

# COSC1101- Programming Fundamentals

Maham Khan

Lecture - 3

# Problem Solving and Implementation

A programming task can be divided into two phases:

1. ***Problem solving***

***Define*** : Clearly describe a problem

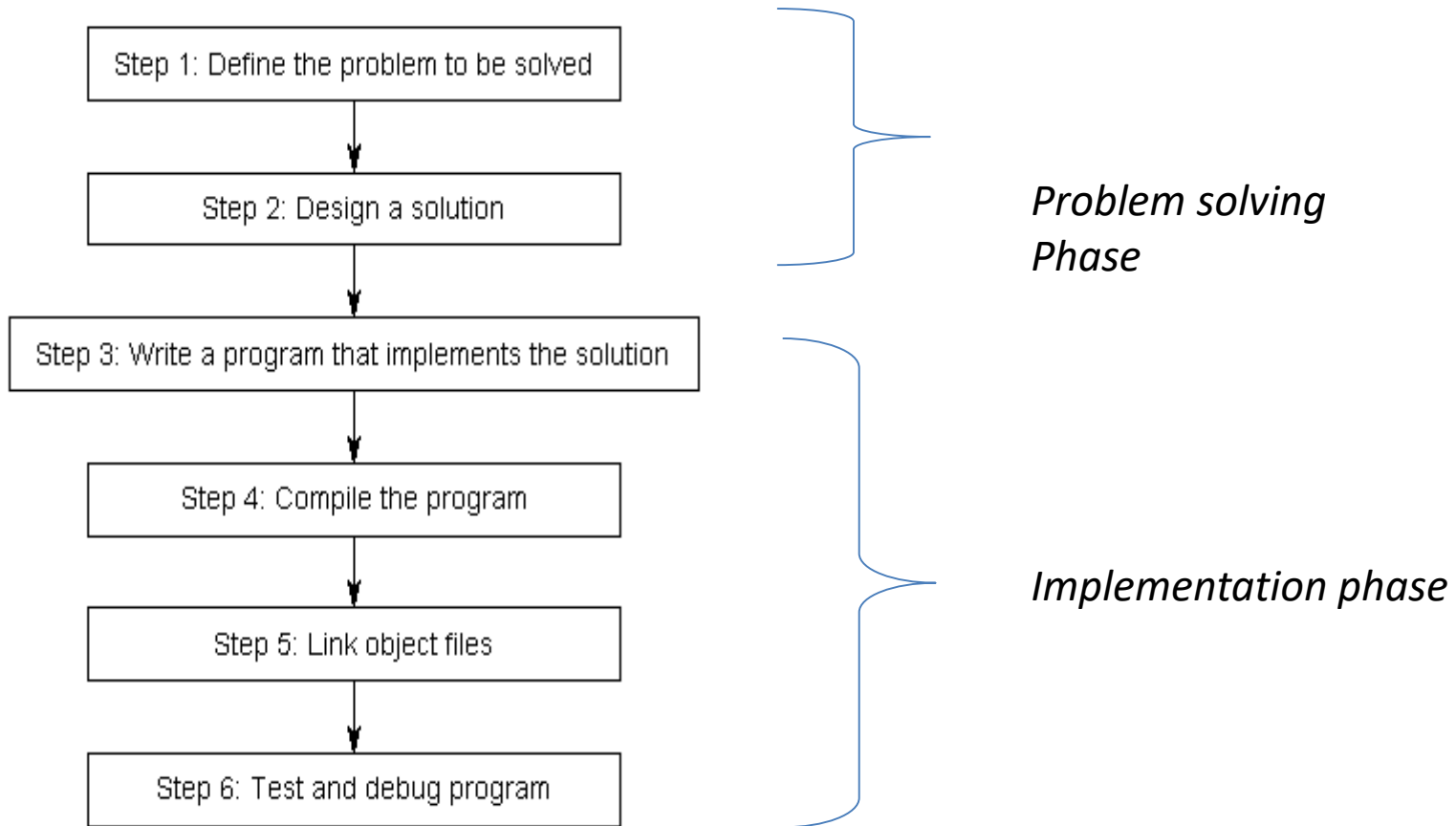
***Design its solution***: Produce an ordered sequence of steps that describe solution to the problem;

2. ***Implementation phase***

Implement the program in some programming language

***write code, compile, link, Test & Debug***

# Developing a program



# Defining a Problem:

Break the definition of the problem down into manageable steps. Example; input, Processing; Output

Input ;                Read the temperature from keyboard

Processing;        Test the Temperature below or above  
freezing

Output;             Display the result on Screen

Users:                Identify the users.

Feasibility & Implementation.

# Design the solution

## Algorithm

A sequence of language independent steps which may be followed to solve a problem. An Algorithm can be developed with a:

- Pseudo Code
- Flowchart

Preferably using control Structures.

# Algorithm

## Pseudo code

- Pseudo code is a method of designing a program using English like statement to describe the logic and processing flow.
- There are no real rules; organizations follow their own standards. Conveniently understood and exchanged between IT professionals.

# Defining a Problem:

Break the definition of the problem down into manageable steps; Input, Processing; Output

Example -1:

Read in the temperature. If the temperature is less than 32 indicate below freezing on the screen. Else if the temperature is above freezing then indicate the same on the monitor screen.

Divide the above problem into manageable parts.

Input ;	Read the temperature from keyboard
Processing;	Test the Temperature below or above freezing
Output;	Display the result on Screen

# Algorithm

## Pseudo code

- Example -1:  
    Read the Temp  
    if (Temp < 32) then  
        Print "BELOW FREEZING"  
    else  
        Print "ABOVE FREEZING"  
    endif



# Algorithm

## Flowchart

- Diagrammatic or Graphical representations of steps for solving the given problem.
- Use standard symbols developed by ANSI (American National Standard Institute)

# Building Blocks of Flowchart

- Start and End Symbols



- Connector



- Arrows (indicate flow of control)



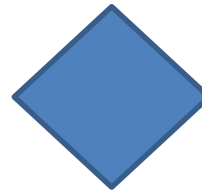
- Processing Steps



- Input/Output



- Decision



# Defining a Problem:

Break the definition of the problem down into manageable steps; Input, Processing; Output

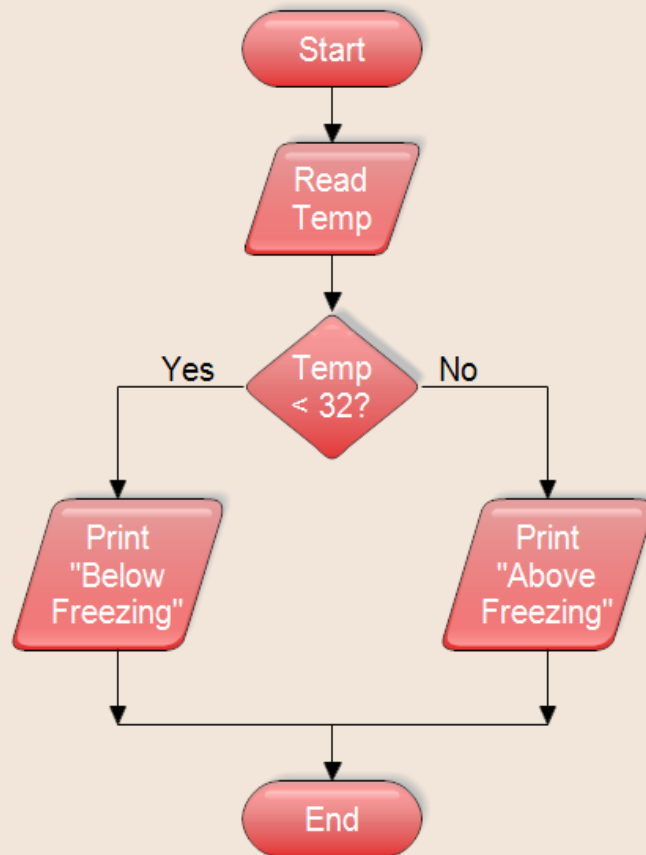
Example -1:

Read in the temperature. If the temperature is less than 32 indicate below freezing on the screen. Else if the temperature is above freezing then indicate the same on the monitor screen.

Divide the above problem into manageable parts.

Input ;	Read the temperature from keyboard
Processing;	Test the Temperature below or above freezing
Output;	Display the result on Screen

# Flowchart



# Define The Problem

## Example-2;

Determine the sum of first 50 natural numbers.

## Break into steps

Input – Nil

Processing: Sum the numbers from 1 to 50

Output - Sum

# Design the Solution

## Example-2;

Determine the sum of first 50 natural numbers.

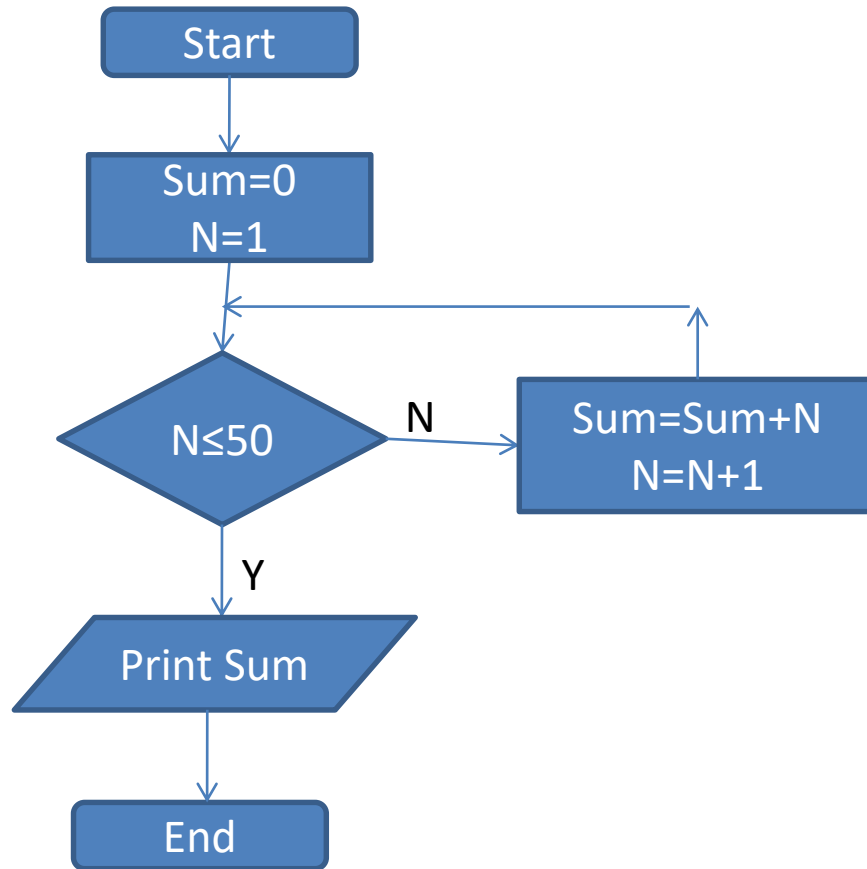
Algorithm; Pseudo Code

1.      Set  $N=1$   
         Set  $Sum = 0$
2.      Repeat step 3 & 4 while  $N \leq 50$
3.       $Sum = Sum + N$
4.       $N = N + 1$
5.      Print Sum
6.      end

# Design the Solution

**Example-2;** Determine the sum of first 50 natural numbers.

Algorithm; Flow Chart



Example-3:

Determine the factorial of input number



## Define The Problem

Example-3; Determine the factorial of input number.

## Break into steps

Input – Number is N

Processing: Factorial =  $N \times N-1 \times N-2 \times N-3 \dots 3 \times 2 \times 1$

Output - Factorial

# Design the Solution

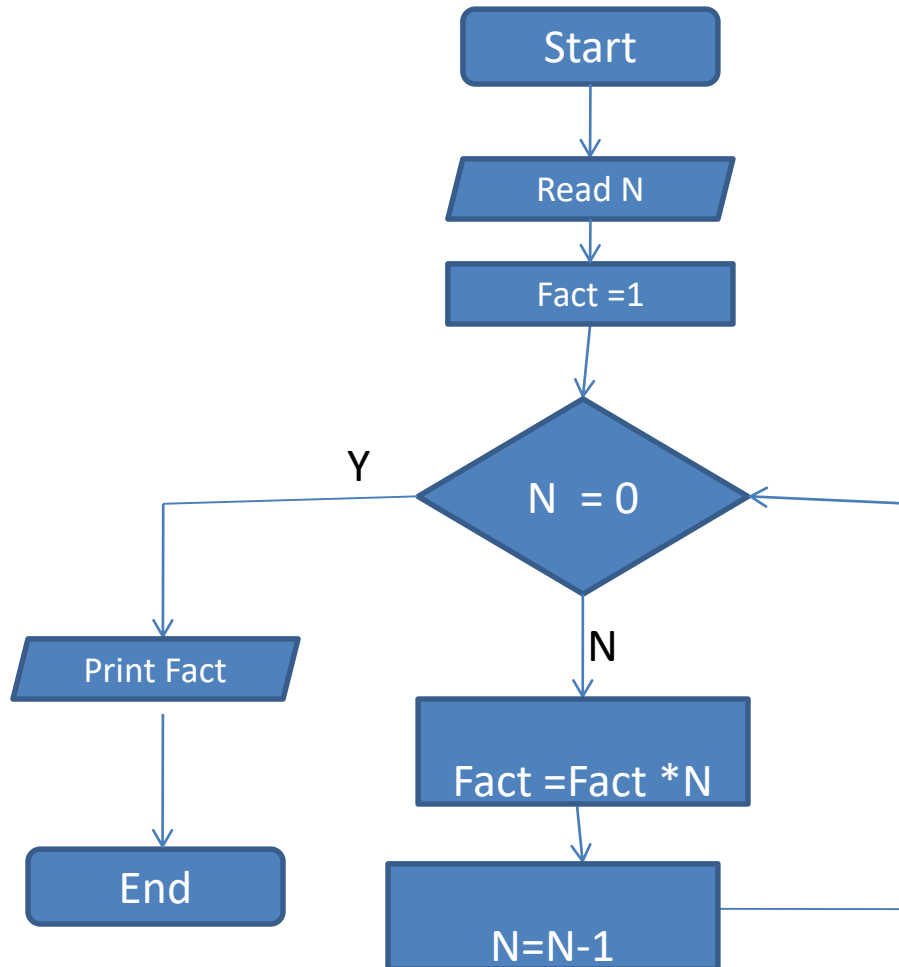
Example-3; Determine the factorial of an input number.  
(assume number is positive)

## Algorithm; Pseudo Code

1. Set Factorial = 1
2. Read N from keyboard
3. if ( N = 0 ) goto step 6
4. Factorial = Factorial x N
5. N = N – 1 ; goto step 3
6. Print Factorial
5. end

# Design the Solution

Determine the factorial of an input number.  
Algorithm; Flowchart



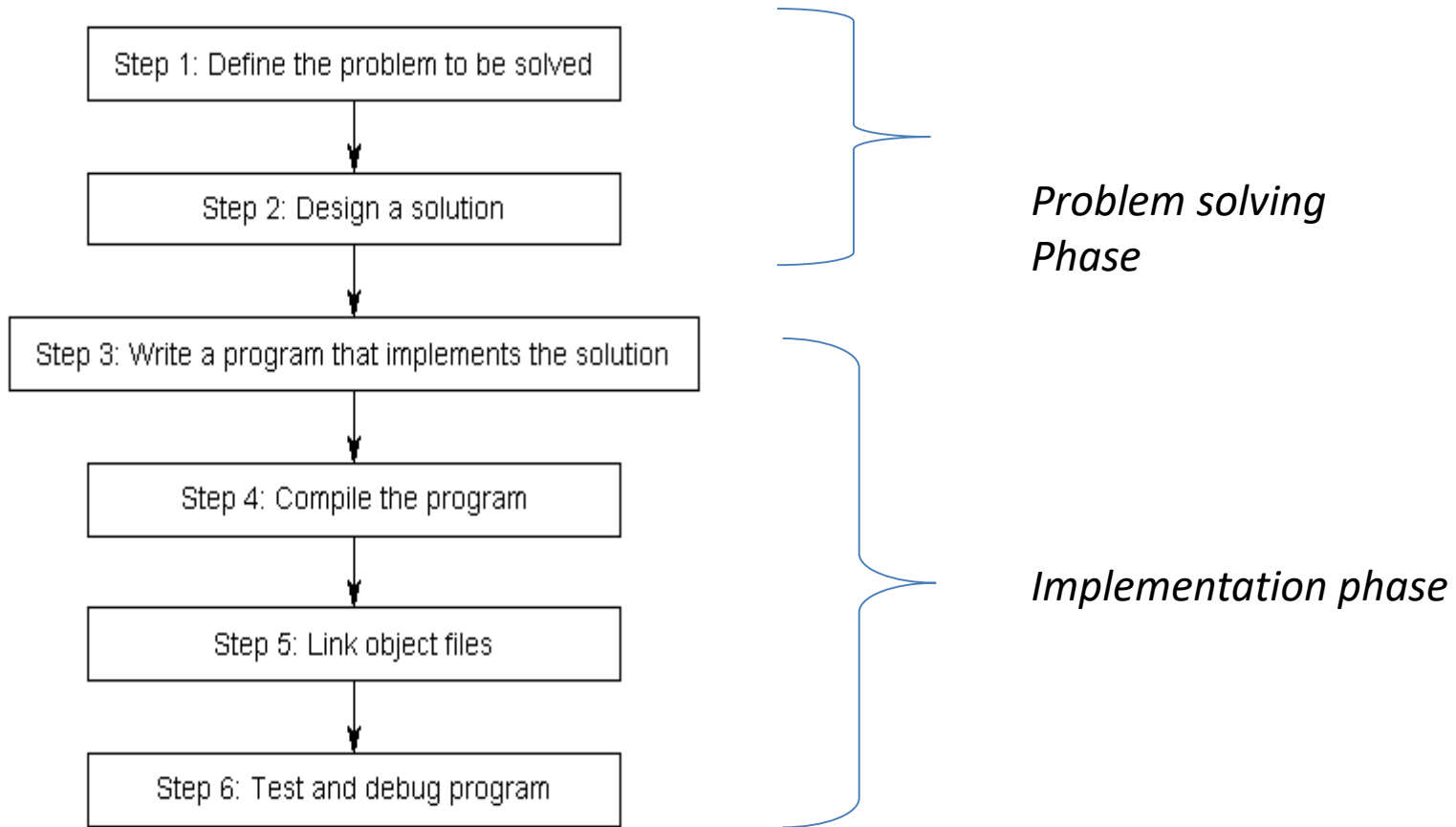
# Control Structure:

## Definition;

A control structure or logic structure is a structure that controls the logical sequence in which the instructions are executed. Three types of control structure are used:

- Sequence
- Selection
- Iteration ( or loop)

# Developing a program



# Implementation Phase

- Write a program (source code)
- Compile a program (source code to Object code)
- Link a Program ( Object code to Executable code)
- Test and Debug the Program (rectify the errors in the program)

# Write a code

Create a source code we need an editor

- Line editor --- line by line editing
- Screen editor --- note pad, word pad, customized editor
- After writing the code we save the code with file extension  
e.g .c .cpp

# Compile a program

source code to Object code

- We need a compiler e.g FORTRAN, PASCAL or C
- It converts user readable code to machine readable code
- Cannot be executed because different sections are not mapped together, say not linked together



# Link a Program

Object code to executable code

- `first.obj` to `first.exe`
- Can be executed because different sections are mapped together.
- Execute the code by simply typing the name of file `first.exe` or even `first`

# Test and Debug the program

Rectifying logical errors in the code, manually or may Use debugger for the assistance. A debugger may:

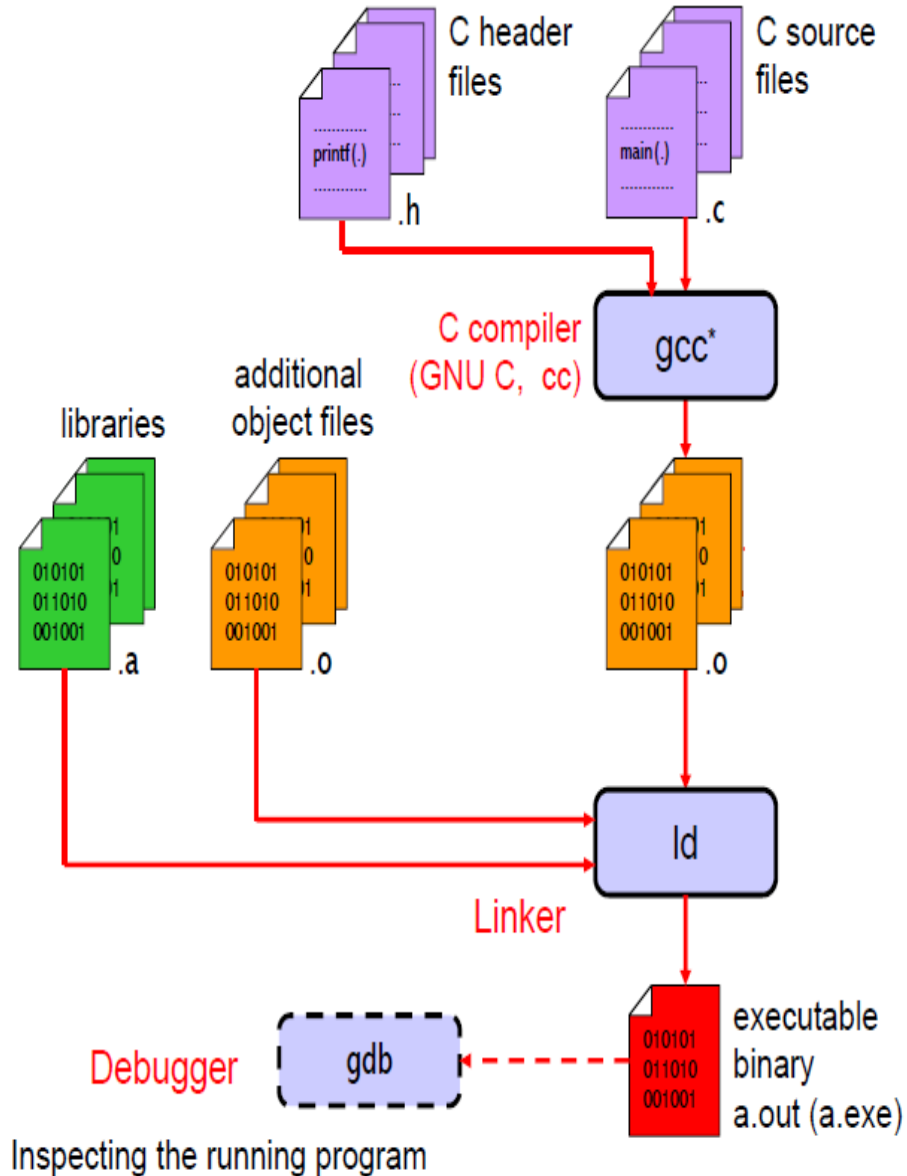
- Execute the code line by line or block by block or in one go
- Examine the contents of variables and registers at each step
- Examining the flow of control structure by tracing code step by step.

# Program Execution

Two ways:

1. Use command prompt e.g DOS or UNIX command prompt
2. Use Integrated Development Environment

# Command prompt



# Integrated Development Environment (IDE)

1. A source code editor

2. A compiler and / or an interpreter

- Compiler – translates complete code in one go
- Interpreter – translates code one line at a time

3. Build automation tools

4. A Debugger

# Typical C environment

