National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Programming Fundamentals

History and Programming Fundamentals

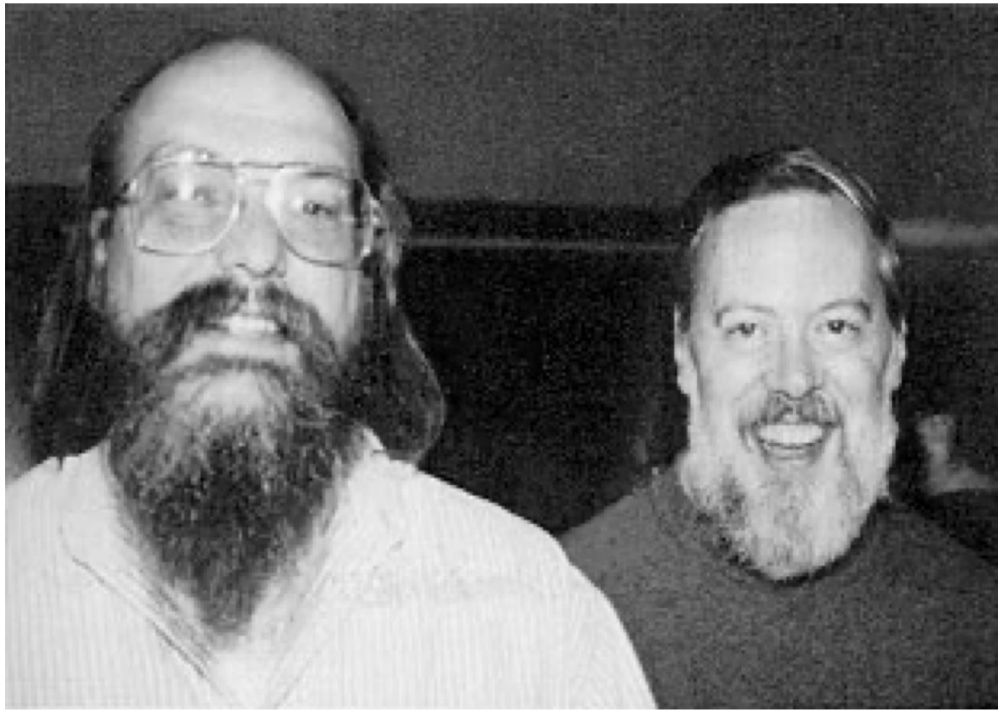
Bilal Khalid Dar

bilal.khalid@nu.edu.pk

BS(SE) FALL-2025

C++ History: Invention of C

- K. Thompson and D. Ritchie created C in 1972
- Strengths: C made it easy to write code that was
  - Fast
  - Simple
  - Cross-platform
- Limitations:
  - No objects or classes
  - Difficult to write code that worked generically
  - Tedious when writing large programs



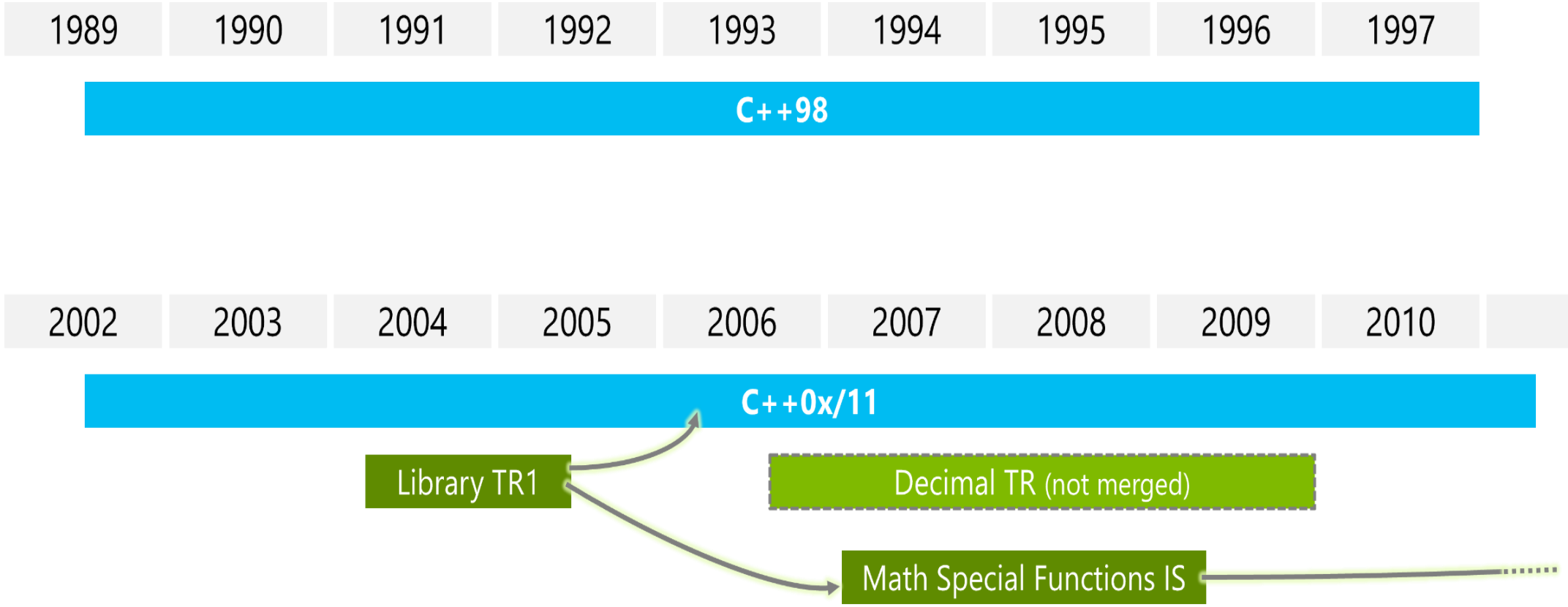
Ken Thompson and Dennis Ritchie, creators of the C language

C++ History: Welcome to C++!

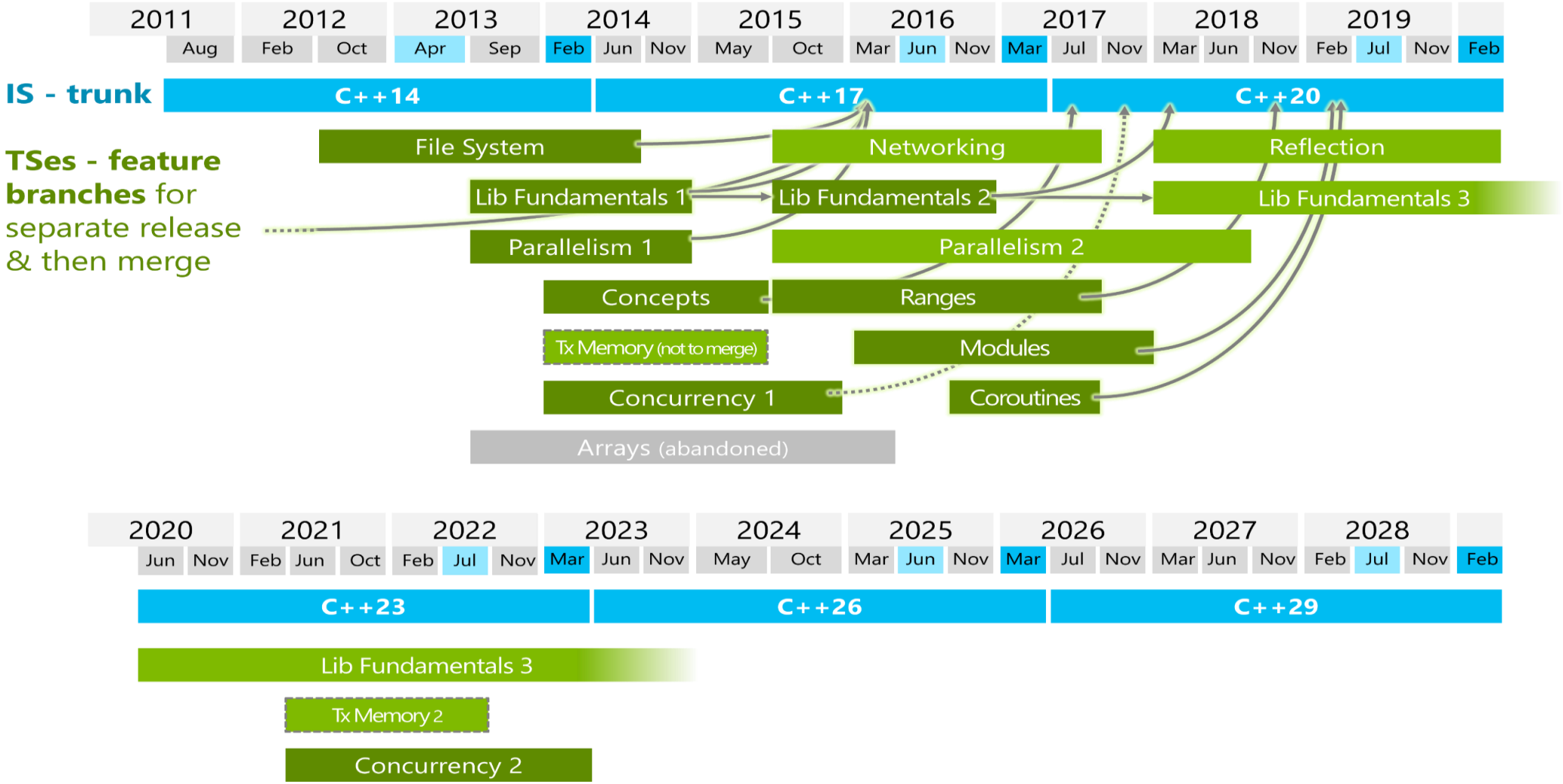
- In 1983, the first vestiges of C++ were created by Bjarne Stroustrup.
- He wanted a language that was:
  - Fast
  - Simple to Use
  - Cross-platform (Portability)
  - Had high level features

Bjarne Stroustrup

C++ History: Welcome to C++!



C++ History: Welcome to C++!



Problem Solving with Pseudo Code and Flowcharts

Problem solving

- Pseudocode:
  - Step by step written outline of your code
  - Pseudocode is a cross between human language and a programming language
  - Computer cannot understand pseudocode
  - programmers often find pseudocode helpful to write an algorithm in a language that's "almost" a programming language, but still very similar to natural language
- Flowchart
  - A flowchart is a diagram that shows the logical flow of a program

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

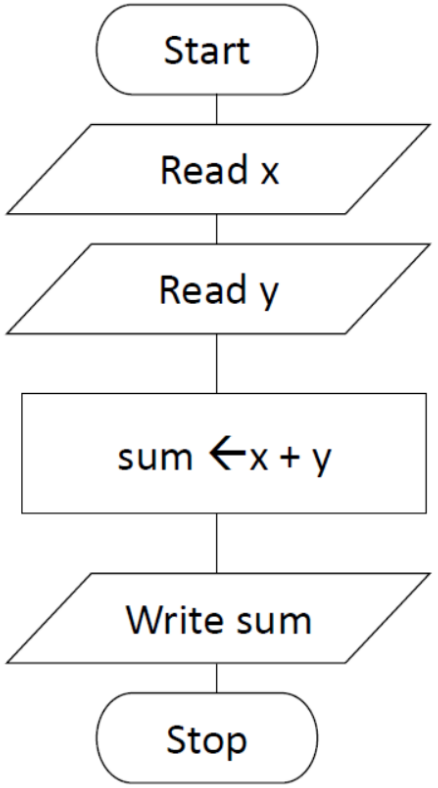
Pseudocode

- *pseudo* means fake, so *pseudocode* is fake code.
- A pseudocode:
  - Is an informal language
  - Has no syntax rules
  - Is not meant to be compiled or executed
  - Is used to create models, or “mock-ups” of programs
  - Can help in focusing all attention on the program’s design
  - Is easier to be translated to actual code

Pseudocode and flowchart (An example)

- Problem: an example pseudocode to design a solution for a program that adds two numbers
    1. Get the first number
    2. Get the second number
    3. Add the two numbers
    4. Display the result
- Program Development:
    - Understand the problem
        1. What do I know? (two numbers e.g., x and y)
        2. What do I have to do? (sum of x and y)
        3. How do I get from (A) to (B)? (sum = x + y)
    - Develop a solution using either flow-charts or pseudo-code
    - Write the program
    - Test the program

A flow-chart:



Designing a Program

- Problem: How to calculate and display the gross pay for an hourly paid employee?
- Steps:

1. Get the number of hours worked.
2. Get the hourly pay rate.
3. Multiply the number of hours worked by the hourly pay rate.
4. Display the result of the calculation that was performed in Step 3.

Example of a formal Pseudocode

- Find the Largest of Two Numbers

```
START
   INPUT number1
   INPUT number2
   IF number1 > number2 THEN
        PRINT "Number1 is larger"
   ELSE
        PRINT "Number2 is larger"
   ENDIF
END
```

Rules of Writing Pseudocode

- Use **simple English**.
- Write **one instruction per line**.
- Use keywords like:
- START, END
- INPUT, OUTPUT
- IF, ELSE, ENDIF
- FOR, WHILE, ENDLOOP
- Indent to show structure.
- Be consistent with formatting

Common Keywords

- **START / END** → Begin and end pseudocode.
- **INPUT / OUTPUT** → Take input, display result.
- **SET /** ← → Assign values.
- **IF / ELSE / ENDIF** → Decision making.
- **FOR / WHILE / ENDLOOP** → Repetition.

Example 1 - Calculate Area of a Rectangle

- START
- INPUT length
- INPUT width
- SET area = length * width
- PRINT "Area = ", area
- END

Pseudocode

- Problem: How to calculate and display the gross pay for an hourly paid employee?

```
Display "Enter the number of hours the employee worked."
Input hours
Display "Enter the employee's hourly pay rate."
Input payRate
Set grossPay = hours * payRate
Display "The employee's gross pay is $", grossPay
```

- Note the display, input, set, etc. keywords that represent a specific type of instruction.

Flowchart

- A flowchart is a diagram that graphically depicts the steps that take place in a program.
- There are four types of symbols in a flowchart:
  - **Ovals** (terminal symbols)

    The Start and End terminal symbols mark the program's starting and ending points respectively
  - **Parallelograms**

    Input and output symbols
  - **Rectangles** (processing symbols)

    Represent a step in the program.
  - **Diamonds** (decision symbols)

    We will discuss them later…
  - The symbols are connected by arrows

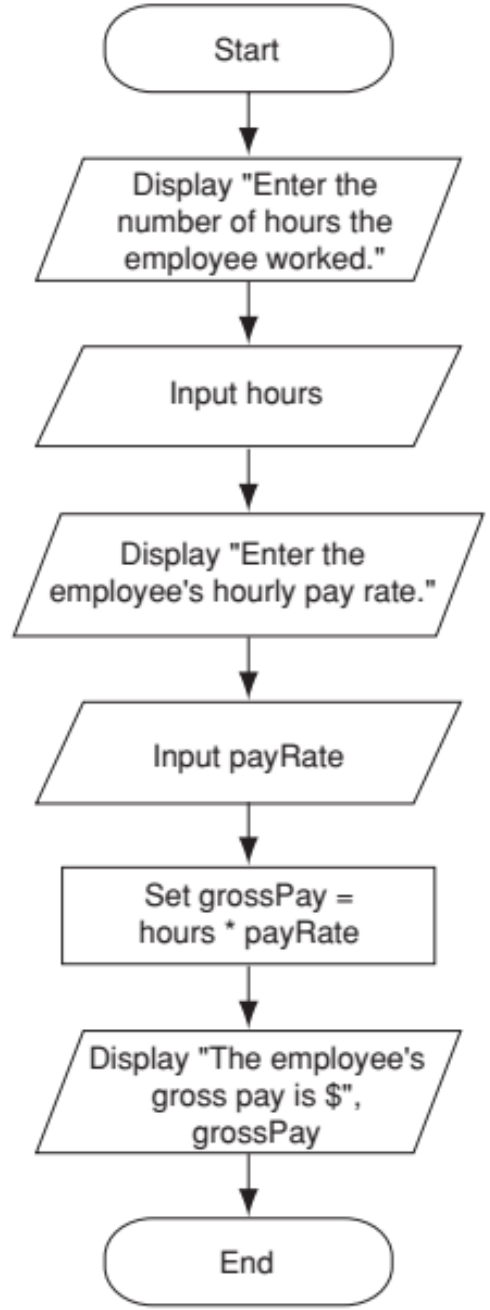
    Arrows represent the "flow" of the program. To step through the symbols in the proper order, you begin at the Start terminal and follow the arrows until you reach the End terminal.

Flowchart Example

- Problem: How to calculate and display the gross pay for an hourly paid employee?

```
Display "Enter the number of hours the employee worked."
Input hours
Display "Enter the employee's hourly pay rate."
Input payRate
Set grossPay = hours * payRate
Display "The employee's gross pay is $", grossPay
```

- Note that the flowchart represents the logic of the pseudocode.



Problem 1

- Check if a Number is Even or Odd

Problem 1

- Check if a Number is Even or Odd

```
START
  INPUT number
  IF number MOD 2 = 0 THEN
      PRINT "Even"
  ELSE
      PRINT "Odd"
  ENDIF
END
```

Problem 2

- Write a program that takes in a number from a user and tells them if it is Divisible by 11
- Input = User Number
- Output = Printed statement on divisibility
- Hint: Mod (%) gives the remainder
- 5 Mod 2 = 1

Pseudo Code Solution

Take a Number as input from user

If Number Mod 11 is 0

    Print Affirmative Statement

Else

    Print Negative Statement

Problem 3

- Ask a student for 5 subject marks, Calculate its average and display it.

- Validate that all numbers are between 0 and 100

New Office

C503G

Solution 1

```
START
    PROMPT "Enter mark1"
    READ mark1
    PROMPT "Enter mark2"
    READ mark2
    PROMPT "Enter mark3"
    READ mark3
    PROMPT "Enter mark4"
    READ mark4
    PROMPT "Enter mark5"
    READ mark5
    IF mark1 < 0 OR mark1 > 100 THEN
        PRINT "Error: mark1 must be between 0 and 100"
    ELSE IF mark2 < 0 OR mark2 > 100 THEN
        PRINT "Error: mark2 must be between 0 and 100"
    ELSE IF mark3 < 0 OR mark3 > 100 THEN
        PRINT "Error: mark3 must be between 0 and 100"
    ELSE IF mark4 < 0 OR mark4 > 100 THEN
        PRINT "Error: mark4 must be between 0 and 100"
    ELSE IF mark5 < 0 OR mark5 > 100 THEN
        PRINT "Error: mark5 must be between 0 and 100"
    ELSE
        total = mark1 + mark2 + mark3 + mark4 + mark5
        average = total / 5
        PRINT "Average marks = ", average
    ENDIF
END
```

Solution 2

```
START
   PROMPT "Enter mark1"
   READ mark1

   PROMPT "Enter mark2"
   READ mark2

   PROMPT "Enter mark3"
   READ mark3

   PROMPT "Enter mark4"
   READ mark4

   PROMPT "Enter mark5"
   READ mark5

 IF (mark1 < 0 OR mark1 > 100) OR
      (mark2 < 0 OR mark2 > 100) OR
      (mark3 < 0 OR mark3 > 100) OR
      (mark4 < 0 OR mark4 > 100) OR
      (mark5 < 0 OR mark5 > 100) THEN
         PRINT "Error: All marks must be between 0
and 100"
   ELSE
         total = mark1 + mark2 + mark3 + mark4 +
mark5
         average = total / 5
         PRINT "Average marks = ", average
   ENDIF
END
```

Problem 4

- Check if a number is positive, negative, or zero.

Problem: Print numbers from 1 to 10

```
START
    SET i = 1
    WHILE i <= 10 DO
        PRINT i
        i = i + 1
    ENDWHILE
END
```

Display multiplication table of a number.

```
START
    Display "Enter a number"
    READ num
    SET i = 1
    WHILE i <= 10 DO
        result = num * i
        PRINT num, " x ", i, " = ", result
        i = i + 1
    ENDWHILE / ENDLOOP
END
```

Sum of first 10 natural numbers

```
START
    SET i = 1
    SET sum = 0

    WHILE i <= 10 DO
        sum = sum + i
        i = i + 1
    ENDWHILE

    PRINT "Sum = ", sum
END
```

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Programming Fundamentals

First C++ Code

Bilal Khalid Dar

bilal.khalid@nu.edu.pk

BS(SE) FALL-2025

First Code in C++

What is a Program Made of?

Three steps that a program typically performs

- Input: (gather input data)
  - from keyboard
  - from files on disk drives
- Processing
  - Process the input data
- Output
  - Display the results as output
  - Send it to the screen
  - Write to a file

Structure of a simple C++ program

```
// sample C++ program
//……..
#include <iostream>
using namespace std;
……
int main()
{   ….
        cout<< "Hello, World!";
        return 0;
}
```

Comments

Pre-processor directives

Which namespace to use

Any global declarations

Beginning of function named main function

Start of the block main

Any local declarations

Statement

Send 0 to operating system

End of the block for main

Other functions