* **Input:**

import java.util.Scanner;

public class Exp\_01 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// If-Else Statement ->>

System.out.println("If-Else Statement");

System.out.println("Enter your age: ");

int age = sc.nextInt();

if (age >= 18 && age <= 70) {

System.out.println("You can drive.");

}

else if (age >= 15 && age < 18) {

System.out.println("You can opt for learning license.");

}

else {

System.out.println("You can't drive.");

}

// While loop ->>

System.out.println("While loop");

System.out.println("Enter a number whose table you want: ");

int num = sc.nextInt();

System.out.println("Multiplication table for " + num + ": ");

int i = 1;

while (i <= 10) {

System.out.println(num + " x " + i + " = " + (num \* i));

i++;

}

// For loop ->>

System.out.println("For loop");

System.out.println("Enter a number whose table you want: ");

int digit = sc.nextInt();

System.out.println("Multiplication table for " + digit + ":");

for (int j = 0; j <= 10; j++) {

System.out.println(digit + " x " + j + " = " + (digit \* j));

}

// Do while loop ->>

System.out.println("Do-while loop");

System.out.println("Enter a number whose table you want: ");

int number = sc.nextInt();

System.out.println("Multiplication table for " + number + ":");

int k = 1;

do {

System.out.println(number + " x " + k + " = " + (number \* k));

k++;

} while (k<=10);

// Switch Statement ->>

System.out.println("Switch Statement");

System.out.println("Enter your rating on scale of 1-5: ");

int rating = sc.nextInt();

String rated;

switch (rating) {

case 1:

rated = "Worst";

break;

case 2:

rated = "Bad";

break;

case 3:

rated = "Good";

break;

case 4:

rated = "Excellent";

break;

case 5:

rated = "Fabulous";

break;

default:

rated = "Invalid rating";

break;

}

System.out.println("You rated us " + rated + ". Thanks for the feedback!");

}

}

* **Output:**

If-Else Statement

Enter your age:

78

You can't drive.

While loop

Enter a number whose table you want:

12

Multiplication table for 12:

12 x 1 = 12

12 x 2 = 24

12 x 3 = 36

12 x 4 = 48

12 x 5 = 60

12 x 6 = 72

12 x 7 = 84

12 x 8 = 96

12 x 9 = 108

12 x 10 = 120

For loop

Enter a number whose table you want:

10

Multiplication table for 10:

10 x 0 = 0

10 x 1 = 10

10 x 2 = 20

10 x 3 = 30

10 x 4 = 40

10 x 5 = 50

10 x 6 = 60

10 x 7 = 70

10 x 8 = 80

10 x 9 = 90

10 x 10 = 100

Do-while loop

Enter a number whose table you want:

11

Multiplication table for 11:

11 x 1 = 11

11 x 2 = 22

11 x 3 = 33

11 x 4 = 44

11 x 5 = 55

11 x 6 = 66

11 x 7 = 77

11 x 8 = 88

11 x 9 = 99

11 x 10 = 110

Switch Statement

Enter your rating on scale of 1-5:

4

You rated us Excellent. Thanks for the feedback!

* **Input:**

import java.util.Scanner;

public class Exp\_02 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number: ");

int num = sc.nextInt();

if (num%2 == 0) {

System.out.println("Number is even.");

}

else{

System.out.println("Number is odd");

}

System.out.println("Multiplication table for " + num + ":");

for (int j = 0; j <= 10; j++) {

System.out.println(num + " x " + j + " = " + (num \* j));

}

}

}

* **Output:**

Enter a number:

5

Number is odd

Multiplication table for 5:

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

* **Input:**

public class Car {

private String brand;

private String model;

private int year;

// Constructor

public Car(String brand, String model, int year) {

this.brand = brand;

this.model = model;

this.year = year;

}

// Method to display the car details

public void displayDetails() {

System.out.println("Brand: " + this.brand);

System.out.println("Model: " + this.model);

System.out.println("Year: " + this.year);

}

public static void main(String[] args) {

// Create an object of Car class

Car myCar = new Car("Tesla", "Model S", 2020);

// Use display function on the object to display details

myCar.displayDetails();

}

}

* **Output:**

Brand: Tesla

Model: Model S

Year: 2020

* **Input:**

public class Shape {

// Constructor with no parameter

public Shape() {

System.out.println("Default Shape constructor called");

}

// Constructor with one parameter for square

public Shape(int side) {

System.out.println("Square constructor called with side: " + side);

}

// Constructor with two parameters for rectangle

public Shape(int length, int width) {

System.out.println("Rectangle constructor called with length: " + length + " and width: " + width);

}

// Method area with no parameter to display default area message

public void area() {

System.out.println("Default area method called");

}

// Method area with one parameter to display area of a square

public void area(int side) {

System.out.println("Area of square with side " + side + " is: " + (side \* side));

}

// Method area with two parameters for area of rectangle

public void area(int length, int width) {

System.out.println("Area of rectangle with length " + length + " and width " + width + " is: " + (length \* width));

}

public static void main(String[] args) {

// Create objects of shape class using different constructor

Shape shape = new Shape();

Shape square = new Shape(5);

Shape rectangle = new Shape(10, 20);

// Call area methods with different parameters

shape.area();

square.area(5);

rectangle.area(10, 20);

}

}

* **Output:**

Default Shape constructor called

Square constructor called with side: 5

Rectangle constructor called with length: 10 and width: 20

Default area method called

Area of square with side 5 is: 25

Area of rectangle with length 10 and width 20 is: 200

* **STRINGBUFFER:**
* **INPUT:**

import java.io.\*;

class A {

public static void main(String args[])

{

StringBuffer sb = new StringBuffer("Hello ");

sb.append("Java"); // now original string is changed

System.out.println(sb);

}

}

* **OUTPUT:**

Hello Java

* **VECTOR:**
* **INPUT:**

import java.util.Vector;

public class VectorExample {

public static void main(String[] args) {

// Create a vector to store integers

Vector<Integer> vector = new Vector<Integer>();

// Add some integers to the vector

vector.add(10);

vector.add(20);

vector.add(30);

// Print the vector

System.out.println("Vector: " + vector);

// Access an element of the vector

System.out.println("First element: " + vector.firstElement());

// Remove an element from the vector

vector.remove(1);

// Print the vector after removing an element

System.out.println("Vector after removing an element: " + vector);

}

}

* **OUTPUT:**

Vector: [10, 20, 30]

First element: 10

Vector after removing an element: [10, 30]

* **INHERITANCE:**
* **INPUT:**

// Define a superclass

public class Animal {

// Define a method in the superclass

public void makeSound() {

System.out.println("Some sound");

}

}

// Define a subclass that inherits from the superclass

public class Dog extends Animal {

// Override the method in the superclass

@Override

public void makeSound() {

System.out.println("Woof");

}

}

// Use the subclass in a program

public class InheritanceExample {

public static void main(String[] args) {

// Create an instance of the subclass

Dog dog = new Dog();

// Call the method in the subclass

dog.makeSound();

}

}

* **OUTPUT:**

Woof

* **INPUT:**

// Parent class

class Animal {

void eat() {

System.out.println("Animal is eating...");

}

}

// Single inheritance: Child class inherits from one parent class

class Dog extends Animal {

void bark() {

System.out.println("Dog is barking...");

}

}

// Multilevel inheritance: Child class inherits from another child class

class Labrador extends Dog {

void color() {

System.out.println("Labrador is brown in color...");

}

}

// Hierarchical inheritance: Multiple child classes inherit from the same parent class

class Cat extends Animal {

void meow() {

System.out.println("Cat is meowing...");

}

}

public class Exp\_06 {

public static void main(String[] args) {

Dog dog = new Dog();

dog.eat(); // Inherited method from Animal

dog.bark(); // Method specific to Dog

Labrador labrador = new Labrador();

labrador.eat(); // Inherited method from Animal

labrador.bark(); // Inherited method from Dog

labrador.color(); // Method specific to Labrador

Cat cat = new Cat();

cat.eat(); // Inherited method from Animal

cat.meow(); // Method specific to Cat

}

}

* **OUTPUT:**

Animal is eating...

Dog is barking...

Animal is eating...

Dog is barking...

Labrador is brown in color...

Animal is eating...

Cat is meowing...

* **INPUT:**

class Parent {

// Method with no parameters

void display() {

System.out.println("Parent class display method");

}

// Method overloaded with a parameter

void display(String message) {

System.out.println("Parent class display method: " + message);

}

}

class Child extends Parent {

// Overriding the display method of Parent class

@Override

void display() {

super.display(); // Calling the parent class method using super keyword

System.out.println("Child class display method");

}

// Overloading the display method of Parent class

void display(int number) {

System.out.println("Child class display method: " + number);

}

// Final method which cannot be overridden in child classes

final void finalDisplay() {

System.out.println("Final display method in Parent class");

}

}

public class Exp\_07 {

public static void main(String[] args) {

Child child = new Child();

// Method overloading

child.display(); // Calls Child class display() method

child.display("Hello"); // Calls Parent class display(String message) method

child.display(10); // Calls Child class display(int number) method

// Final method

child.finalDisplay(); // Calls finalDisplay() method of Parent class

}

}

* **OUTPUT:**

Parent class display method

Child class display method

Parent class display method: Hello

Child class display method: 10

Final display method in Parent class

* **INPUT:**

public class Exp\_08 {

public static void main(String[] args) {

// Creating a 2D array

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

// Accessing elements of the 2D array

System.out.println("Elements of the 2D array:");

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[i].length; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

// Creating a 3D array

int[][][] cube = {

{

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

},

{

{10, 11, 12},

{13, 14, 15},

{16, 17, 18}

},

{

{19, 20, 21},

{22, 23, 24},

{25, 26, 27}

}

};

// Accessing elements of the 3D array

System.out.println("\nElements of the 3D array:");

for (int i = 0; i < cube.length; i++) {

for (int j = 0; j < cube[i].length; j++) {

for (int k = 0; k < cube[i][j].length; k++) {

System.out.print(cube[i][j][k] + " ");

}

System.out.println();

}

System.out.println();

}

}

}

* **OUTPUT:**

Elements of the 2D array:

1 2 3

4 5 6

7 8 9

Elements of the 3D array:

1 2 3

4 5 6

7 8 9

10 11 12

13 14 15

16 17 18

19 20 21

22 23 24

25 26 27

* **INPUT:**

import java.util.Scanner;

public class Exp\_09 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter two integers:");

int num1 = scanner.nextInt();

int num2 = scanner.nextInt();

try {

int result = divide(num1, num2);

System.out.println("Result of division: " + result);

} catch (ArithmeticException e) {

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

} finally {

System.out.println("Finally block is executed.");

scanner.close();

}

}

public static int divide(int num1, int num2) {

if (num2 == 0) {

throw new ArithmeticException("Division by zero is not allowed");

}

return num1 / num2;

}

}

* **OUTPUT:**

Enter two integers:

8

0

Exception caught: Division by zero is not allowed

Finally block is executed.

* **INPUT:**

import java.io.\*;

public class Exp\_10 {

public static void main(String[] args) {

// Writing to a file

try {

FileWriter writer = new FileWriter("output.txt");

writer.write("Hello, this is a file handling example.\n");

writer.write("We are writing this text to a file.");

writer.close();

System.out.println("Data written to the file successfully.");

} catch (IOException e) {

System.out.println("An error occurred while writing to the file.");

e.printStackTrace();

}

// Reading from a file

try {

FileReader reader = new FileReader("output.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

String line;

System.out.println("\nContent of the file:");

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

System.out.println(line);

}

reader.close();

} catch (IOException e) {

System.out.println("An error occurred while reading the file.");

e.printStackTrace();

}

}

}

* **OUTPUT:**

Data written to the file successfully.

Content of the file:

Hello, this is a file handling example.

We are writing this text to a file.