**LAB 2** JAVA

1.     Write a program that takes a student's score as input and outputs the corresponding grade based on the following scale:

A: 90-100

B: 80-89

C: 70-79

D: 60-69

F: 0-59

**Code:**

**import** java.util.Scanner;

**public** **class** Lab2 {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the student's score: ");

**try** {

**double** score = scanner.nextDouble();

**if** (score < 0 || score > 100) {

System.***out***.println("Invalid score! Please enter a score between 0 and 100.");

} **else** {

**char** grade = *calculateGrade*(score);

System.***out***.println("The student's grade is: " + grade);

}

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid number.");

} **finally** {

scanner.close();

}

}

**public** **static** **char** calculateGrade(**double** score) {

**if** (score >= 90 && score <= 100) {

**return** 'A';

} **else** **if** (score >= 80 && score < 90) {

**return** 'B';

} **else** **if** (score >= 70 && score < 80) {

**return** 'C';

} **else** **if** (score >= 60 && score < 70) {

**return** 'D';

} **else** {

**return** 'F';

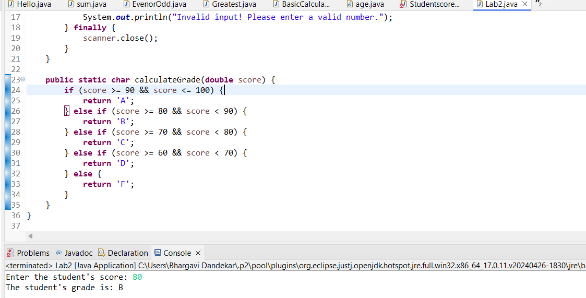
}

}

}

**Output:**

> Uses a Scanner object to take input from the user. It then validates the input score to ensure it falls within the range of 0 to 100. If the input is valid, it calls the calculateGrade() method to determine the corresponding grade based on the given scale.

****

2.     Write a program to check if a given year is a leap year. (A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.)

**Code:**

**import** java.util.Scanner;

**public** **class** Leapyear {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter a year: ");

**try** {

**int** year = scanner.nextInt();

**if** (*isLeapYear*(year)) {

System.***out***.println(year + " is a leap year.");

} **else** {

System.***out***.println(year + " is not a leap year.");

}

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid year.");

} **finally** {

scanner.close();

}

}

**public** **static** **boolean** isLeapYear(**int** year) {

**return** (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

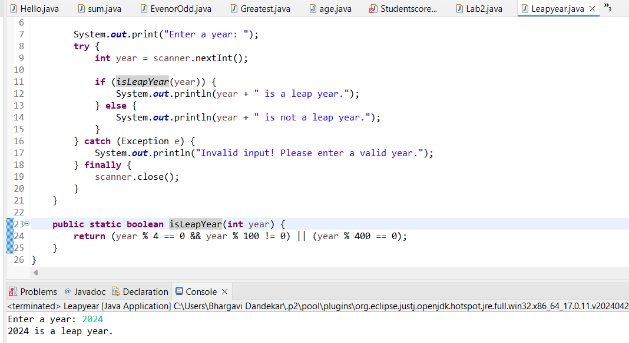
}

}

**Output:**

>A LeapYearChecker class with a main() method is defined.

main() prompts the user to enter a year, then checks if it's a leap year using the is LeapYear() method. LeapYear() method implements the logic to determine if the given year is a leap year according to the provided conditions.



3.     Write a program that takes an integer as input and checks if it is positive, negative, or zero.

**Code:**

**import** java.util.Scanner;

**public** **class** Integer {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter an integer: ");

**try** {

**int** num = scanner.nextInt();

**if** (num > 0) {

System.***out***.println(num + " is positive.");

} **else** **if** (num < 0) {

System.***out***.println(num + " is negative.");

} **else** {

System.***out***.println("The number is zero.");

}

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid integer.");

} **finally** {

scanner.close();

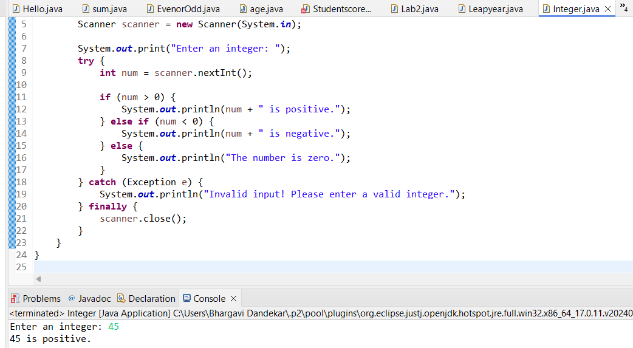
}

}

}

**Output:**

Scanner is used to read the input, and then checks whether the input is positive, negative, or zero. Depending on the result, it prints the corresponding message.



4.     Write a program that prints numbers from 1 to 10 using a loop.

**Code:**

**public** **class** Loop {

**public** **static** **void** main(String[] args) {

// Using a for loop to print numbers from 1 to 10

**for** (**int** i = 1; i <= 10; i++) {

System.***out***.println(i);

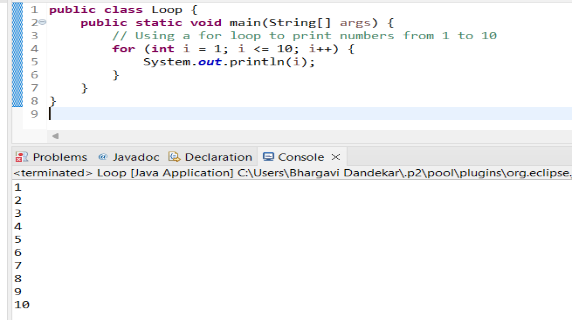
}

}

}

**Output:**

>This program uses a **for** loop to iterate from 1 to 10, and within each iteration, it prints the value of the loop variable **i**. As a result, it prints numbers from 1 to 10.

****

5.     Write a program that takes an integer N as input and calculates the sum of entered numbers.

**Code:**

**import** java.util.Scanner;

**public** **class** Sum {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the value of N: ");

**try** {

**int** N = scanner.nextInt();

**int** sum = 0;

System.***out***.println("Enter " + N + " numbers:");

**for** (**int** i = 0; i < N; i++) {

System.***out***.print("Enter number " + (i + 1) + ": ");

**int** num = scanner.nextInt();

sum += num;

}

System.***out***.println("The sum of the entered numbers is: " + sum);

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid integer.");

} **finally** {

scanner.close();

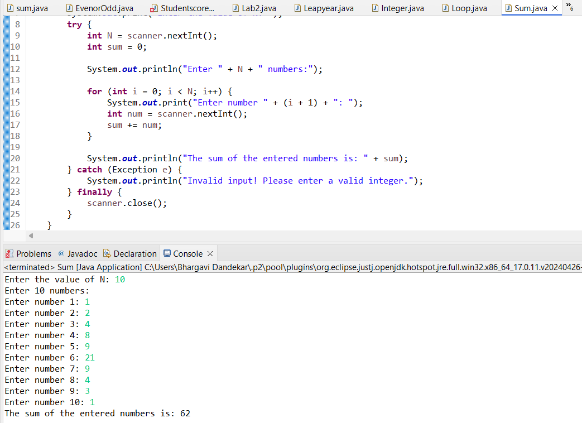
}

}

}

**Output:**

This program prompts the user to enter the value of N, the number of integers to be entered. Then, it prompts the user to enter N integers and calculates their sum. Finally, it prints out the sum of the entered numbers.

****

6.     Write a program that takes an integer as input and prints its multiplication table up to 10.

**Code:**

**public** **class** multiplication {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter an integer: ");

**try** {

**int** num = scanner.nextInt();

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

**for** (**int** i = 1; i <= 10; i++) {

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

}

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid integer.");

} **finally** {

scanner.close();

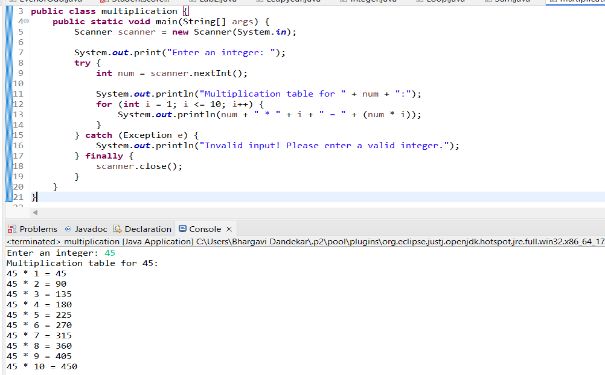
}

}

}

**Output:**

> This program prompts the user to enter an integer, reads the input using Scanner, and then prints the multiplication table of that integer up to 10 using a for loop. Each line of the table shows the multiplication of the input number by the numbers from 1 to 10



7.     Write a program that takes a positive integer as input and prints its digits in reverse order.

**Code:**

**import** java.util.Scanner;

**public** **class** Integerpositivenegative {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter a positive integer: ");

**try** {

**int** number = scanner.nextInt();

**if** (number < 0) {

System.***out***.println("Invalid input! Please enter a positive integer.");

} **else** {

System.***out***.print("Digits in reverse order: ");

**while** (number > 0) {

**int** digit = number % 10;

System.***out***.print(digit);

number /= 10;

}

System.***out***.println();

}

} **catch** (Exception e) {

System.***out***.println("Invalid input! Please enter a valid positive integer.");

} **finally** {

scanner.close();

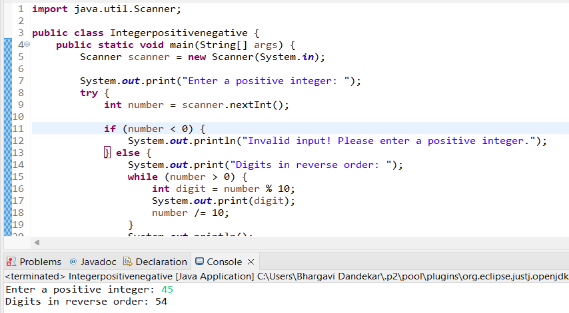
}

}

}

**Output:**

**>** This program prompts the user to enter a positive integer, reads the input using Scanner, and then iterates through the digits of the number while printing them in reverse order. It does this by repeatedly taking the remainder of the number divided by 10 (which gives the last digit), printing it, and then dividing the number by 10 to remove the last digit.

****

8.     Create a class Animal with a method makeSound() that prints "Some generic animal sound". Create another class Dog that extends Animal and overrides the makeSound() method to print "Bark". Write a main method to demonstrate calling the makeSound() method on an Animal reference holding a Dog object.

**Code:**

**class** Animal {

**public** **void** makeSound() {

System.***out***.println("Some generic animal sound");

}

}

**class** Dog **extends** Animal {

@Override

**public** **void** makeSound() {

System.***out***.println("Bark");

}

}

**public** **class** Class {

**public** **static** **void** main(String[] args) {

// Creating an Animal reference holding a Dog object

Animal animal = **new** Dog();

// Calling makeSound() method

animal.makeSound(); // This will call the makeSound() method of Dog class

}

}

**Output:**

>The Animal class has a method makeSound() that prints "Some generic animal sound".

The Dog class extends Animal and overrides the makeSound() method to print "Bark".

In the main method, an Animal reference animal is created and initialized with a Dog object.

When animal.makeSound() is called, it invokes the makeSound() method of the Dog class, since animal is holding a Dog object. Therefore it prints "Bark".

