close the scanner

            scanner.close();

```

```

        } catch (FileNotFoundException e) {
            System.err.println("File not found: "
+ e.getMessage());
        }
    }
}

```

Grid Layout

```

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

public class grid_layout {
    public static void main(String[] args) {
        // Create a JFrame
        JFrame frame = new
JFrame("GridLayout Demo");

        frame.setDefaultCloseOperation(JFrame.E
XIT_ON_CLOSE);
        frame.setSize(300, 200);

        // Create a panel with a GridLayout
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3,
2)); // 3 rows and 2 columns

        // Create and add components to the
panel
        panel.add(new JButton("Button 1"));
        panel.add(new JButton("Button 2"));
        panel.add(new JButton("Button 3"));

```

```

        panel.add(new JButton("Button 4"));
        panel.add(new JButton("Button 5"));
        panel.add(new JButton("Button 6"));

```

```

// Add the panel to the frame
frame.add(panel);

```

```

        frame.setVisible(true);
    }
}

```

Hierarchical Java

```

class Vehicle {
    void start() {
        System.out.println("Vehicle started");
    }

    void stop() {
        System.out.println("Vehicle
stopped");
    }
}

class Car extends Vehicle {
    void drive() {
        System.out.println("Car is driving");
    }
}

```

```

class Motorcycle extends Vehicle {

    void ride() {

        System.out.println("Motorcycle is
riding");

    }

}

```

```

public class hierachical {

    public static void main(String[] args) {

        Car car = new Car();

        car.start();

        car.drive();

        car.stop();

        System.out.println();

        Motorcycle motorcycle = new
Motorcycle();

        motorcycle.start();

        motorcycle.ride();

        motorcycle.stop();

    }

}

```

Interface Area

```

// Define the Shape interface with the
area() method

interface Shape {

    double area();

}

```

```

// Create a Rectangle class that
implements the Shape interface

```

```

class Rectangle implements Shape {

    private double length;

    private double width;

    public Rectangle(double length, double
width) {

        this.length = length;

        this.width = width;

    }

    @Override

    public double area() {

        return length * width;

    }

}

```

```

// Create a Triangle class that implements
the Shape interface

```

```

class Triangle implements Shape {

    private double base;

    private double height;

    public Triangle(double base, double
height) {

        this.base = base;

        this.height = height;

    }

}

```



```

@Override

public double area() {
    return 0.5 * base * height;
}
}

public class interfaceArea {

    public static void main(String[] args) {

        // Create a rectangle and calculate its
        area

        Rectangle rectangle = new
        Rectangle(5.0, 4.0);

        double rectangleArea =
        rectangle.area();

        System.out.println("Area of the
        Rectangle: " + rectangleArea);

        // Create a triangle and calculate its
        area

        Triangle triangle = new Triangle(6.0,
        8.0);

        double triangleArea = triangle.area();

        System.out.println("Area of the
        Triangle: " + triangleArea);
    }
}

```

Itemevent java

```

import java.awt.*;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

```

```

public class itemevent {

    private Checkbox checkbox1,
    checkbox2;

    private Label resultLabel;

    public itemevent() {

        Frame frame = new Frame("ItemEvent
        Demo");

        frame.setLayout(new FlowLayout());

        checkbox1 = new Checkbox("Option
        1");

        checkbox2 = new Checkbox("Option
        2");

        resultLabel = new Label("Selected
        Options: ");

        frame.add(checkbox1);

        frame.add(checkbox2);

        frame.add(resultLabel);

        checkbox1.addItemListener(new
        ItemListener() {

            public void
            itemStateChanged(ItemEvent e) {

                updateResultLabel();

            }

        });

        checkbox2.addItemListener(new
        ItemListener() {

```

```

        public void
itemStateChanged(ItemEvent e) {
            updateResultLabel();
        }
    });

    frame.setSize(500, 200);
    frame.setVisible(true);
}

private void updateResultLabel() {
    String result = "Selected Options: ";
    if (checkbox1.getState()) {
        result += "Option 1 ";
    }
    if (checkbox2.getState()) {
        result += "Option 2";
    }
    resultLabel.setText(result);
}

public static void main(String[] args) {
    new itemevent();
}
}

```

Muiltlevel java

```

class Grandparent {
    void displayGrandparent() {

```

```

        System.out.println("Grandparent
class");
    }
}

class Parent extends Grandparent {
    void displayParent() {
        System.out.println("Parent class");
    }
}

class Child extends Parent {
    void displayChild() {
        System.out.println("Child class");
    }
}

public class multilevel {
    public static void main(String[] args) {
        Child child = new Child();
        child.displayGrandparent(); //
Method from Grandparent class
        child.displayParent();    // Method
from Parent class
        child.displayChild();    // Method
from Child class
    }
}

```

Prime No.

```
class prime_num
{
    public static void main(String
args[])
    {
        int n;

        n = Integer.parseInt(args[0]);

        int m = 0;

        int flag = 0;

        m = n/2;

        if(n==0 || n==1)
        {
            System.out.println(n+" is not prime
number");
        }
        else
        {
            for(int i=2;i<=m;i++)
            {
                if(n%i==0)
                {
                    System.out.println(n+" is not
prime number");

                    flag=1;

                    break;
                }
            }

            if(flag==0)
```

```
        {
            System.out.println(n+" is prime
number");
        }
    }
}
```

Rectangle

```
import java.util.Scanner;

class Area {
    double length;

    double breadth;

    void setDim(double len, double brd) {
        length = len;

        breadth = brd;
    }

    double getArea() {
        return length * breadth;
    }

    public static void main(String[] args)
    {
        Scanner input = new
Scanner(System.in);
```

```

        Area rectangle = new Area();

        System.out.print("Enter the length of
the rectangle: ");

        double len = input.nextDouble();

        System.out.print("Enter the breadth
of the rectangle: ");

        double brd = input.nextDouble();

        rectangle.setDim(len, brd);

        double area = rectangle.getArea();

        System.out.println("The area of the
rectangle is: " + area);

        input.close();
    }
}

```

Set functions

```

import java.util.HashSet;
import java.util.Set;

public class set_functions {

    public static void main(String[] args) {

        // Create two sets

        Set<Integer> set1 = new HashSet<>();

        Set<Integer> set2 = new HashSet<>();
    }
}

```

```

        // Add elements to the first set

        set1.add(1);

        set1.add(2);

        set1.add(3);

        set1.add(4);

        // Add elements to the second set

        set2.add(3);

        set2.add(4);

        set2.add(5);

        set2.add(6);

        // Perform union

        Set<Integer> union = new
HashSet<>(set1);

        union.addAll(set2);

        System.out.println("Union: " + union);

        // Perform intersection

        Set<Integer> intersection = new
HashSet<>(set1);

        intersection.retainAll(set2);

        System.out.println("Intersection: " +
intersection);
    }
}

```

```

        // Perform difference (set1 - set2)

        Set<Integer> difference1 = new
HashSet<>(set1);

        difference1.removeAll(set2);
    }
}

```

```

        System.out.println("Difference (set1 -
set2): " + difference1);

        // Perform difference (set2 - set1)
        Set<Integer> difference2 = new
HashSet<>(set2);
        difference2.removeAll(set1);

        System.out.println("Difference (set2 -
set1): " + difference2);
    }
}

```

Static

```

public class static_
{
    // Static variable
    static int staticVariable = 10;

    // Static block
    static
    {
        System.out.println("Static block is
executed.");
        staticVariable = 20;
    }

    // Static method
    static void staticMethod()
    {
        System.out.println("Static method is
called.");

        System.out.println("Static variable
value: " + staticVariable);
    }
}

```

```

    }

    public static void main(String[] args)
    {
        System.out.println("Main method is
called.");

        System.out.println("Static variable
value: " + staticVariable);

        // Calling the static method
        staticMethod();
    }
}

```

Thread

```

class MyThread extends Thread {
    @Override
    public void run() {
        // Code to be executed in the new
thread
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread: " + i);
            try {
                Thread.sleep(1000); // Sleep for 1
second
            } catch (InterruptedException e) {
                System.out.println("Thread
interrupted");
            }
        }
    }
}

```

```
public class thread {  
    public static void main(String[] args) {  
        MyThread myThread = new  
MyThread(); // Create an instance of the  
custom thread class  
        myThread.start(); // Start the thread  
  
        // Code in the main thread  
        for (int i = 1; i <= 5; i++) {  
            System.out.println("Main: " + i);  
            try {  
                Thread.sleep(1000); // Sleep for 1  
second  
            } catch (InterruptedException e) {  
                System.out.println("Main thread  
interrupted");  
            }  
        }  
    }  
}
```