1. Basic Java Program with Input/Output

```
import java.util.Scanner;

public class StudentDetails {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter name: ");
        String name = sc.nextLine();

        System.out.print("Enter roll number: ");
        int roll = sc.nextInt();

        System.out.print("Enter marks: ");
        float marks = sc.nextFloat();

        System.out.printf("\n--- Student Details ---\n");
        System.out.printf("Name: %s\nRoll No: %d\nMarks: %.2f\n", name, roll, marks);
    }
}
```

2. Decision-Making System for Library

```
import java.util.Scanner;

public class LibrarySystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] books = {"Java", "Python", "C++"};
        boolean[] available = {true, false, true};

        System.out.print("Enter book name to check: ");
        String book = sc.nextLine();

        boolean found = false;
        for (int i = 0; i < books.length; i++) {
            if (books[i].equalsIgnoreCase(book)) {
                found = true;
                if (available[i])
                    System.out.println(book + " is available for issue.");
                else
                    System.out.println(book + " is currently unavailable.");
                break;
            }
        }

        if (!found)
            System.out.println("Book not found in library.");
    }
}
```

3. Flight Reservation System using Loops

```
import java.util.Scanner;

public class FlightReservation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] passengers = new String[5];
        int count = 0;

        while (count < 5) {
            System.out.print("Enter passenger name (or type 'stop'): ");
            String name = sc.nextLine();
            if (name.equalsIgnoreCase("stop"))
                break;
            passengers[count++] = name;
        }

        System.out.println("\nBooked Passengers:");
        for (int i = 0; i < count; i++) {
            System.out.println((i + 1) + ". " + passengers[i]);
        }
    }
}
```

4. Array-Based Passenger Booking System

```
import java.util.Scanner;

public class PassengerBooking {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[][] bookings = new String[3][3]; // name, seat, class

        for (int i = 0; i < 3; i++) {
            System.out.print("Enter Passenger Name: ");
            bookings[i][0] = sc.nextLine();
            System.out.print("Enter Seat Number: ");
            bookings[i][1] = sc.nextLine();
            System.out.print("Enter Class: ");
            bookings[i][2] = sc.nextLine();
        }

        System.out.println("\nAll Bookings:");
        System.out.println("Name\tSeat\tClass");
        for (int i = 0; i < 3; i++) {
            System.out.println(bookings[i][0] + "\t" + bookings[i][1] + "\t" + bookings[i][2]);
        }
    }
}
```

5. Banking Transaction Tracker using Strings

```
import java.util.*;

public class BankingTransaction {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StringBuilder log = new StringBuilder();

        while (true) {
            System.out.print("Enter transaction (or 'exit'): ");
            String input = sc.nextLine();
            if (input.equalsIgnoreCase("exit")) break;
            log.append(input).append("\n");
        }

        System.out.println("\n--- Transaction Log ---");
        System.out.println(log.toString());
    }
}
```

6. File I/O for Customer Accounts

```
import java.io.*;
import java.util.*;

public class CustomerAccounts {
    public static void main(String[] args) throws IOException {
        File file = new File("accounts.txt");
        Scanner sc = new Scanner(System.in);
        FileWriter fw = new FileWriter(file, true);

        System.out.print("Enter customer name: ");
        String name = sc.nextLine();
        System.out.print("Enter balance: ");
        double balance = sc.nextDouble();

        fw.write(name + " - " + balance + "\n");
        fw.close();

        System.out.println("Data written successfully.");
    }
}
```

7. Shopping Cart with Price Formatting

```
import java.text.DecimalFormat;
import java.util.Scanner;

public class ShoppingCart {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DecimalFormat df = new DecimalFormat("#.00");

        System.out.print("Enter item price: ");
        double price = sc.nextDouble();
        System.out.print("Enter quantity: ");
        int qty = sc.nextInt();

        double total = price * qty;
        double tax = total * 0.18;
        double finalPrice = total + tax;

        System.out.println("Final Price (with 18% tax): Rs." + df.format(finalPrice));
    }
}
```

8. Date and Pattern Matching for Orders

```
import java.util.regex.*;
import java.util.Scanner;

public class OrderPattern {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter order (format: Name1234): ");
        String input = sc.nextLine();

        Pattern p = Pattern.compile("[A-Za-z]+\d+");
        Matcher m = p.matcher(input);

        if (m.matches())
            System.out.println("Valid Order ID format.");
        else
            System.out.println("Invalid format!");
    }
}
```

9. GUI Application with Swing for Personal Assistant

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

public class PersonalAssistant {
    public static void main(String[] args) {
        JFrame f = new JFrame("Personal Assistant");
        JLabel l = new JLabel("Enter Task:");
        JTextField t = new JTextField();
        JButton b = new JButton("Save");

        l.setBounds(30, 40, 100, 30);
        t.setBounds(130, 40, 150, 30);
        b.setBounds(130, 100, 80, 30);

        b.addActionListener(e -> JOptionPane.showMessageDialog(f, "Task Saved: " + t.getText()));

        f.add(l); f.add(t); f.add(b);
        f.setSize(350, 200);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

10. Event and Exception Handling in Personal Assistant App

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

public class PersonalAssistantApp {
    public static void main(String[] args) {
        JFrame f = new JFrame("Event Handling Example");
        JTextField t = new JTextField();
        JButton b = new JButton("Submit");

        t.setBounds(50, 40, 200, 30);
        b.setBounds(100, 100, 100, 30);

        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    if (t.getText().isEmpty()) throw new Exception("Field cannot be empty!");
                    JOptionPane.showMessageDialog(f, "Input: " + t.getText());
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(f, ex.getMessage());
                }
            }
        });

        f.add(t); f.add(b);
        f.setSize(300, 200);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```