**Vidyavardhini's College Of Engineering & Technology**

—-----------------------------------------------------------------------------------------------

NAME : CHETAN BHUYAL ROLL NO. : 16

DATE : 04/08/2025 SUBJECT : FULL STACK JAVA

—-----------------------------------------------------------------------------------------------

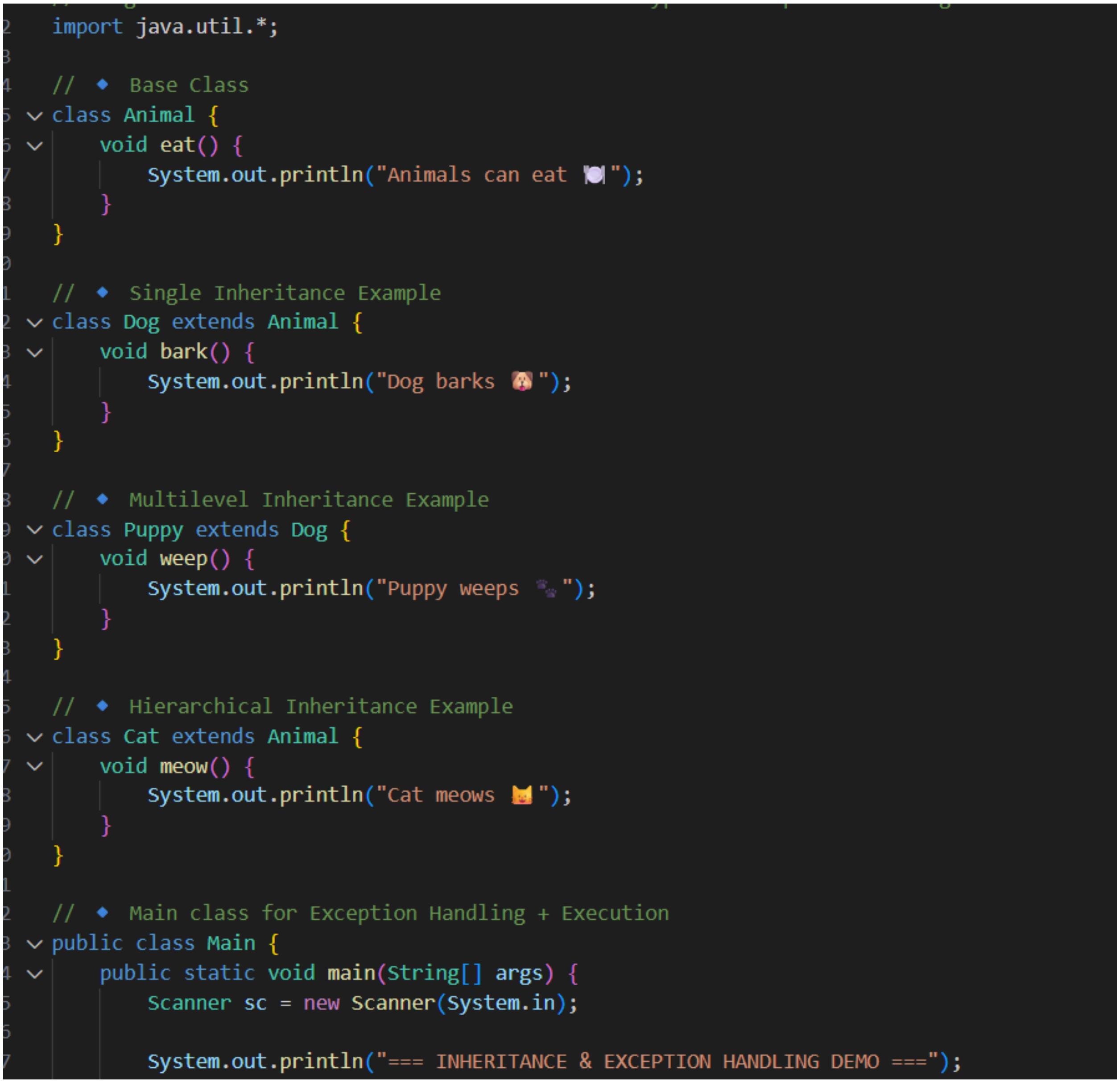
AIM :

Programs on classes and objects with Looping and Branching

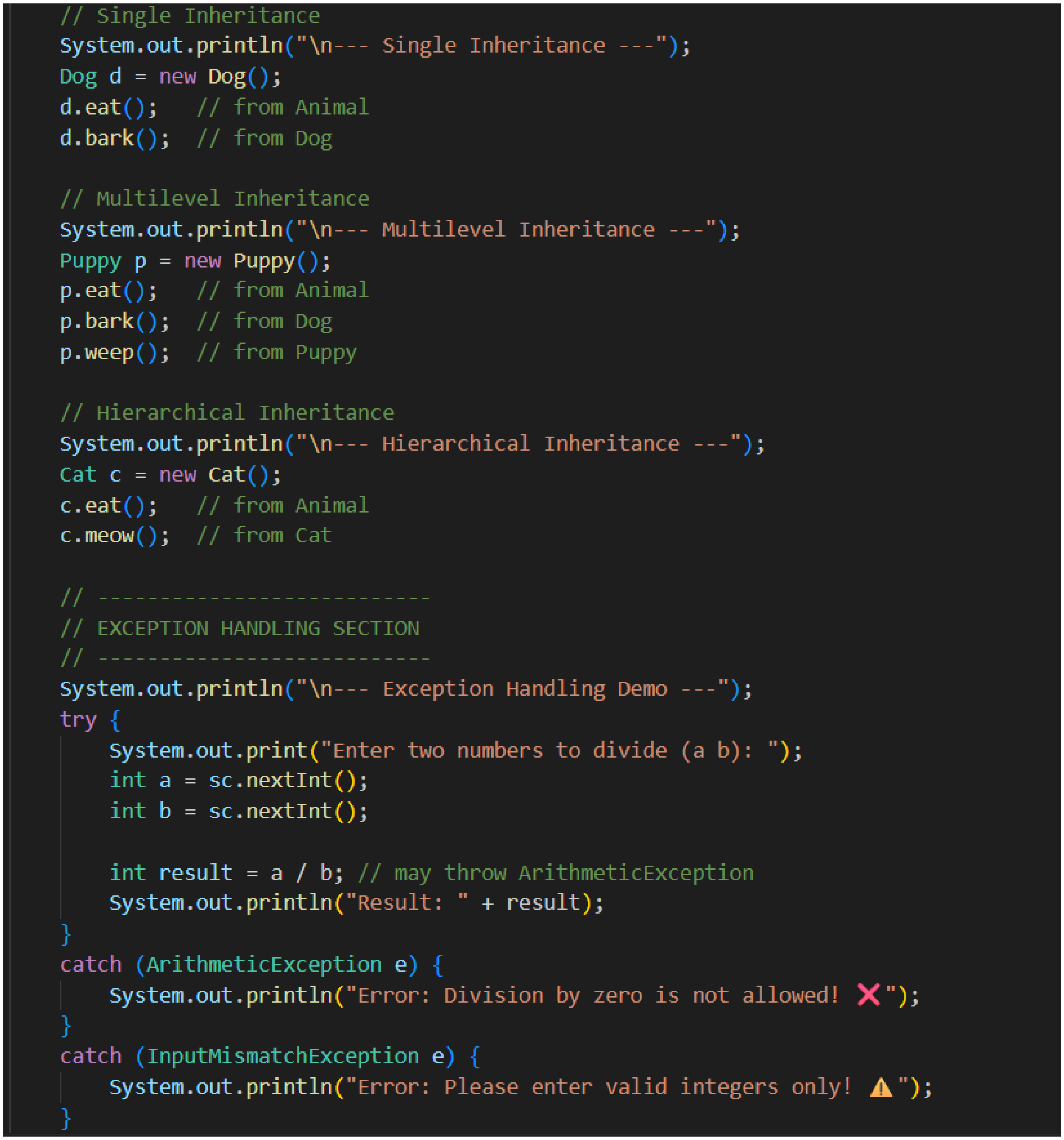
DESCRIPTION :

This program illustrates various types of inheritance (single, multilevel, hierarchical) and handles runtime errors using exception handling in Java.

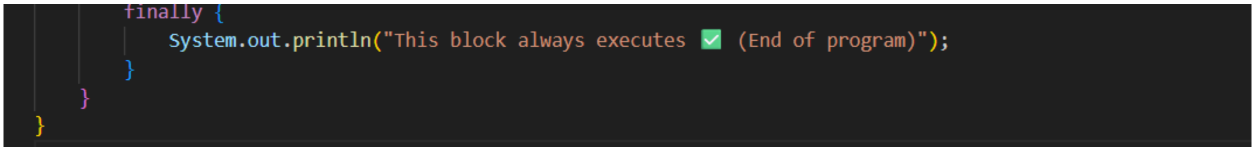
CODE :



```java
import java.util.*;

// 🔹 Base Class
class Animal {
    void eat() {
        System.out.println("Animals can eat 🍽️");
    }
}

// 🔹 Single Inheritance Example
class Dog extends Animal {
    void bark() {
        System.out.println("Dog barks 🐶");
    }
}

// 🔹 Multilevel Inheritance Example
class Puppy extends Dog {
    void weep() {
        System.out.println("Puppy weeps 🐾");
    }
}

// 🔹 Hierarchical Inheritance Example
class Cat extends Animal {
    void meow() {
        System.out.println("Cat meows 🐱");
    }
}

// 🔹 Main class for Exception Handling + Execution
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("=== INHERITANCE & EXCEPTION HANDLING DEMO ===");
```

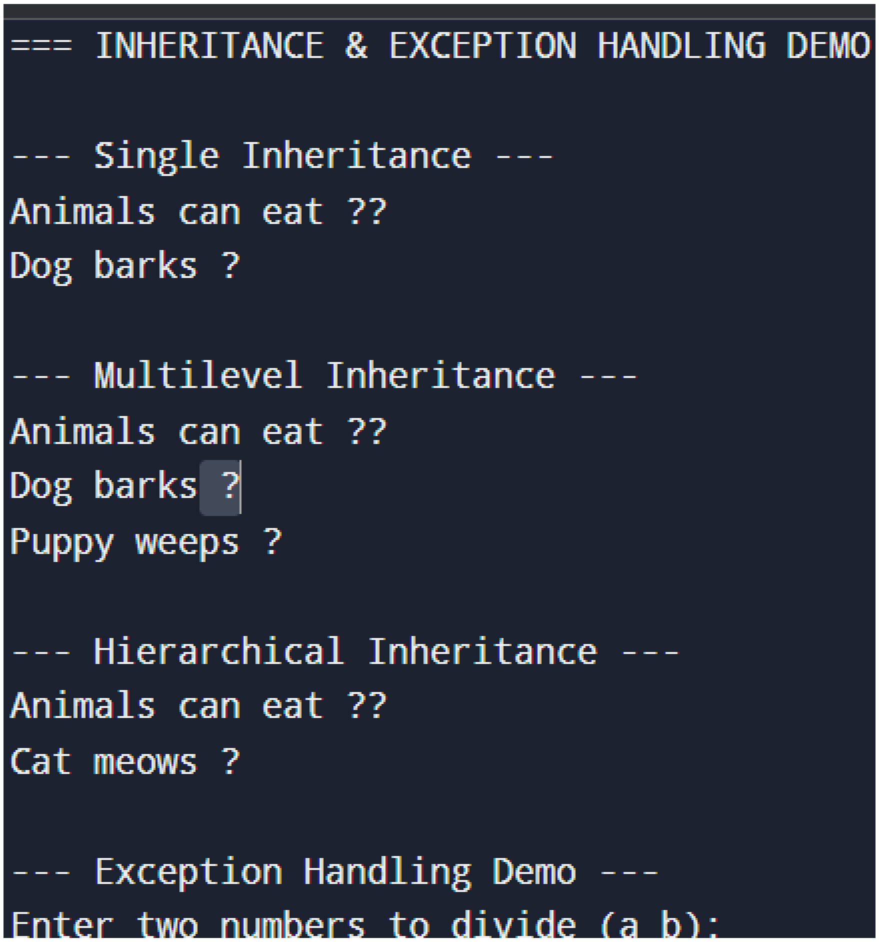


```java
        // Single Inheritance
        System.out.println("\n--- Single Inheritance ---");
        Dog d = new Dog();
        d.eat();   // from Animal
        d.bark();  // from Dog

        // Multilevel Inheritance
        System.out.println("\n--- Multilevel Inheritance ---");
        Puppy p = new Puppy();
        p.eat();   // from Animal
        p.bark();  // from Dog
        p.weep();  // from Puppy

        // Hierarchical Inheritance
        System.out.println("\n--- Hierarchical Inheritance ---");
        Cat c = new Cat();
        c.eat();   // from Animal
        c.meow();  // from Cat

        // ---------------------------
        // EXCEPTION HANDLING SECTION
        // ---------------------------
        System.out.println("\n--- Exception Handling Demo ---");
        try {
            System.out.print("Enter two numbers to divide (a b): ");
            int a = sc.nextInt();
            int b = sc.nextInt();

            int result = a / b; // may throw ArithmeticException
            System.out.println("Result: " + result);
        }
        catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed! ❌");
        }
        catch (InputMismatchException e) {
            System.out.println("Error: Please enter valid integers only! ⚠️");
        }
```



```java
        finally {
            System.out.println("This block always executes ✅ (End of program)");
        }
    }
}
```

## OUTPUT :



```
=== INHERITANCE & EXCEPTION HANDLING DEMO

--- Single Inheritance ---
Animals can eat ??
Dog barks ?

--- Multilevel Inheritance ---
Animals can eat ??
Dog barks ?
Puppy weeps ?

--- Hierarchical Inheritance ---
Animals can eat ??
Cat meows ?

--- Exception Handling Demo ---
Enter two numbers to divide (a b):
```

CONCLUSION :

This program demonstrates different types of inheritance and exception handling in Java. It shows how inheritance promotes code reusability and how `try-catch-finally` makes programs more reliable and error-free. ✅