

# 1. Complex Number Arithmetic

```
class Complex {
    double x, y;
    Complex(double r, double i) {
        x = r; y = i;
    }
    static Complex add(Complex a, Complex b) {
        return new Complex(a.x + b.x, a.y + b.y);
    }
    static Complex multiply(Complex a, Complex b) {
        return new Complex(a.x*b.x, a.y*b.y);
    }
    public String toString() {
        return x + " + i" + y;
    }
    public static void main(String[] args) {
        Complex c1 = new Complex(2, 3);
        Complex c2 = new Complex(4, 5);
        String sum = add(c1, c2).toString();
        String mult = multiply(c1, c2).toString();
        System.out.println("Sum: " + sum + "\nMult: " + mult);
    }
}
```

# 2. TwoDim and ThreeDim (Packages)

File: P1/TwoDim.java

```
package P1;
public class TwoDim
{
    private int x, y;
    public TwoDim(int x, int y) {
        this.x=x; this.y=y;
    }
    public String toString() {
        return "x=" + x + " y=" + y;
    }
}
```

File: P2/ThreeDim.java

```
package P2;
import P1.TwoDim;
public class ThreeDim extends TwoDim {
    private int z;
    public ThreeDim() {
        super();
        z=0;
    }
    public ThreeDim(int x, int y, int z) {
        super(x, y);
        this.z=z;
    }
    public String toString() {
        return super.toString() + " z=" + z;
    }
}
```

File: P/Main.java

```
package P;
import P1.TwoDim;
import P2.ThreeDim;
public class Main {
    public static void main(String[] args) {
        TwoDim d1 = new ThreeDim(1, 2, 3);
        System.out.println(d1.toString());
    }
}
```

### 3. Shape Area (Dynamic Dispatch)

P1/Shape.java:

```
package P1;
public abstract class Shape {
    public abstract double area();
}
```

P2/Rectangle.java:

```
package P2;
import P1.Shape;
public class Rectangle extends Shape {
    double l, b;
    public Rectangle(double l, double b) {
        this.l=l;
        this.b=b;
    }
    public double area(){
        return l*b;
    }
}
```

P3/Circle.java:

```
package P3;
import P1.Shape;
public class Circle extends Shape {
    double r;
    public Circle(double r) {
        this.r=r;
    }
    public double area() {
        return 3.14*r*r;
    }
}
```

Main.java (No package):

```
import java.util.Scanner;
import P1.Shape; import P2.Rectangle; import P3.Circle;
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("1.Rect 2.Circle: ");
    }
}
```

```

        int choice = s.nextInt();
        Shape sh = (choice == 1) ? new Rectangle(s.nextDouble(),
s.nextDouble()) : new Circle(s.nextDouble());
        System.out.println("Area: " + sh.area());
    }
}

```

## 4. Custom Exception (UnderAge)

```

class UnderAge extends Exception {
    int age;
    UnderAge(int a) {
        age = a;
    }
    public String toString() {
        return "Under Age: " + age;
    }
}

public class ExceptionDemo {
    static void test(int age) throws UnderAge {
        if (age < 18) throw new UnderAge(age);
        System.out.println("Access Granted");
    }
    public static void main(String[] args) {
        try {
            test(15);
        }
        catch (UnderAge e) {
            System.out.println(e);
        }
    }
}

```

## 5. Stack with Underflow/Overflow

```

class Stack {
    int arr[] = new int[5], top = -1;
    void push(int x) throws Exception {
        if (top == 4) throw new Exception("Overflow");
        arr[++top] = x;
    }
    int pop() throws Exception {
        if (top == -1) throw new Exception("Underflow");
        return arr[top--];
    }
    public static void main(String[] args) {

```

```

        Stack s = new Stack();
        try {
            s.push(10); s.pop(); s.pop();
        }
        catch(Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

## 6. File Copy (Command Line)

```

import java.io.*;
public class FileCopy {
    public static void main(String[] args) throws IOException {
        try (FileInputStream in = new FileInputStream(args[0]);
            FileOutputStream out = new FileOutputStream(args[1])) {
            int b; while ((b = in.read()) != -1) out.write(b);
        }
    }
}

```

## 7. Read Filtered Lines (Try-with-resources)

```

import java.io.*;
public class FilterRead {
    public static void main(String[] args) throws IOException {
        try (BufferedReader br = new BufferedReader(new
FileReader("test.txt"))) {
            String line;
            while ((line = br.readLine()) != null)
                if (line.startsWith("//")) System.out.println(line);
        }
    }
}

```

## 8. Applet with Child Frame (Mouse Events)

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class AppFrame extends Applet {
    Frame f = new Frame("Child");
    public void init() {
        f.setSize(100, 100);
    }
}

```

```

        addMouseListener(new MouseAdapter() {
            public void mouseEntered(MouseEvent e) { f.setVisible(true); }
            public void mouseExited(MouseEvent e) { f.setVisible(false); }
        });
    }
}

```

## 9. Frame with Pink Background

```

import java.awt.*;
public class PinkFrame extends Frame {
    PinkFrame() {
        setBackground(Color.PINK);
        Label lab = new Label("Hello");
        add(lab);
        setSize(300, 200);
        setVisible(true);
    }

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

```

## 10. Applet: Red/Blue Buttons

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class BtnColor extends Applet implements ActionListener {
    public void init() {
        Button r = new Button("Red"), b = new Button("Blue");
        add(r); add(b);
        r.addActionListener(this); b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        setBackground(e.getActionCommand().equals("Red") ? Color.RED :
Color.BLUE);
    }
}

```

## 11. Applet: KeyAdapter

```

import java.applet.*;

```

```
import java.awt.event.*;

public class KeyApp extends Applet {
    public void init() {
        addKeyListener(new KeyAdapter() {
            public void keyTyped(KeyEvent e) { showStatus("Typed character
is: " + e.getKeyChar()); }
        });
    }
}
```

## 12. Applet: Info vs CGPA

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class InfoApp extends Applet implements ActionListener {
    String msg = "";
    public void init() {
        Button a = new Button("A"), b = new Button("B");
        add(a); add(b); a.addActionListener(this);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        msg = e.getActionCommand().equals("A") ? "Name: RAM, Roll: 1, Coll:
ABC" : "CGPA: 9.0";
        repaint();
    }
    public void paint(Graphics g) { g.drawString(msg, 20, 100); }
}
```

## 13. Banner Thread (Scrolling)

```
import java.applet.*;
import java.awt.*;

public class Banner extends Applet implements Runnable {
    String msg = " Hello World ";
    public void init() { new Thread(this).start(); }
    public void run() {
        try {
            while(true) {
                msg = msg.substring(1) + msg.charAt(0);
                repaint();
                Thread.sleep(200);
            }
        }
    }
}
```

```

    }
    } catch(Exception e) {}
}

public void paint(Graphics g) { g.drawString(msg, 50, 30); }
}

```

## 14. Swing Rewrite (Example: Banner and Buttons)

To rewrite Applets in Swing, replace `Applet` with `JApplet` (or `JFrame`) and `Button` with `JButton`.

```

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

public class SwingButtons extends JFrame {
    SwingButtons() {
        setLayout(new FlowLayout());
        JButton r = new JButton("Red")
        JButton b = new JButton("Blue");
        add(r);
        add(b);
        r.addActionListener(e ->
getContentPane().setBackground(Color.RED));
        b.addActionListener(e ->
getContentPane().setBackground(Color.BLUE));
        setSize(300, 300); setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        new SwingButtons();
    }
}

```