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

**Program:-**

**Package demo;**

**import** java.util.Scanner;

**public** **class** GradeCalculator {

// Function to determine the grade based on the score

**public** **static** String determineGrade(**double** score) {

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

**return** "A";

} **else** **if** (score >= 80 && score <= 89) {

**return** "B";

} **else** **if** (score >= 70 && score <= 79) {

**return** "C";

} **else** **if** (score >= 60 && score <= 69) {

**return** "D";

} **else** **if** (score >= 0 && score <= 59) {

**return** "F";

} **else** {

**return** "Invalid score";

}

}

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

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

**try** {

// Get the score from the user

System.***out***.print("Enter the student's score (0-100): ");

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

// Determine and print the grade

String grade = *determineGrade*(score);

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

} **catch** (Exception e) {

System.***out***.println("Please enter a valid number for the score.");

} **finally** {

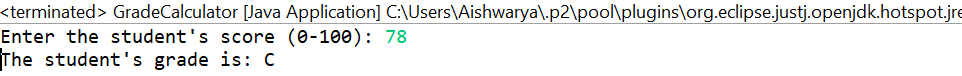
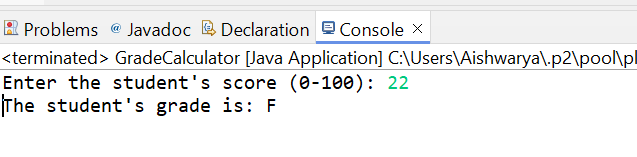
scanner.close();

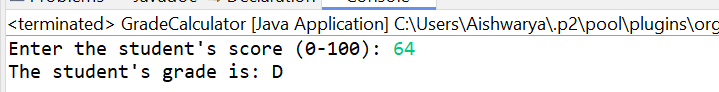
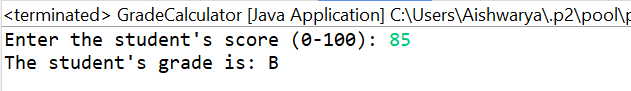
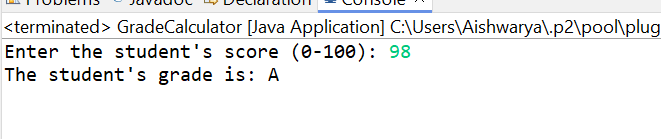
}

}

}

Output :-





1. 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.)

**package** demo;

**import** java.util.Scanner;

**public** **class** LeapYearChecker {

// Function to determine if a year is a leap year

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

**if** (year % 400 == 0) {

**return** **true**;

} **else** **if** (year % 100 == 0) {

**return** **false**;

} **else** **if** (year % 4 == 0) {

**return** **true**;

} **else** {

**return** **false**;

}

}

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

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

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

**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.");

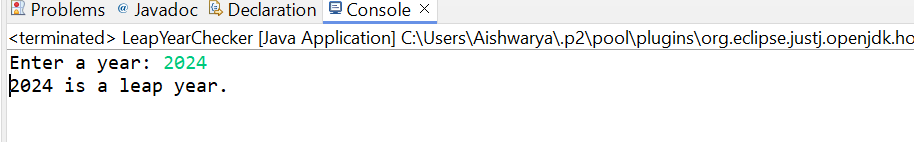
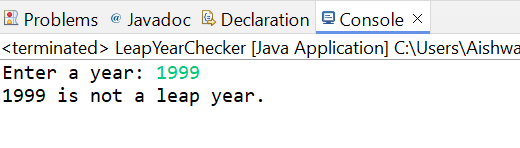
}

scanner.close();

}

}

Output:-



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

Program:-

**package** demo;

**import** java.util.Scanner;

**public** **class** NumberChecker {

// Function to determine if the number is positive, negative, or zero

**public** **static** String checkNumber(**int** number) {

**if** (number > 0) {

**return** "positive";

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

**return** "negative";

} **else** {

**return** "zero";

}

}

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

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

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

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

// Determine and print if the number is positive, negative, or zero

String result = *checkNumber*(number);

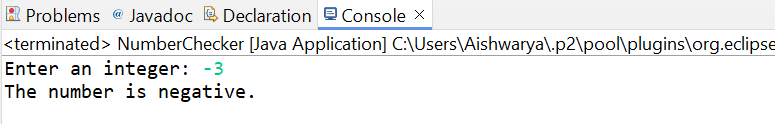
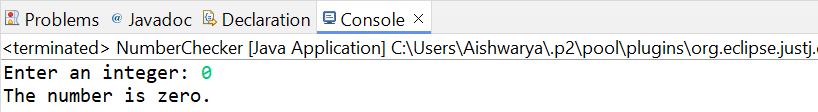
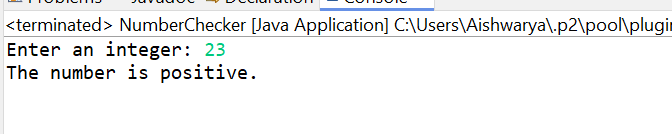
System.***out***.println("The number is " + result + ".");

scanner.close();

}

}

Output:-



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

**Program:-**

**package** demo;

**public** **class** NumberPrinter {

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

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

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

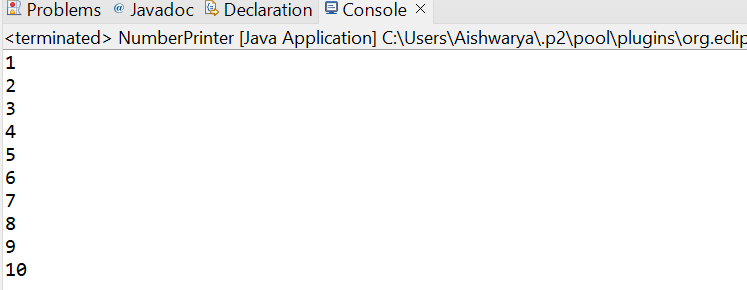
System.***out***.println(i);

}

}

}

Output:-



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

Program:-

**package** demo;

**import** java.util.Scanner;

**public** **class** SumCalculator {

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

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

// Prompt the user to enter the number of integers (N)

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

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

// Initialize the sum to 0

**int** sum = 0;

// Loop to read N integers and calculate the sum

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

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

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

sum += number;

}

// Print the calculated sum

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

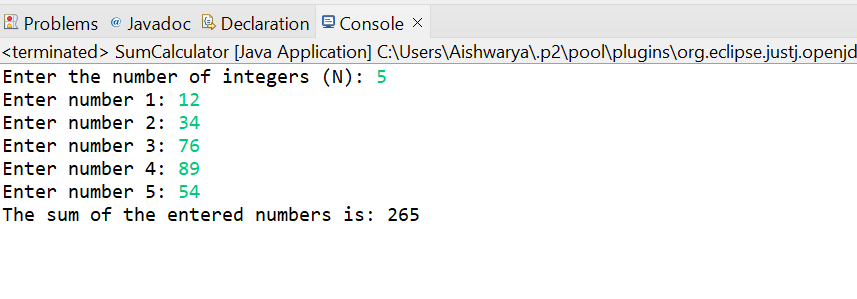
// Close the scanner

scanner.close();

}

}

Output :-



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

**package** demo;

**import** java.util.Scanner;

**public** **class** MultiplicationTable {

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

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

// Prompt the user to enter an integer

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

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

// Print the multiplication table up to 10

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

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

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

}

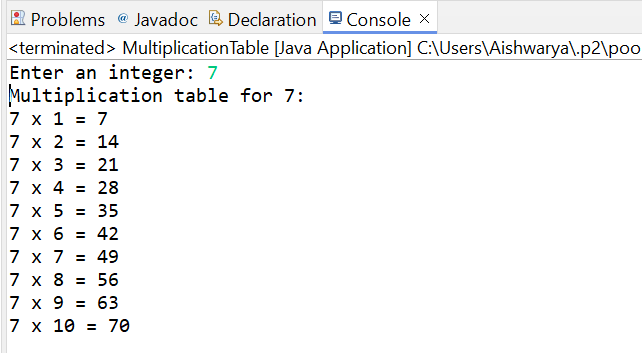
// Close the scanner

scanner.close();

}

}

Output:-



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

**package** demo;

**import** java.util.Scanner;

**public** **class** ReverseDigits {

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

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

// Prompt the user to enter a integer

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

**while** (!scanner.hasNextInt()) {

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

scanner.next(); // Clear the invalid input

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

}

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

// Ensure the number is positive

**while** (number <= 0) {

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

number = scanner.nextInt();

}

// Print the digits in reverse order

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

**while** (number > 0) {

**int** digit = number % 10;

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

number /= 10;

}

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

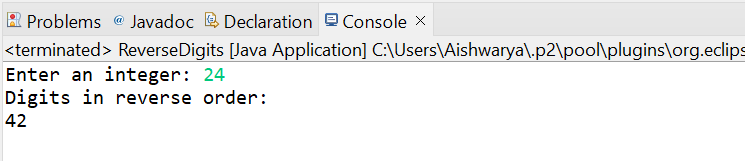
// Close the scanner

scanner.close();

}

}

Output:-



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.

Program:-

**package** demo;

// Animal class

**class** Animal {

// Method to make sound

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

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

}

}

// Dog class extending Animal

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

// Override makeSound method

@Override

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

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

}

}

// Main class

**public** **class** Main {

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

// Create an Animal reference holding a Dog object

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

// Call makeSound method

animal.makeSound(); // Output will be "Bark"

}

}

Output:-

