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:-

package Lab\_2;

import java.util.Scanner;

public class StudentGradeConverter {

public static void main(String[] args) {

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

// Take User Input

System.***out***.print("Enter Student Marks (0-100) : ");

int score = scanner.nextInt();

scanner.close();

// converting marks to grade

char grade = *marksToGradeConverter*(score);

// Output

System.***out***.println("Student Grade is : " + grade);

}

public static char marksToGradeConverter(int score) {

char grade;

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

grade = 'A';

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

grade = 'B';

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

grade = 'C';

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

grade = 'D';

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

grade = 'F';

} else {

throw new IllegalArgumentException("Score must be between 0 and 100");

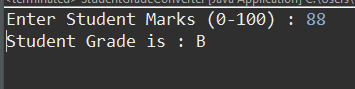
}

return grade;

}

}

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.)

Code :-

package Lab\_2;

import java.util.Scanner;

public class LeapYearChecker {

public static void main(String[] args) {

// User Input

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

System.***out***.print("Enter Year : ");

int year = sc.nextInt();

sc.close();

// if year is divisible by 400 then year is leap year OR

// if year is divisible by 4 and not by 100 then year is leap year

// otherwise year is not leap year

if (year%400==0 || (year%4==0 && year%100!=0) ) {

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

} else {

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

}

}

}

Output :-

A screenshot of a computer

Description automatically generated

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

Code:-

package Lab\_2;

import java.util.Scanner;

public class NumberPositiveNegativeZero {

public static void main(String[] args) {

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

// Take User Input

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

Double num = scanner.nextDouble();

scanner.close();

*checkNumber*(num);

}

public static void checkNumber(Double n) {

if (n == 0) {

System.***out***.println(n + " is Zero.");

} else if (n < 0) {

System.***out***.println(n + " is Negative.");

} else if (n > 0) {

System.***out***.println(n + " is Positive.");

} else {

throw new IllegalArgumentException("Invalid Number");

}

}

}

Output :-

A screenshot of a computer

Description automatically generated

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

Code :-

package Lab\_2;

public class PrintNumber {

public static void main(String[] args) {

// Print number from 1 to 10

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

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

}

}

}

Output:-

A screenshot of a computer

Description automatically generated

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

Code :-

package Lab\_2;

import java.util.Scanner;

public class SumOfDigits{

public static void main(String[] args) {

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

// Take User Input

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

int num = sc.nextInt();

sc.close();

int sum = 0;

int currentNum = num;

int reminder = 0;

while (currentNum>0) {

// get last digit

reminder = currentNum % 10;

// add last digit to sum variable

sum = sum + reminder;

// divide number by 10 and store in integer variable

// so the last digit will remove

currentNum = currentNum / 10;

}

System.***out***.println("Sum of number is : " + sum);

}

}

Output :-

A screenshot of a computer

Description automatically generated

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

Code :-

package Lab\_2;

import java.util.Scanner;

public class MultiplicationTable{

public static void main(String[] args) {

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

// Take User Input

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

int num = sc.nextInt();

sc.close();

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

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

}

}

}

Output :-

A screenshot of a computer

Description automatically generated

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

Code :-

package java\_06\_25\_2024;

import java.util.Scanner;

public class HomeWork4 {

public static void main(String[] args) {

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

// Take User Input

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

int num = sc.nextInt();

sc.close();

String ans = "";

int currentNum = num;

int reminder = 0;

while (currentNum>0) {

// Get last digit of number

reminder = currentNum % 10;

// Add last digit to 'ans' String

// Because ans is String it will not add number

// it will concatenate the number and string

ans = ans + reminder;

// remove last digit from number

currentNum = currentNum / 10;

}

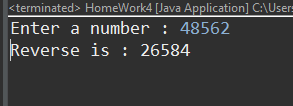
// print answer

System.***out***.println("Reverse is : " + ans);

}

}

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.

Code :-

package Lab\_2;

class Animal {

void makeSound() {

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

}

}

class Dog extends Animal {

*@Override*

void makeSound() {

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

}

}

public class OverrideExample {

public static void main(String[] args) {

Dog dogObject = new Dog();

// Animal reference holding a Dog object.

Animal animalObject = dogObject;

// animal makeSound method call

animalObject.makeSound();

}

}

Output :-

A screen shot of a computer

Description automatically generated