*Submitted in partial fulfillment of requirements for the award of*

**Bachelors of Computer Applications**



*Submitted by:*

**ABHISHEK SHARMA (CS23BCAGN002)**

*Under the Supervision and Guidance of*

**NAWAB WASHIM RAHMAN**

**School of Computing Sciences**

**The Assam Kaziranga University, Jorhat, Assam**

# Greeting Function (funtion.py)

This simple Python script is designed to greet the user.

It contains a function called `greet\_user(name)` that takes a user's name as input. If a name is provided, it returns a personalized greeting. If not, it defaults to greeting the user as "Guest."

This kind of script is helpful in learning how to work with functions, conditionals, and string formatting in Python.

# Linear Equation Solver (solve\_linear\_equation.py)

This program solves linear equations of the form `ax + b = 0`.

It considers all edge cases:

* If `a` is zero and `b` is also zero, the solution is infinite.
* If `a` is zero but `b` is not, there's no solution.
* Otherwise, it calculates the value of `x` using the formula `x = -b/a`.

This project introduces the concept of conditional logic, mathematical operations, and return values in Python.

# Star Pattern and Line Graph (star and graph.py)

This file contains two separate functionalities:

1. \*\*Star Pattern Printing\*\*: It prints a right-angle triangle pattern made of asterisks (`\*`). This is a classic beginner exercise to understand loops.
2. \*\*Graph Plotting\*\*: It uses the `matplotlib` library to draw a simple line graph with x and y values. This part introduces basic data visualization in Python.

Combining both gives a creative twist: one part strengthens logical thinking, while the other introduces external libraries and visualization.

# GUI Greeting App (tinker\_greet.py)

This is another tkinter-based project that takes a user's name through an input field and greets them when a button is clicked.

If no name is entered, it defaults to greeting the user as "Guest."

This project is great for understanding GUI basics, event handling, and interacting with user input.

# 5.Arithmetic and Quadratic Operation(Arithmatic\_Quadratic.py)

From this program, you can learn:

1. **Basic Arithmetic Operations**: How to perform addition, subtraction, multiplication, and division in Python, including handling special cases like division by zero.
2. **Quadratic Equation Solving**: How to use the quadratic formula to calculate roots of a quadratic equation and determine if the roots are real or complex based on the discriminant.
3. **Using Functions**: How to organize code into reusable functions, making it cleaner and easier to understand.
4. **Error Handling**: How to use conditions (like checking if b=0b = 0b=0 or the discriminant is negative) to handle specific cases and avoid runtime errors.
5. **Modular Programming**: How to divide a program into smaller, independent parts for better organization and reusability.