**LAB 2**

**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** lect;

**import** java.util.Scanner;

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

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

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

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

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

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

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

scanner.close();

}

**public** **static** **char** calculateGrade(**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** {

grade = 'F';

}

**return** grade;

}

}

**Output:**

A screenshot of a computer

Description automatically generated

**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.)   
Program:**

**package** lect;

**import** java.util.Scanner;

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

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

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

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();

}

// Method to check if a year is a leap year

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

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

**return** **true**;

} **else** {

**return** **false**;

}

}

}

**Output:**

A screen shot 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**

**Program:**

**package** lect;

**import** java.util.Scanner;

**public** **class** NumberClassifier {

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

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

// Input the number from the user

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

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

// Check if the number is positive, negative, or zero

**if** (number > 0) {

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

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

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

} **else** {

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

}

scanner.close();

}

}

**Output:**



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

**Program:**

**package** lect;

**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:**

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**.

**Program:**

**package** lect;

**import** java.util.Scanner;

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

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

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

// Input the value of N

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

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

// Input N integers and calculate their sum

**int** sum = 0;

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

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

sum += scanner.nextInt();

}

// Display the sum

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

scanner.close();

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

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

**Program:**

**package** lect;

**import** java.util.Scanner;

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

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

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

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

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

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

// Using a for loop to print the multiplication table up to 10

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

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

}

scanner.close();

}

}

**Output:**

A screenshot of a math problem

Description automatically generated

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

**Program:**

**package** lect;

**import** java.util.Scanner;

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

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

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

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

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

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

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

} **else** {

System.***out***.println("Numbers in decreasing order:");

// Print numbers in decreasing order from the input number down to 1

**for** (**int** i = number; i >= 1; i--) {

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

}

}

scanner.close();

}

}

**Output:**

A screenshot of a computer

Description automatically generated

**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** lect;

//Parent class

**class** Animal {

// Method to make sound

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

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

}

}

//Child class extending Animal

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

// Method overriding makeSound() to bark

@Override

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

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

}

}

//Main class

**public** **class** AnimalDemo {

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

// Creating Animal reference holding a Dog object

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

// Demonstrating calling the makeSound() method

animal.makeSound(); // This will call Dog's makeSound() method

}

}

**Output:**

