ELE 503

Advanced Computer Programming and Statistics

Week#5: Revision of FORTRAN and BASIC Programming

By Kingsley E. Erhabor

Week 05.

Revision of FORTRAN and BASIC Programming

Introduction to FORTRAN and BASIC

FORTRAN Programming Language

History and Significance

Syntax and Structure

BASIC Programming Language

History and Significance

Syntax and Structure

Comparative Analysis of FORTRAN and BASIC

Applications of FORTRAN and BASIC in Engineering

Q&A

Closing Take away

Efosa's Introduction

Engineer | Programmer | Innovator

Technical Authority

Shell Nigeria

Subject Mater Expert (EMEA) for Process Automation & Control (PACO)-Subsea control systems and Subsea Distribution

Containerization 85 %

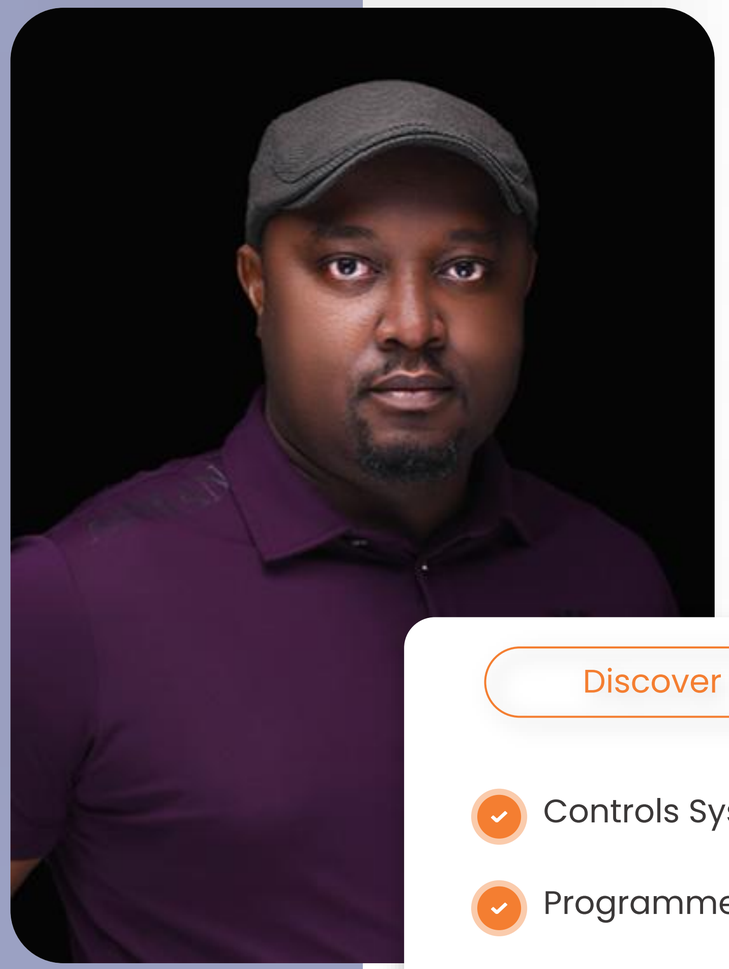
AI & ML 80 %

Renewable Energy 90 %

Innovator, VC

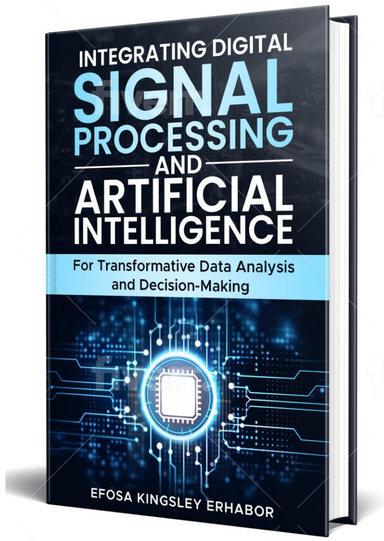
Katharos Technologies

Linux, Devops, AI and Software SME, Innovator and enterprenur



Discover

- Controls System
- Programmer
- Subsea Engineer
- AI and ML



X or Twitter

Linkedin

Learning Objectives

ELE 503: Advanced Computer Programming and Statistics

- **Review** the syntax and structure of FORTRAN and BASIC programming languages.

- **Write and execute** simple programs for numerical computations and simulations.

- **Understand** the historical significance and current applications of these languages in engineering.

Part 1:

Introduction to FORTRAN and BASIC

Importance of Revisiting FORTRAN and BASIC in Engineering

ELE 503: Advanced Computer Programming and Statistics

- **Foundational Knowledge:** Understanding the origins of modern programming languages.

- **Legacy Systems:** Many engineering applications still use or are influenced by FORTRAN and BASIC.

- **Problem-Solving Skills:** Reinforce programming fundamentals through these languages.

- **Appreciation of Evolution:** Recognize the advancements from FORTRAN and BASIC to current languages.

Part 2:

FORTRAN Programming Language

Introduction to FORTRAN

ELE 503: Advanced Computer Programming and Statistics

- **Full Form:** FORTRAN stands for "Formula Translation.“

- **History:**
  - Developed in the 1950s by IBM for scientific and engineering computations.
  - One of the earliest high-level programming languages.

- **Significance:**
  - Pioneered concepts like loops, conditionals, and subroutines.
  - Widely used in computational physics, engineering simulations, and numerical weather prediction

FORTRAN Syntax and Structure

ELE 503: Advanced Computer Programming and Statistics

- **Basic Structure:**
- **Program Statement:** Begins with PROGRAM and ends with END.

- **Variable Declaration:** Types like INTEGER, REAL, DOUBLE PRECISION, CHARACTER.

- **Control Structures:** IF, DO loops, GOTO statements.

- **Input/Output:** READ, WRITE statements.

- **Example: Simple FORTRAN Program Structure**

```
PROGRAM HelloWorld
    PRINT *, 'Hello, World!'
END PROGRAM HelloWorld
```

FORTRAN Sample Code and Explanation

ELE 503: Advanced Computer Programming and Statistics

- Example 1: Calculating the Area of a Circle

```
PROGRAM CircleArea
    IMPLICIT NONE
    REAL :: radius, area
    PRINT *, 'Enter the radius of the circle:'
    READ *, radius
    area = 3.14159 * radius**2
    PRINT *, 'The area of the circle is:', area
END PROGRAM CircleArea
```

Line-by-Line Explanation:

1.PROGRAM CircleArea
- Defines the start of the program named CircleArea.

2.IMPLICIT NONE
- Disables implicit typing, requiring all variables to be explicitly declared.

3.REAL :: radius, area
- Declares two real-number variables: radius and area.

4.PRINT *, 'Enter the radius of the circle:'
- Prompts the user to input the radius.

5.READ *, radius
- Reads the user input and assigns it to the variable radius.

6.area = 3.14159 * radius2**
- Calculates the area using the formula πr2\pi r^2πr2.

7.PRINT *, 'The area of the circle is:', area
- Outputs the calculated area.

8.END PROGRAM CircleArea
- Marks the end of the program.

Part 3:

BASIC Programming Language

Introduction to BASIC

ELE 503: Advanced Computer Programming and Statistics

- **Full Form:** BASIC stands for "Beginner's All-purpose Symbolic Instruction Code.“

- **History:**
  - Developed in the mid-1960s at Dartmouth College.
  - Designed to be an easy-to-learn language for beginners.

- **Significance:**
  - Widely used in early personal computers.
  - Popular in educational settings for teaching programming fundamentals

BASIC Syntax and Structure

ELE 503: Advanced Computer Programming and Statistics

- **Basic Structure:**

- **Line Numbers:** Traditional BASIC uses line numbers for program order and GOTO statements.

- **Variable Declaration:** Dynamic typing; variables are inferred from their names.

- **Control Structures:** IF...THEN, FOR...NEXT loops.

- **Input/Output:** INPUT, PRINT statements.

- **Example: Simple BASIC Program Structure**

```
10 PRINT "Hello, World!"
20 END
```

BASIC Sample Code and Explanation

ELE 503: Advanced Computer Programming and Statistics

- Example 1: Calculating the Area of a Circle

```
10 PRINT "Enter the radius of the circle:"
20 INPUT R
30 AREA = 3.14159 * R^2
40 PRINT "The area of the circle is: "; AREA
50 END
```

Line-by-Line Explanation:

1.10 PRINT "Enter the radius of the circle:"
- Displays a prompt for the user to input the radius.

2.20 INPUT R
- Takes user input and stores it in the variable R.

3.30 AREA = 3.14159 * R^2
- Calculates the area using the formula πr2 - pi r^2

4.40 PRINT "The area of the circle is: "; AREA
- Outputs the calculated area.

5.50 END
- Ends the program

Part 4:

Comparative Analysis of FORTRAN and BASIC

Comparative Analysis of FORTRAN and BASIC

ELE 503: Advanced Computer Programming and Statistics

Feature	FORTRAN	BASIC
Purpose	Scientific and engineering computations	Beginner-friendly programming
Syntax	Strict, requires explicit declarations	Flexible, often uses line numbers
Control Structures	`IF`, `DO`, `GOTO`	`IF...THEN`, `FOR...NEXT`, `GOTO`
Variable Typing	Explicit (e.g., `REAL`, `INTEGER`)	Implicit (based on variable names)
Use Cases	Numerical simulations, computational physics	Educational purposes, early personal computing
Convergence	Designed for high-performance computing	Designed for ease of use and learning

Applications of FORTRAN and BASIC in Engineering

ELE 503: Advanced Computer Programming and Statistics

- **FORTRAN:**
  - **Computational Fluid Dynamics (CFD):** Simulating fluid flows.
  - **Finite Element Analysis (FEA):** Structural analysis and stress testing.
  - **Weather Prediction Models:** Numerical weather forecasting.
  - **Aerospace Engineering:** Trajectory calculations and simulations.
- **BASIC:**
  - **Educational Tools:** Teaching programming basics to engineering students.
  - **Early Control Systems:** Simple embedded systems in machinery.
  - **Rapid Prototyping:** Quick development of simple simulations and calculations.
  - **Legacy Systems:** Maintaining older engineering software.

Writing and Executing Simple FORTRAN Programs

ELE 503: Advanced Computer Programming and Statistics

- **Choose an Integrated Development Environment (IDE) or Compiler:**
  - Examples: GNU Fortran (gfortran), Intel Fortran Compiler, Visual Studio with FORTRAN extensions.
- **Write the Program:**
  - Use a text editor or IDE to write your FORTRAN code.
- **Compile the Program:**
  - Use the compiler to convert the code into an executable.
  - Example Command: gfortran CircleArea.f90 -o CircleArea
- **Execute the Program:**
  - Run the compiled executable.
  - Example Command: ./CircleArea

Writing and Executing Simple BASIC Programs

ELE 503: Advanced Computer Programming and Statistics

Steps:

1.Choose a BASIC Interpreter or Compiler:

- Examples: QB64, FreeBASIC, Microsoft Small Basic.

2.Write the Program:

- Use a text editor or IDE to write your BASIC code.

3.Run the Program:

- Use the interpreter to execute the code directly or compile it.
- Example Command (QB64): Open QB64, load the program, and run.

Hands-On Exercise 1 - FORTRAN Programming

ELE 503: Advanced Computer Programming and Statistics

Task: Write a FORTRAN program to calculate the factorial of a number.

Instructions:

1.Define the Program Structure:

- Begin with PROGRAM Factorial.
- End with END PROGRAM Factorial

2.Declare Variables:

- Use INTEGER for the number and factorial.

3.Input the Number:

- Use PRINT and READ statements.

4.Calculate the Factorial:

- Use a DO loop.

5.Output the Result:

- Use PRINT statement.

- Sample Code

```
PROGRAM Factorial
   IMPLICIT NONE
   INTEGER :: N, i, fact

   PRINT *, 'Enter a positive integer:'
   READ *, N

   IF (N < 0) THEN
      PRINT *, 'Factorial is not defined for negative numbers.'
      STOP
   END IF

   fact = 1
   DO i = 1, N
      fact = fact * i
   END DO

   PRINT *, 'The factorial of ', N, ' is ', fact
END PROGRAM Factorial
```

Hands-On Exercise 2 – BASIC Programming

ELE 503: Advanced Computer Programming and Statistics

Task: Write a BASIC program to solve the quadratic equation ax2+bx+c=0 (ax^2 + bx + c = 0).

Instructions:

1.Prompt User for Coefficients:

- Use PRINT and INPUT statements.

2.Calculate Discriminant:

- D=b2−4acD = b^2 - 4acD=b2−4ac

3.Determine Nature of Roots:

- Based on the discriminant.

4.Calculate and Display Roots:

- Use IF...THEN statements.

- Sample Code

```
10 PRINT "Quadratic Equation Solver"
20 PRINT "Enter coefficient a:"
30 INPUT A
40 PRINT "Enter coefficient b:"
50 INPUT B
60 PRINT "Enter coefficient c:"
70 INPUT C
80 D = B^2 - 4*A*C
90 IF D > 0 THEN GOTO 110
100 IF D = 0 THEN GOTO 130
110 PRINT "Roots are real and distinct."
120 GOTO 160
130 PRINT "Roots are real and equal."
140 GOTO 160
160 IF D >= 0 THEN
170 X1 = (-B + SQR(D)) / (2*A)
180 X2 = (-B - SQR(D)) / (2*A)
190 PRINT "Root 1: "; X1
200 PRINT "Root 2: "; X2
210 ELSE
220 REAL_PART = -B / (2*A)
230 IMAG_PART = SQR(-D) / (2*A)
240 PRINT "Roots are complex."
250 PRINT "Root 1: "; REAL_PART; " + "; IMAG_PART; "i"
260 PRINT "Root 2: "; REAL_PART; " - "; IMAG_PART; "i"
270 END IF
```

Visualizing Convergence in FORTRAN and BASIC

ELE 503: Advanced Computer Programming and Statistics

- **Approach:**
  - Modify the factorial program to record each multiplication step.
  - Output the intermediate factorial values.
  - Use external tools (e.g., Excel, MATLAB) to plot the growth.

BASIC: Plotting Quadratic Roots Calculation

- **Approach:**
  - Record the discriminant and roots at each step.
  - Use BASIC's plotting capabilities or export data for external visualization.

Note: Modern visualization may require exporting data to external applications, as classic BASIC has limited built-in plotting functionalities.

Historical Significance and Modern Applications

ELE 503: Advanced Computer Programming and Statistics

FORTRAN:

- **Foundation for Scientific Computing:**
  - Influenced the development of other scientific programming languages like MATLAB and Julia.
- **Legacy Code:**
  - Extensive use in legacy systems within aerospace, automotive, and energy sectors.
- **High-Performance Computing:**
  - Optimized for numerical computations and parallel processing.

BASIC:

- **Educational Impact:**
  - Introduced thousands of students to programming.
- **Early Personal Computing:**
  - Integral in the development of early personal computers (e.g., Apple II, Commodore 64).
- **Scripting and Prototyping:**
  - Quick development of simple scripts and prototypes.

Comparative Code Explanation

ELE 503: Advanced Computer Programming and Statistics

FORTRAN vs. BASIC: Factorial Calculation

FORTRAN:

```
PROGRAM Factorial
   IMPLICIT NONE
   INTEGER :: N, i, fact

   PRINT *, 'Enter a positive integer:'
   READ *, N

   IF (N < 0) THEN
      PRINT *, 'Factorial is not defined for negative numbers.'
      STOP
   END IF

   fact = 1
   DO i = 1, N
      fact = fact * i
   END DO

   PRINT *, 'The factorial of ', N, ' is ', fact
END PROGRAM Factorial
```

BASIC:

```
10 PRINT "Factorial Calculator"
20 PRINT "Enter a positive integer:"
30 INPUT N
40 IF N < 0 THEN
   50 PRINT "Factorial is not defined for negative numbers."
   60 END
   70 fact = 1
   80 FOR i = 1 TO N
   90 fact = fact * i
   100 NEXT i
   110 PRINT "The factorial of "; N; " is "; fact
   120 END
```

Key Differences:

- **Line Numbers:** BASIC uses line numbers to denote the order, while FORTRAN uses structured blocks.
- **Variable Declaration:** FORTRAN requires explicit declarations (INTEGER), whereas BASIC infers types.
- **Control Structures:** FORTRAN uses DO...END DO, while BASIC uses FOR...NEXT.

Practical Considerations

ELE 503: Advanced Computer Programming and Statistics

- **Transition to Modern Languages:**
- Understanding FORTRAN and BASIC aids in learning structured and procedural programming concepts.

- **Interoperability:**
- Some modern systems still interface with legacy FORTRAN code for performance-critical tasks.

- **Preservation of Knowledge:**
- Maintaining familiarity with these languages is essential for maintaining and upgrading older engineering systems

Hands-On Exercise 3 – Writing Simple Programs

ELE 503: Advanced Computer Programming and Statistics

•Task: Write a FORTRAN program to solve a system of linear equations using Gaussian elimination.

Instructions:

1.Define the Program Structure:

- •Begin with PROGRAM GaussianElimination.
- •End with END PROGRAM GaussianElimination.

2.Declare Variables:

- •Use arrays to represent the matrix and vectors.

3.Input the System:

- •Allow user input for matrix coefficients and constants.

4.Implement Gaussian Elimination:

- •Perform row operations to reduce the matrix to upper triangular form.

5.Back Substitution:

- •Solve for variables starting from the last equation.

6.Output the Results:

- •Display the solution vector.

•PROGRAM GaussianElimination

```
  IMPLICIT NONE
  INTEGER, PARAMETER :: N = 3
  REAL :: A(N,N+1)
  INTEGER :: i, j, k
  REAL :: factor, sum
  REAL :: x(N)

  PRINT *, 'Enter the augmented matrix coefficients (3x
+ y + z = ...):'
DO i = 1, N
  PRINT *, 'Row ', i, ':'
  DO j = 1, N+1
    READ *, A(i,j)
  END DO
END DO

! Forward Elimination
DO k = 1, N-1
  IF (A(k,k) == 0.0) THEN
    PRINT *, 'Zero pivot encountered.'
    STOP
  END IF
  DO i = k+1, N
    factor = A(i,k) / A(k,k)
    DO j = k, N+1
      A(i,j) = A(i,j) - factor * A(k,j)
    END DO
  END DO
END DO

! Back Substitution
DO i = N, 1, -1
  sum = 0.0
  DO j = i+1, N
    sum = sum + A(i,j) * x(j)
  END DO
  x(i) = (A(i,N+1) - sum) / A(i,i)
END DO

PRINT *, 'Solution:'
DO i = 1, N
  PRINT *, 'x(', i, ') = ', x(i)
END DO
END PROGRAM GaussianElimination
```

Hands-On Exercise 4 - Writing Simple Programs in BASIC

ELE 503: Advanced Computer Programming and Statistics

Task: Write a BASIC program to compute the roots of a quadratic equation using the quadratic formula.

Instructions:

1.Prompt User for Coefficients:
- Use PRINT and INPUT statements.

2.Calculate Discriminant:
- D=b2−4acD = b^2 - 4acD=b2−4ac

3.Compute Roots Based on Discriminant:
- Real and distinct, real and equal, or complex.

4.Display Results:
- Use PRINT statements.

```
10 PRINT "Quadratic Equation Solver"
20 PRINT "Enter coefficient a:"
30 INPUT A
40 PRINT "Enter coefficient b:"
50 INPUT B
60 PRINT "Enter coefficient c:"
70 INPUT C
80 D = B^2 - 4*A*C
90 IF D > 0 THEN GOTO 110
100 IF D = 0 THEN GOTO 130
110 PRINT "Roots are real and distinct."
120 GOTO 160
130 PRINT "Roots are real and equal."
140 GOTO 160
160 IF D >= 0 THEN
170 X1 = (-B + SQR(D)) / (2*A)
180 X2 = (-B - SQR(D)) / (2*A)
190 PRINT "Root 1: "; X1
200 PRINT "Root 2: "; X2
210 ELSE
220 REAL_PART = -B / (2*A)
230 IMAG_PART = SQR(-D) / (2*A)
240 PRINT "Roots are complex."
250 PRINT "Root 1: "; REAL_PART; " + "; IMAG_PART; "i"
260 PRINT "Root 2: "; REAL_PART; " - "; IMAG_PART; "i"
270 END IF
```

Coding Best Practices in FORTRAN and BASIC

ELE 503: Advanced Computer Programming and Statistics

- **FORTRAN:**
  - **Use IMPLICIT NONE:** Enforces explicit variable declarations, reducing errors.
  - **Commenting:** Use ! for single-line comments.
  - **Modular Programming:** Utilize subroutines and functions for code reusability.
  - **Formatting:** Maintain readable code with proper indentation.

BASIC:

- **Avoid Excessive GOTO Statements:** Encourage structured programming practices.
- **Clear Variable Naming:** Use descriptive names for better code readability.
- **Commenting:** Use REM or ' for comments.
- **Consistent Indentation:** Enhances code clarity.

Understanding the Evolution from FORTRAN and BASIC to Modern Languages

ELE 503: Advanced Computer Programming and Statistics

- **Structured Programming:**
  - Transition from line-number-based BASIC to structured BASIC variants.
  - FORTRAN evolved to support modular and object-oriented programming.
- **Advanced Features:**
  - Introduction of complex data structures, error handling, and object-oriented paradigms in modern languages.
  - Enhanced performance optimizations and parallel computing capabilities.
- **Integration and Interoperability:**
  - Modern languages integrate with various APIs, databases, and web technologies.
  - Legacy FORTRAN and BASIC code can interface with modern systems through APIs and wrappers.
- **Educational Impact:**
  - Foundations laid by FORTRAN and BASIC continue to influence programming pedagogy and language design.

Historical Impact of FORTRAN and BASIC

ELE 503: Advanced Computer Programming and Statistics

FORTRAN:

- **First High-Level Language:** Enabled programmers to write complex scientific computations more efficiently.
- **Influence on Other Languages:** Inspired the development of languages like C, C++, and Python.
- **Standardization:** Promoted the creation of standardized coding practices and language specifications.

BASIC:

- **Democratizing Programming:** Made programming accessible to non-experts and beginners.
- **Personal Computing Revolution:** Played a pivotal role in the early personal computer era.
- **Educational Tool:** Continues to be used for teaching fundamental programming concepts.

Example Slide 1 – FORTRAN Factorial Program Explained

ELE 503: Advanced Computer Programming and Statistics

- Code Snippet: Calculating Factorial in FORTRAN

```
PROGRAM Factorial
    IMPLICIT NONE
    INTEGER :: N, i, fact

    PRINT *, 'Enter a positive integer:'
    READ *, N

    IF (N < 0) THEN
        PRINT *, 'Factorial is not defined for negative numbers.'
        STOP
    END IF

    fact = 1
    DO i = 1, N
        fact = fact * i
    END DO

    PRINT *, 'The factorial of ', N, ' is ', fact
END PROGRAM Factorial
```

Explanation:

1.PROGRAM Factorial
- Starts the program named Factorial.

2.IMPLICIT NONE
- Disables implicit variable typing, enforcing explicit declarations.

3.INTEGER :: N, i, fact
- Declares three integer variables: N (input number), i (loop counter), and fact (factorial result).

4.PRINT *, 'Enter a positive integer:'
- Prompts the user to enter a positive integer.

5.READ *, N
- Reads the user input and assigns it to variable N.

6.IF (N < 0) THEN ... END IF
- Checks if the input is negative and handles it by displaying a message and stopping the program.

7.fact = 1
- Initializes the factorial result to 1.

8.DO i = 1, N
- Begins a loop from 1 to N.

9.fact = fact * i
- Multiplies the current value of fact by i in each iteration.

10.END DO
- Ends the loop.

11.PRINT *, 'The factorial of ', N, ' is ', fact
- Outputs the calculated factorial.

12.END PROGRAM Factorial
- Marks the end of the program.

Example Slide 2 – BASIC Quadratic Solver Explained

ELE 503: Advanced Computer Programming and Statistics

- Code Snippet: Solving Quadratic Equations in BASIC

```
10 PRINT "Quadratic Equation Solver"
20 PRINT "Enter coefficient a:"
30 INPUT A
40 PRINT "Enter coefficient b:"
50 INPUT B
60 PRINT "Enter coefficient c:"
70 INPUT C
80 D = B^2 - 4*A*C
90 IF D > 0 THEN GOTO 110
100 IF D = 0 THEN GOTO 130
110 PRINT "Roots are real and distinct."
120 GOTO 160
130 PRINT "Roots are real and equal."
140 GOTO 160
160 IF D >= 0 THEN
170 X1 = (-B + SQR(D)) / (2*A)
180 X2 = (-B - SQR(D)) / (2*A)
190 PRINT "Root 1: "; X1
200 PRINT "Root 2: "; X2
210 ELSE
220 REAL_PART = -B / (2*A)
230 IMAG_PART = SQR(-D) / (2*A)
240 PRINT "Roots are complex."
250 PRINT "Root 1: "; REAL_PART; " + "; IMAG_PART; "i"
260 PRINT "Root 2: "; REAL_PART; " - "; IMAG_PART; "i"
270 END IF
```

Explanation:

1.10 PRINT "Quadratic Equation Solver"
- Displays the program title.

2.20 PRINT "Enter coefficient a:"
- Prompts the user for coefficient a.

3.30 INPUT A
- Reads user input for a.

4.40 PRINT "Enter coefficient b:"
- Prompts the user for coefficient b.

5.50 INPUT B
- Reads user input for b.

6.60 PRINT "Enter coefficient c:"
- Prompts the user for coefficient c.

7.70 INPUT C
- Reads user input for c.

8.80 D = B^2 - 4*A*C
- Calculates the discriminant DDD.

9.90 IF D > 0 THEN GOTO 110
- Checks if discriminant is positive for real and distinct roots.

10.100 IF D = 0 THEN GOTO 130
- Checks if discriminant is zero for real and equal roots.

11.110 PRINT "Roots are real and distinct."
- Informs user about the nature of roots.

12.120 GOTO 160
- Jumps to root calculation.

13.130 PRINT "Roots are real and equal."
- Informs user about the nature of roots.

14.140 GOTO 160
- Jumps to root calculation.

15.160 IF D >= 0 THEN
- Checks if roots are real.

16.170 X1 = (-B + SQR(D)) / (2*A)
- Calculates the first root.

17.180 X2 = (-B - SQR(D)) / (2*A)
- Calculates the second root.

18.190 PRINT "Root 1: "; X1

Example Slide 2 – BASIC Quadratic Solver Explained

ELE 503: Advanced Computer Programming and Statistics

- Code Snippet: Solving Quadratic Equations in BASIC

```
10 PRINT "Quadratic Equation Solver"
20 PRINT "Enter coefficient a:"
30 INPUT A
40 PRINT "Enter coefficient b:"
50 INPUT B
60 PRINT "Enter coefficient c:"
70 INPUT C
80 D = B^2 - 4*A*C
90 IF D > 0 THEN GOTO 110
100 IF D = 0 THEN GOTO 130
110 PRINT "Roots are real and distinct."
120 GOTO 160
130 PRINT "Roots are real and equal."
140 GOTO 160
160 IF D >= 0 THEN
170 X1 = (-B + SQR(D)) / (2*A)
180 X2 = (-B - SQR(D)) / (2*A)
190 PRINT "Root 1: "; X1
200 PRINT "Root 2: "; X2
210 ELSE
220 REAL_PART = -B / (2*A)
230 IMAG_PART = SQR(-D) / (2*A)
240 PRINT "Roots are complex."
250 PRINT "Root 1: "; REAL_PART; " + "; IMAG_PART; "i"
260 PRINT "Root 2: "; REAL_PART; " - "; IMAG_PART; "i"
270 END IF
```

Explanation:

- Displays the first root.

19.200 PRINT "Root 2: "; X2

- Displays the second root.

20.210 ELSE

- Handles complex roots.

21.220 REAL_PART = -B / (2*A)

- Calculates the real part of the roots.

22.230 IMAG_PART = SQR(-D) / (2*A)

- Calculates the imaginary part of the roots.

23.240 PRINT "Roots are complex."

- Informs user about complex roots.

24.250 PRINT "Root 1: "; REAL_PART; " + "; IMAG_PART; "i"

- Displays the first complex root.

25.260 PRINT "Root 2: "; REAL_PART; " - "; IMAG_PART; "i"

- Displays the second complex root.

26.270 END IF

- Ends the conditional statement.

Case Study 1 - FORTRAN in Aerospace Engineering

ELE 503: Advanced Computer Programming and Statistics

- **Problem:** Simulate the trajectory of a spacecraft considering gravitational forces.

- **Approach:**
  - Model the equations of motion using differential equations.
  - Implement a numerical solver (e.g., Runge-Kutta) in FORTRAN.

- **Outcome:** Accurate trajectory simulations aiding mission planning and control.

Case Study 2 – BASIC in Early Robotics

ELE 503: Advanced Computer Programming and Statistics

- **Problem:** Control the movement of a simple robot using BASIC programming.

- **Approach:**
  - Write BASIC scripts to handle sensor inputs and motor outputs.
  - Implement basic decision-making algorithms.

- **Outcome:** Demonstrated the feasibility of programmable robotics, influencing modern robotic programming languages.

Summary of Key Concepts

ELE 503: Advanced Computer Programming and Statistics

- **FORTRAN and BASIC:** Fundamental programming languages with significant historical impact.

- **Syntax and Structure:** Reviewed the core syntax, control structures, and program flow of both languages.

- **Practical Programming:** Developed and executed simple programs for numerical computations and simulations.

- **Historical Significance:** Appreciated the role of FORTRAN and BASIC in shaping modern programming paradigms.

- **Modern Applications:** Recognized the continued relevance of these languages in certain engineering domains and legacy systems.

Importance of Revisiting FORTRAN and BASIC

ELE 503: Advanced Computer Programming and Statistics

- **Foundation for Learning:** Reinforces basic programming concepts applicable to modern languages.

- **Legacy Systems Maintenance:** Essential for maintaining and upgrading existing engineering applications.

- **Problem-Solving Skills:** Enhances logical thinking and algorithm development abilities.

- **Appreciation of Evolution:** Understand the progression from early languages to contemporary programming practices.

Q&A Session

ELE 503: Advanced Computer Programming and Statistics

- **Open Floor:** Address any questions or clarifications.

- **Discussion Points:**
  - Challenges faced during programming exercises.
  - Insights on the differences between FORTRAN and BASIC.
  - Thoughts on the evolution of programming languages.
  - Applications of FORTRAN and BASIC in current engineering projects.

Homework Assignment

ELE 503: Advanced Computer Programming and Statistics

1. **Programming Task:**
    1. **Objective:** Write and execute programs in FORTRAN and BASIC.
    2. **Tasks:**
        1. **FORTRAN:**
            1. Implement a program to perform matrix multiplication.
            2. Ensure the program can handle matrices of size up to 10x10.
            3. Test the program with sample matrices and report the results.

        2. BASIC:
            1. Create a program to simulate simple harmonic motion.
            2. Use user inputs for amplitude, frequency, and phase.
            3. Output the position of the oscillator at discrete time intervals.

2. **Data Analysis Project:**

    1. **Objective:** Utilize FORTRAN and BASIC to solve engineering problems.
    2. **Tasks:**
        1. **FORTRAN:**
            1. Develop a program to compute the roots of a cubic equation using the Newton-Raphson method.
            2. Include error handling for cases with no real roots.

        2. BASIC:
            1. Design a program to calculate the electrical power consumption of a circuit based on user inputs for voltage and current.
            2. Extend the program to compute energy consumption over time.

Homework Assignment

ELE 503: Advanced Computer Programming and Statistics

3. **Historical Reflection:**
   1. **Objective:** Understand the evolution and impact of FORTRAN and BASIC.
   2. **Tasks:**
      1. Write a short essay (300-500 words) on how FORTRAN and BASIC have influenced modern programming languages.
      2. Discuss specific features from these languages that are present in today's languages.
      3. Reflect on the importance of preserving knowledge of legacy programming languages in the engineering field.