Internship Report

Internship report submitted in fulfillment of the requirement for the certificate of the internship

By

Your Name: **ARPAN KUNDU**

College Name: **SANAKA EDUCATIONAL TRUST GROUP OF INSTITUTIONS**

Domain Name: **JAVA DEVELOPER**

Batch (JUNE-JULY)

Date of Submission: 30th JULY

2024

DECLARATION

I hereby declare that the projects entitled "**SIMPLE CALCULATOR**" and submitted for the completion of my internship at InternJunction, are my original work. These projects have not been submitted for any degree, diploma, or other internship programs at any other organizations.

Name: Arpan Kundu

Place: Durgapur

Date: 20th july

ACKNOWLEDGEMENT

It is my proud privilege to express my heartfelt gratitude to several individuals who have assisted me directly or indirectly in completing this project. First and foremost, I would like to extend my deepest appreciation to my internship guides at InternJunction. Their sincere guidance, unwavering support, and invaluable inspiration have been instrumental in the successful completion of this internship.

The insights and knowledge I gained during this period have been profound and enlightening, significantly broadening my understanding of the topics at hand. This experience has not only deepened my academic and professional knowledge but also prepared me for future challenges and opportunities.

I am genuinely grateful for the opportunities provided by InternJunction, which have allowed me to explore and engage with new and exciting avenues of knowledge. This internship has been an enriching journey, and I am confident that the skills and insights I have gained will be of immense benefit in my future endeavours.

To all those who have contributed to this project, whether directly or indirectly, I extend my sincere thanks. Your support and encouragement have been invaluable, and I am truly indebted to you all.

TABLE OF CONTENTS

INTRODUCTION

Project Report: Simple Calculator (Console-Based)

Project Overview

Project Name:

Simple Calculator (Console-Based)

Description:

The Simple Calculator is a console-based application designed to perform basic arithmetic operations. It supports addition, subtraction, multiplication, and division for various numeric types including integers, floats, and doubles. This project aims to provide a user-friendly tool for everyday calculations while serving as an educational resource for individuals interested in programming and understanding the core principles of object-oriented programming (OOP).

Scope of the Project

Functional Scope: *Basic Calculations:*

1. Perform addition, subtraction, multiplication, and division.

2.Support for different numeric types (integers, floats, doubles).

Educational Scope: *Learning Resource:*

1. Ideal for beginners interested in coding and software development.

2. Provides practical experience in implementing arithmetic operations.

Demonstrates the four pillars of OOP:

1.Encapsulation

2.Abstraction

3.Inheritance

4.Polymorphism

Objectives

Primary Objective:

To create a simple, user-friendly calculator application for daily use that performs accurate arithmetic operations based on user input.

Secondary Objectives:

1. Educational Goals:

- Teach beginners the basics of programming through a practical project.

- Enhance understanding of OOP concepts through hands-on implementation.

2. Industry-Level Learning:

- Gain experience in developing a console-based application.

- Understand the process of creating, testing, and refining software.

Features

User Input:

- Accepts two numbers and an operation choice (addition, subtraction, multiplication, division) from the user.

Output:

- Displays the accurate result of the chosen arithmetic operation.

Error Handling:

- Manages invalid inputs and division by zero scenarios gracefully.

METHODOLOGY

```
public class SimpleCalculator
{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);

        try {
            System.out.println("Please Enter your choice: ");
        System.out.println("PRESS '1' FOR ADDITION\tPRESS '2' FOR
SUBTRACTION\tPRESS '3' FOR MULTIPLICATION\tPRESS '4' FOR DIVISION: ");
        int sch=sc.nextInt();
        switch (sch) {
            case 1:new TakeInput(sch).input();
                break;
            case 2:new TakeInput(sch).input();
                break;
            case 3:new TakeInput(sch).input();
                break;
            case 4:new TakeInput(sch).input();
                break;
            default:System.out.println("Please enter valid choice.");
        }
        } catch (Exception e) {
            System.out.println(e.getMessage());
            System.out.println("Please enter the right option which is
provided above... ");
        }
        finally
        {
            sc.close();
        }
    }
}
```

Main Class: SimpleCalculator

The `SimpleCalculator` class serves as the entry point for our console-based calculator application. It includes the `main` method, where the application flow begins. Exception handling is implemented using a try-catch block to manage any unexpected errors that may arise during the execution.

Functionality

1.User Interaction:

- The application prompts the user to choose the desired operation by entering a number corresponding to the operation:
  - `1` for addition
  - `2` for subtraction
  - `3` for multiplication
  - `4` for division
- After selecting an operation, the program calls the `input()` method from the `TakeInput` class.

2.Operation Handling:

- The user's choice is managed using a switch-case statement, which determines the appropriate arithmetic operation.
- The chosen operation is then passed to the `TakeInput` class constructor for further processing.

In this implementation, the SimpleCalculator class handles user input and directs the flow of the program based on the selected operation. By utilizing a try-catch block, it ensures that any unexpected exceptions are caught and managed gracefully, providing a robust and user-friendly experience.

```
class TakeInput
{
    int op=0;
    TakeInput(){}
    public TakeInput(int op)
    {
        this.op=op;
    }
    int aI=0;
    int bI=0;
    double aD=0.0;
    double bD=0.0;
    int ch1=0;
    int ch2=0;
    Scanner sc=new Scanner(System.in);
    public void input()
    {
        try {
            System.out.print("Please enter a number: (IF THE NUMBER HAVE NO
DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT
THEN PRESS 2): ");
            ch1=sc.nextInt();
            System.out.println();
```

```
            if(ch1==1)
                aI=sc.nextInt();
            else
                aD=sc.nextDouble();

            System.out.println("Enter another number: (IF THE NUMBER HAVE NO
DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT
THEN PRESS 2): ");
            ch2=sc.nextInt();
            System.out.println();
            if(ch2==1)
                bI=sc.nextInt();
            else
                bD=sc.nextDouble();
        } catch (Exception e) {
            System.out.println(e.getMessage());
            System.out.println("Please enter the right option which is
provided above... ");
        }

        //Redirect to that method.
        if(op==1){
            Addition ob=new Addition();
            if(ch1==1 && ch2==1)
                ob.add(aI,bI);
            else if(ch1==2 && ch2==2)
                ob.add(aD, bD);
            else if(ch1==1 && ch2==2)
                ob.add(aI, bD);
            else if(ch1==2 && ch2==1)
                ob.add(aD,bI);
        }
        else if(op==2){
            Subtraction ob=new Subtraction();
            if(ch1==1 && ch2==1)
                ob.sub(aI,bI);
            else if(ch1==2 && ch2==2)
                ob.sub(aD, bD);
            else if(ch1==1 && ch2==2)
                ob.sub(aI, bD);
            else if(ch1==2 && ch2==1)
                ob.sub(aD,bI);
        }
        else if(op==3){
            Multiplication ob=new Multiplication();
            if(ch1==1 && ch2==1)
                ob.mul(aI,bI);
            else if(ch1==2 && ch2==2)
```

```
                ob.mul(aD, bD);
            else if(ch1==1 && ch2==2)
                ob.mul(aI, bD);
            else if(ch1==2 && ch2==1)
                ob.mul(aD,bI);
        }
        else if(op==4){
            Division ob=new Division();
            if(ch1==1 && ch2==1)
                ob.div(aI,bI);
            else if(ch1==2 && ch2==2)
                ob.div(aD, bD);
            else if(ch1==1 && ch2==2)
                ob.div(aI, bD);
            else if(ch1==2 && ch2==1)
                ob.div(aD,bI);
        }
    }
}
```

TakeInput Class

The `TakeInput` class is designed to handle user inputs and perform the chosen arithmetic operation. It stores the user's operation choice and processes the operands accordingly. This class ensures that the input is correctly interpreted and the appropriate calculation is performed.

Functionality

1. **Variable Declaration:**
    - `op`: Stores the user's choice of operation, passed through the `TakeInput` class constructor.
    - `aI` and `bI`: Integer variables to store user-provided integer values.
    - `aD` and `bD`: Double variables to store user-provided double values.
    - `choice1` and `choice2`: Integer variables to determine whether the user-provided values are integers or doubles.
2. **User Input Handling:**
    - The class prompts the user to input two operands.
    - It uses if-else statements to check the user's operation choice and then calls the corresponding method to perform the calculation.

In this implementation, the `TakeInput` class is responsible for collecting and validating the user's inputs. It checks whether the inputs are integers or doubles

and processes them accordingly. Based on the user's choice of operation, the class redirects the flow to the appropriate method to perform the calculation and display the result. This approach ensures flexibility in handling different types of numeric inputs and provides a clear structure for performing arithmetic operations.

```
class Addition{
    public void add(int a , int b)
    {
        System.out.println("THE RESULT OF "+a+" + "+b+" IS "+"="+(a+b));
    }
    public void add(int a, double b)
    {
        System.out.println("THE RESULT OF "+a+" + "+b+" IS "+"="+(a+b));
    }
    public void add(double a, int b)
    {
        System.out.println("THE RESULT OF "+a+" + "+b+" IS "+"="+(a+b));
    }
    public void add(double a , double b)
    {
        System.out.println("THE RESULT OF "+a+" + "+b+" IS "+"="+(a+b));
    }
}
class Subtraction{
    public void sub(int a , int b)
    {
        System.out.println("THE RESULT OF "+a+" - "+b+" IS "+"="+(a-b));
    }
    public void sub(int a, double b)
    {
        System.out.println("THE RESULT OF "+a+" - "+b+" IS "+"="+(a-b));
    }
    public void sub(double a, int b)
    {
        System.out.println("THE RESULT OF "+a+" - "+b+" IS "+"="+(a-b));
    }
    public void sub(double a , double b)
    {
        System.out.println("THE RESULT OF "+a+" - "+b+" IS "+"="+(a-b));
    }
}
class Division{
    public void div(int a , int b)
    {
        System.out.println("THE RESULT OF "+a+" / "+b+" IS "+"="+(a/b));
    }
    public void div(int a, double b)
```

```
    {
        System.out.println("THE RESULT OF "+a+" / "+b+" IS "+"="+(a/b));
    }
    public void div(double a, int b)
    {
        System.out.println("THE RESULT OF "+a+" / "+b+" IS "+"="+(a/b));
    }
    public void div(double a , double b)
    {
        System.out.println("THE RESULT OF "+a+" / "+b+" IS "+"="+(a/b));
    }
}
class Multiplication{
    public void mul(int a , int b)
    {
        System.out.println("THE RESULT OF "+a+" * "+b+" IS "+"="+(a*b));
    }
    public void mul(int a, double b)
    {
        System.out.println("THE RESULT OF "+a+" * "+b+" IS "+"="+(a*b));
    }
    public void mul(double a, int b)
    {
        System.out.println("THE RESULT OF "+a+" * "+b+" IS "+"="+(a*b));
    }
    public void mul(double a,double b)
    {
        System.out.println("THE RESULT OF "+a+" * "+b+" IS "+"="+(a*b));
    }
}
```

Overview of Arithmetic Operation Classes: Class Structure

1. Addition Class

2. Subtraction Class

3. Multiplication Class

4. Division Class

These classes represent the four basic arithmetic operations: addition, subtraction, multiplication, and division. Each class contains methods to perform operations on both integer and double inputs.

Key Concepts: ***Polymorphism***

The implementation of these classes demonstrates the concept of polymorphism. Depending on the type of user-provided inputs (integer or double), the appropriate method is executed to perform the calculation. This allows the same operation to be performed on different data types, showcasing the flexibility and power of polymorphism in object-oriented programming.

Execution Flow:

1. User Input:

- The user selects an arithmetic operation (addition, subtraction, multiplication, or division) and provides two operands.

2. Method Selection:

- Based on the user input, the corresponding method for the chosen operation and operand types (integer or double) is selected.

- Polymorphism ensures that the correct method is called, matching the data type of the operands.

3. Calculation and Output:

- The selected method is executed, performing the calculation with the provided operands.

- The result is displayed to the user.

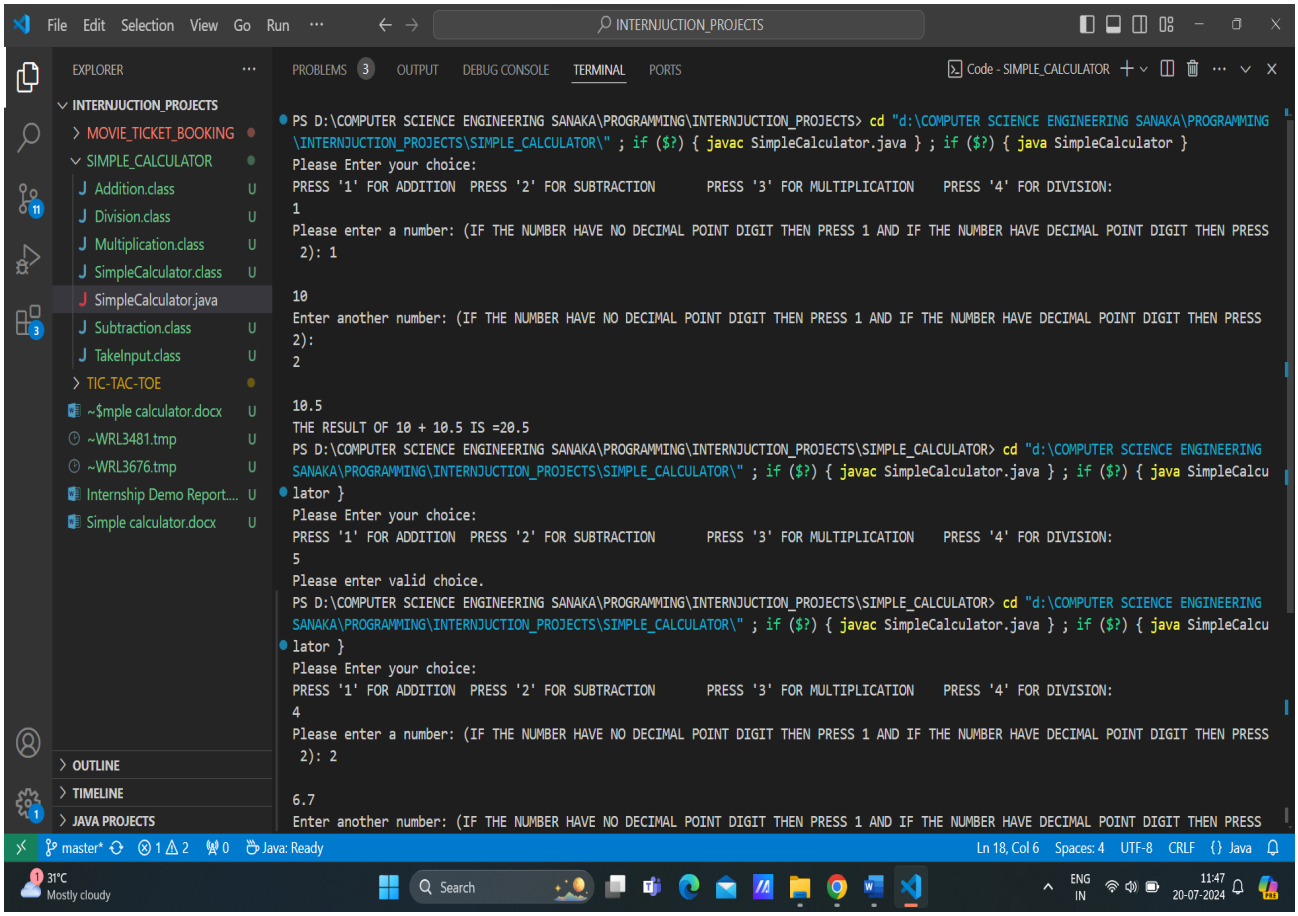
4. Program Termination:

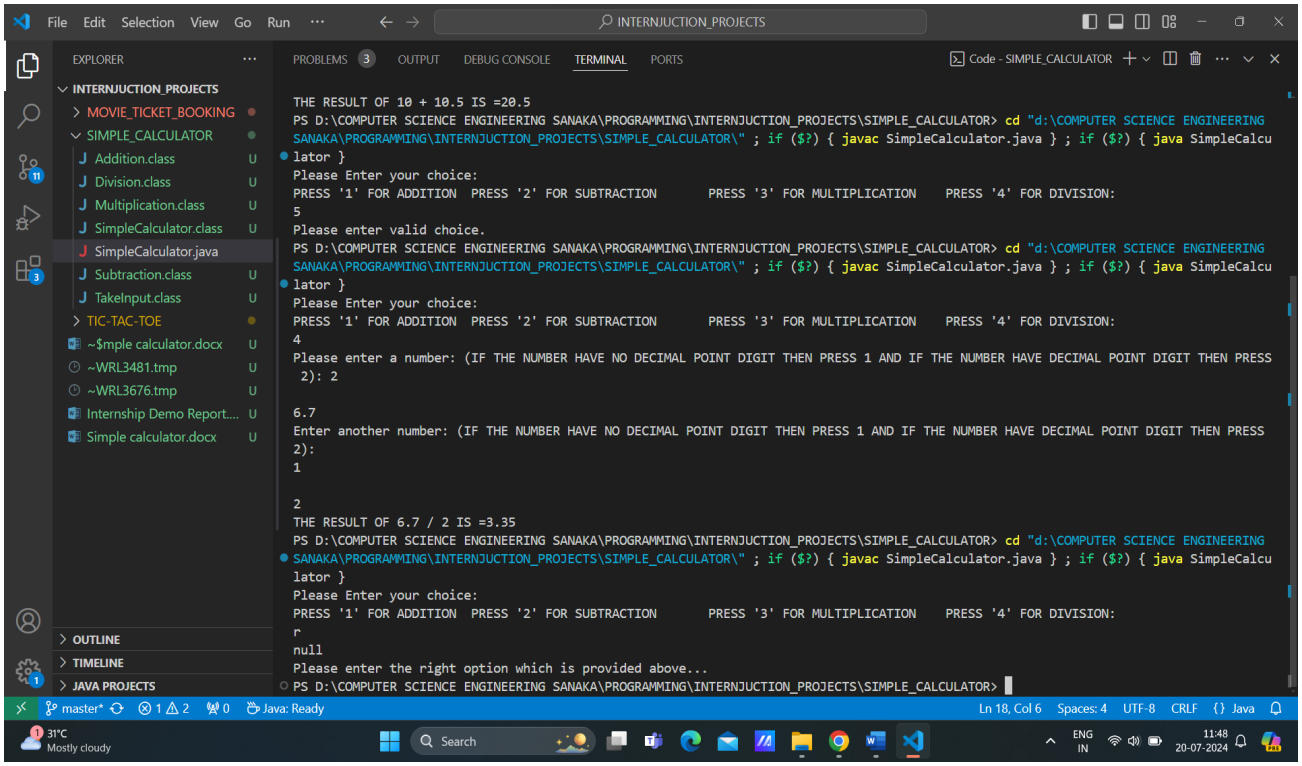
- After displaying the result, the program concludes its execution.

Summary

The design of the `Addition`, `Subtraction`, `Multiplication`, and `Division` classes effectively utilizes polymorphism to handle different types of numeric inputs seamlessly. The flow of the program—from user input to method execution and result display—is streamlined, ensuring a user-friendly experience. This approach not only simplifies the code structure but also enhances the flexibility and maintainability of the application.

RESULTS


File Edit Selection View Go Run
INTERNJUCTION_PROJECTS
EXPLORER
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - SIMPLE_CALCULATOR
INTERNJUCTION_PROJECTS
MOVIE_TICKET_BOOKING
SIMPLE_CALCULATOR
Addition.class
Division.class
Multiplication.class
SimpleCalculator.class
SimpleCalculator.java
Subtraction.class
TakeInput.class
TIC-TAC-TOE
~$mple calculator.docx
~WRL3481.tmp
~WRL3676.tmp
Internship Demo Report....
Simple calculator.docx
OUTLINE
TIMELINE
JAVA PROJECTS
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
1
Please enter a number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS 2): 1
10
Enter another number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS 2):
2
10.5
THE RESULT OF 10 + 10.5 IS =20.5
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
5
Please enter valid choice.
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
4
Please enter a number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS 2): 2
6.7
Enter another number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS
master* 1 2 0 Java: Ready
Ln 18, Col 6 Spaces: 4 UTF-8 CRLF Java
31°C Mostly cloudy
Search
ENG IN
11:47 20-07-2024


File Edit Selection View Go Run
INTERNJUCTION_PROJECTS
EXPLORER
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - SIMPLE_CALCULATOR
INTERNJUCTION_PROJECTS
MOVIE_TICKET_BOOKING
SIMPLE_CALCULATOR
Addition.class
Division.class
Multiplication.class
SimpleCalculator.class
SimpleCalculator.java
Subtraction.class
TakeInput.class
TIC-TAC-TOE
~$mple calculator.docx
~WRL3481.tmp
~WRL3676.tmp
Internship Demo Report....
Simple calculator.docx
OUTLINE
TIMELINE
JAVA PROJECTS
THE RESULT OF 10 + 10.5 IS =20.5
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
5
Please enter valid choice.
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
4
Please enter a number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS 2): 2
6.7
Enter another number: (IF THE NUMBER HAVE NO DECIMAL POINT DIGIT THEN PRESS 1 AND IF THE NUMBER HAVE DECIMAL POINT DIGIT THEN PRESS 2):
1
2
THE RESULT OF 6.7 / 2 IS =3.35
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR> cd "d:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR\" ; if ($?) { javac SimpleCalculator.java } ; if ($?) { java SimpleCalculator }
Please Enter your choice:
PRESS '1' FOR ADDITION PRESS '2' FOR SUBTRACTION PRESS '3' FOR MULTIPLICATION PRESS '4' FOR DIVISION:
r
null
Please enter the right option which is provided above...
PS D:\COMPUTER SCIENCE ENGINEERING SANAKA\PROGRAMMING\INTERNJUCTION_PROJECTS\SIMPLE_CALCULATOR>
master* 1 2 0 Java: Ready
Ln 18, Col 6 Spaces: 4 UTF-8 CRLF Java
31°C Mostly cloudy
Search
ENG IN
11:48 20-07-2024

CONCLUTION

Successful Completion of the Simple Calculator Project

The Simple Calculator (Console-Based) project provided by InternJunction has been successfully completed, marking a significant achievement in practical programming.

Practical Application of Knowledge

We applied theoretical knowledge effectively, implementing basic arithmetic operations and object-oriented programming concepts to develop a functional and user-friendly application.

Achievement of Objectives

The project met its primary objective of creating a user-friendly calculator and its educational goal of serving as a valuable learning resource for beginners in coding.

Skill Enhancement

Our programming skills were significantly enhanced through hands-on experience with Java, user input handling, error management, and polymorphism.

Positive Impact on Professional Growth

The project positively impacted our professional growth by improving problem-solving abilities, strengthening our understanding of OOP principles, and providing insights into industry-level software development.

Overall Experience

The project was enriching and fulfilling, solidifying our technical skills and boosting our confidence in real-world programming tasks. The knowledge and experience gained will contribute to our future success in software development.

We are grateful to InternJunction for this valuable opportunity and look forward to applying these skills in future endeavors.

FUTURE WORK

Advanced Technologies Integration

Graphical User Interface (GUI): Develop a GUI version of the calculator using frameworks like JavaFX or Swing for a more user-friendly experience.

Mobile Application: Create a mobile app version of the calculator for Android and iOS platforms.

New Applications Exploration

Scientific Calculations: Expand functionality to include scientific operations like trigonometric functions, logarithms, and exponentials.

Statistical Calculations: Add features for statistical analysis, including mean, median, mode, and standard deviation calculations.

Long-Term Evaluations

Performance Optimization: Conduct long-term evaluations to optimize performance and ensure the calculator remains efficient with larger datasets and more complex calculations.

User Feedback: Implement a system for collecting user feedback to continually improve the calculator's functionality and user experience.

These future work directions aim to enhance and expand the Simple Calculator project, ensuring its continued effectiveness and adaptability in various contexts.