GE6151 COMPUTER PROGRAMMING

Dr. PAULRAJ M P,
SRIT
COIMBATORE

Email: principal@srit.org

Course Outcomes

CO1 : Ability to define the basic programming techniques.

CO2: Ability to apply suitable programming techniques to solve a given problem.

CO3: Ability to develop and analyze computer programs in C and C++ for Engineering Applications.

General Information

- **Main Objective:**
  - **Students can independently write a computer program to solve problems related to engineering.**

Evaluation Contribution

Examination:
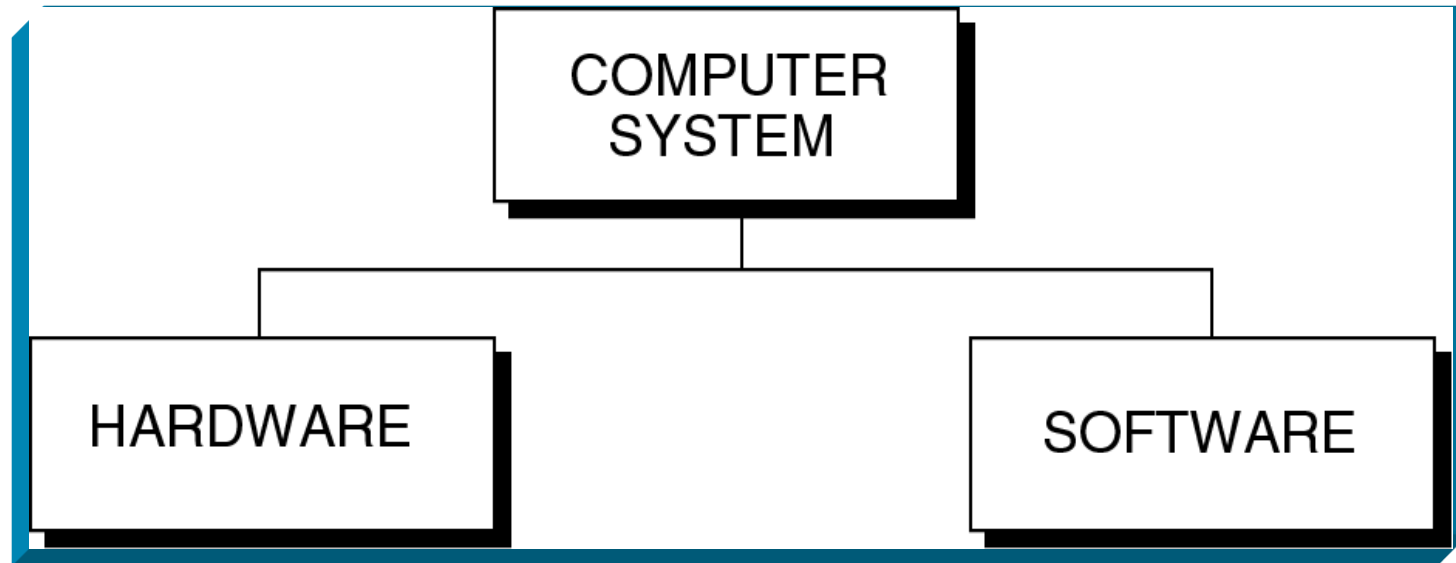**Periodic Tests (I, II, III) : 20%**
Final Examination : 80%

Total: 100%

References

- C How to program, Deitel, Suhizaz,

  R Badlishah, Yasmin, Pearson Prentice Hall

- Beginning Visual C++", Ivor Hortons, Wiley Publishing, Inc Indiana 2003.

- "C Programming for Engineering & Computer Science ", H.H.Tan and T.B.Orazio, Mc Graw Hill, 1999.

- Any other C programming books

INTRODUCTION TO COMPUTER SYSTEMS

```
                    ┌─────────────────┐
                    │    COMPUTER     │
                    │     SYSTEM      │
                    └────────┬────────┘
                   ┌─────────┴─────────┐
          ┌────────┴────────┐   ┌───────┴────────┐
          │    HARDWARE     │   │    SOFTWARE    │
          └─────────────────┘   └────────────────┘
```

| **Hardware** | → | **It is a Physical Equipment.** |

| **Software** | → | **It is a collection of Instructions.** |

HARDWARE COMPONENT

1. Input Devices

2. Central Processing Unit (CPU)

3. Primary Storage

4. Output Devices

5. Auxiliary Storage Devices

Keyboard

INPUT DEVICE

CPU

Primary Storage

Monitor

Disk

Tape

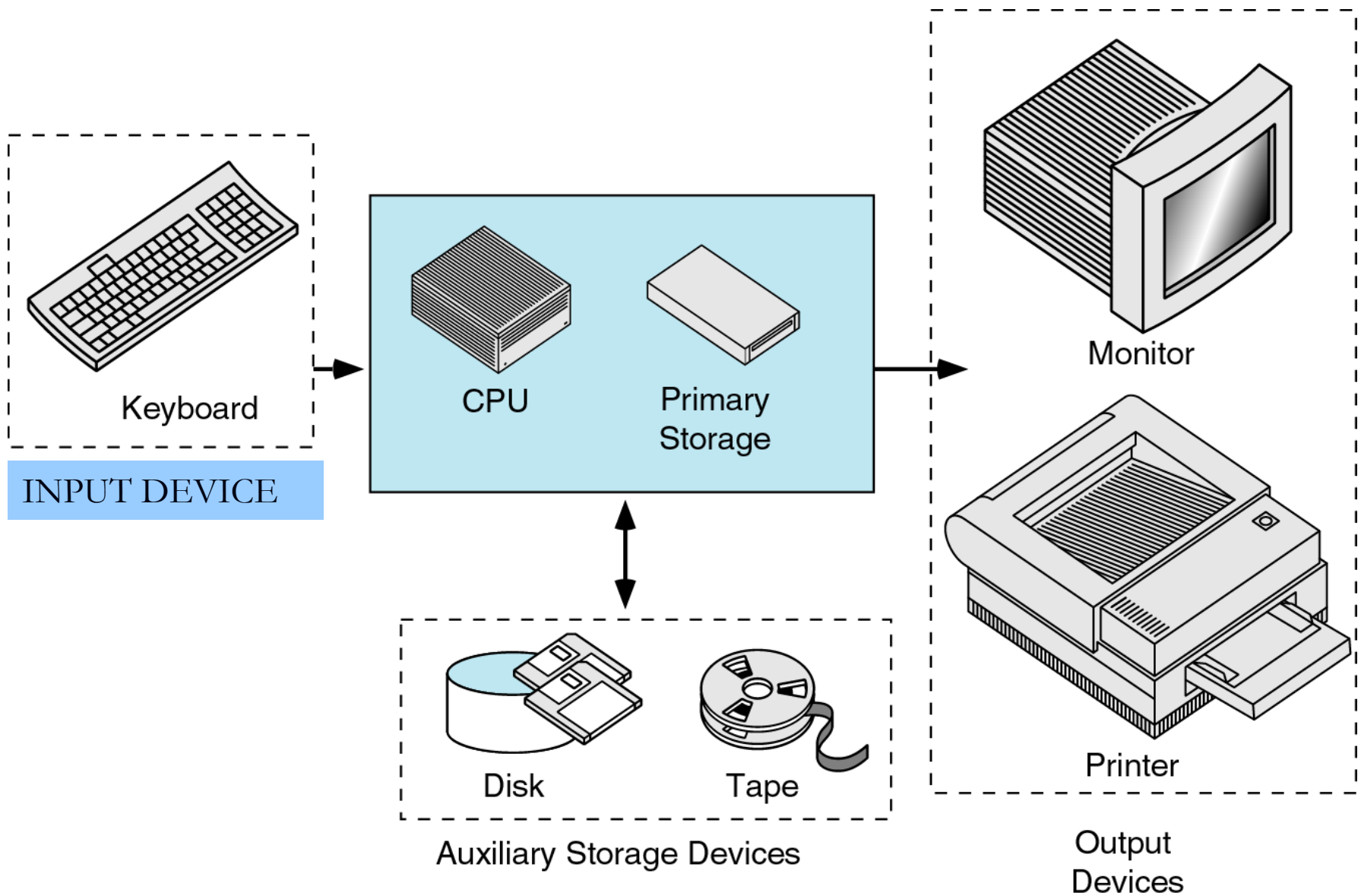Auxiliary Storage Devices

Printer

Output Devices

Figure Source: A Structured Programming Approach using C: Behrouz A Forouzan

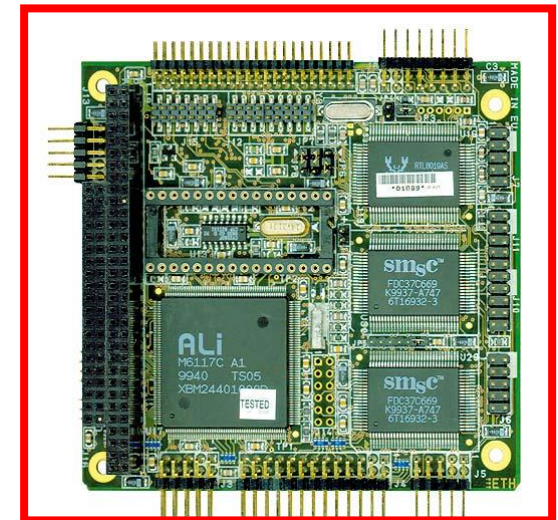# Hardware: Physical Component

Input Device

Through which programs and data are entered.

Key Board, Mouse, a pen or stylus, a touch screen.

Central Processing Unit (CPU)

Responsible for executing instructions such as arithmetic calculations, comparisons among data and movement of data inside the system.

Primary Storage

It is a place where data and programs are stored temporarily during processing.

The data in the main memory are erased when the system is turned off.

OUTPUT DEVICE

To show the processed result output devices are used.
If the output is shown in a monitor it is called soft copy.
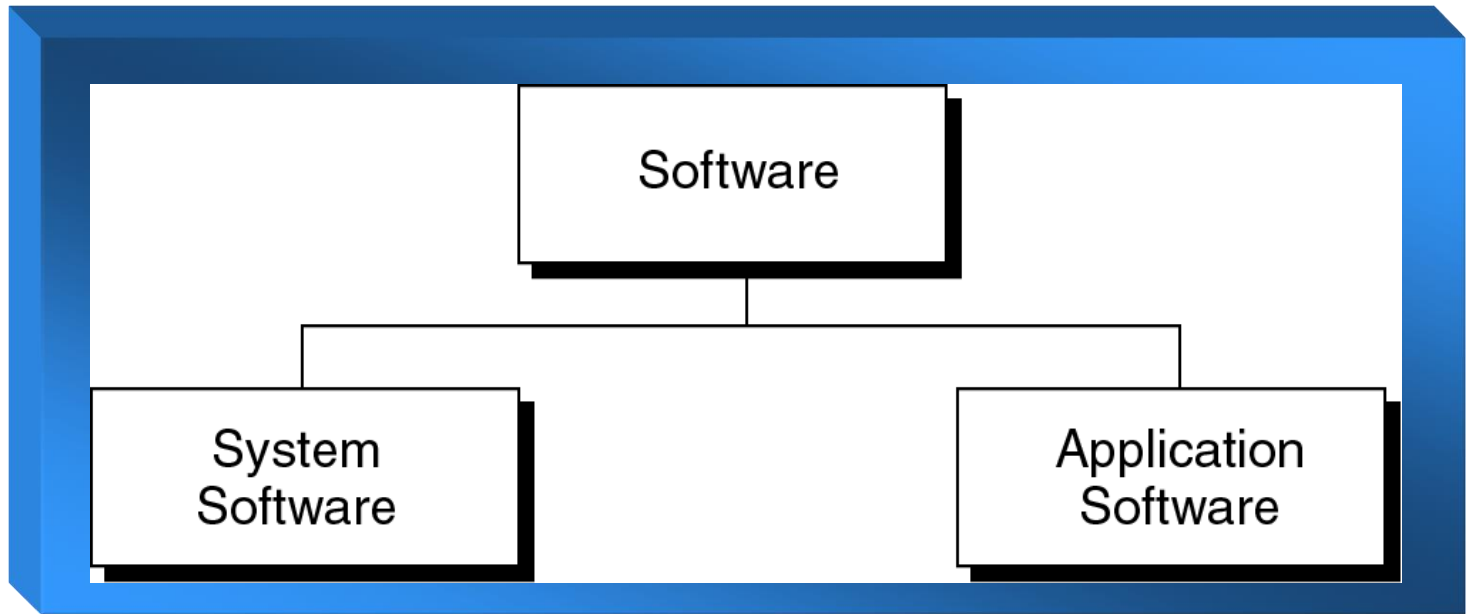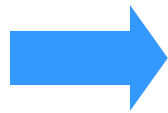**If it is printed on the printer, it is called as hard copy.**

It is the place where all the programs and data are stored permanently.
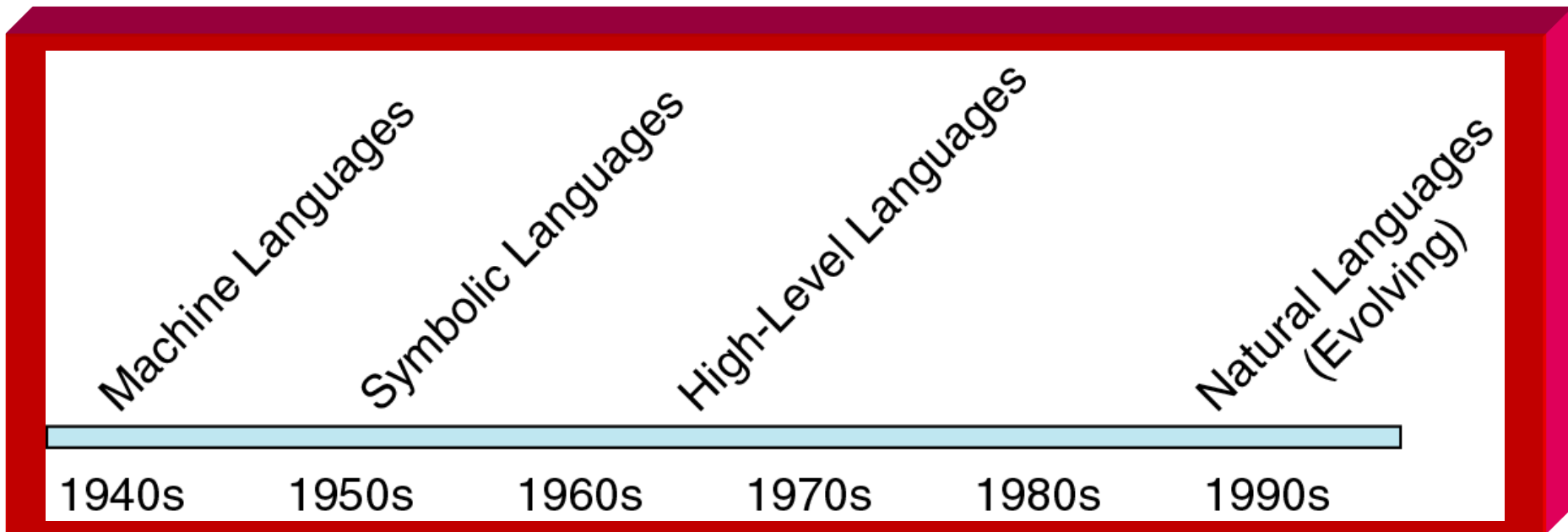
SOFTWARE

System Software → It manages the computer resources. It provides an interface between the hardware and the users but does nothing to directly serve the users.

COMPUTER PROGRAM

- **It is a sequence of instructions used to perform a job.**

- **Programming is the process of writing the instructions in a language that the computer can understand.**

PROGRAMMING LANGUAGE

The set of instructions that can be used to construct a program is called a programming language

Machine Languages

Symbolic Languages

High-Level Languages

Natural Languages (Evolving)

| 1940s | 1950s | 1960s | 1970s | 1980s | 1990s |

COMPUTER LANGUAGES

MACHINE LANGUAGE

Made up of '0' s and '1' s

Example: 00110011 is an instruction

SYMBOLIC LANGUAGE

Mnemonics are used to denote the strings of '1's and '0's.
Example: ADD A

HIGH LEVEL LANGUAGES

Working with symbolic language is very difficult.

To improve programmer's efficiency High level languages are developed.

HLL are portable.

HLL relieve the programmer from the assembly language

HIGH LEVEL LANGUAGES

i) FORTRAN – FORmula TRANSlation
Created by John Backus in 1957
Used for Scientific and Engineering applications.

ii) COBOL – Common Business Oriented Language
Developed by Admiral hopper

iii) C – Middle level language
Developed by Dennis Ritchie at AT&T Bell Laboratories
of USA in 1972.

HOW TO FEED YOUR PROGRAM TO A COMPUTER?

A text editor is used to Enter, Modify and Save the programs and data.
Some of the text editors are available with search commands to locate and replace statements, copy and paste commands to copy or move statements from one part to another.
The programs written and saved will be input to the compiler. The program is known as source program.

COMPILER

Compiler is a program that converts source program into machine language.
A C compiler contains two separate programs as : preprocessor and translator.

PREPROCESSOR

A preprocessor search for the preprocessor directives (special commands) and make substitutions of code. After the preprocessor has prepared the code for compilation, the translator converts the program into machine language and this code is called object code.

LINKER

Even though the output of the compiler is machine language, it can not be executed. The object code is linked to the library functions using the linker and a executable code is produced. The final executable program is executed in the system.
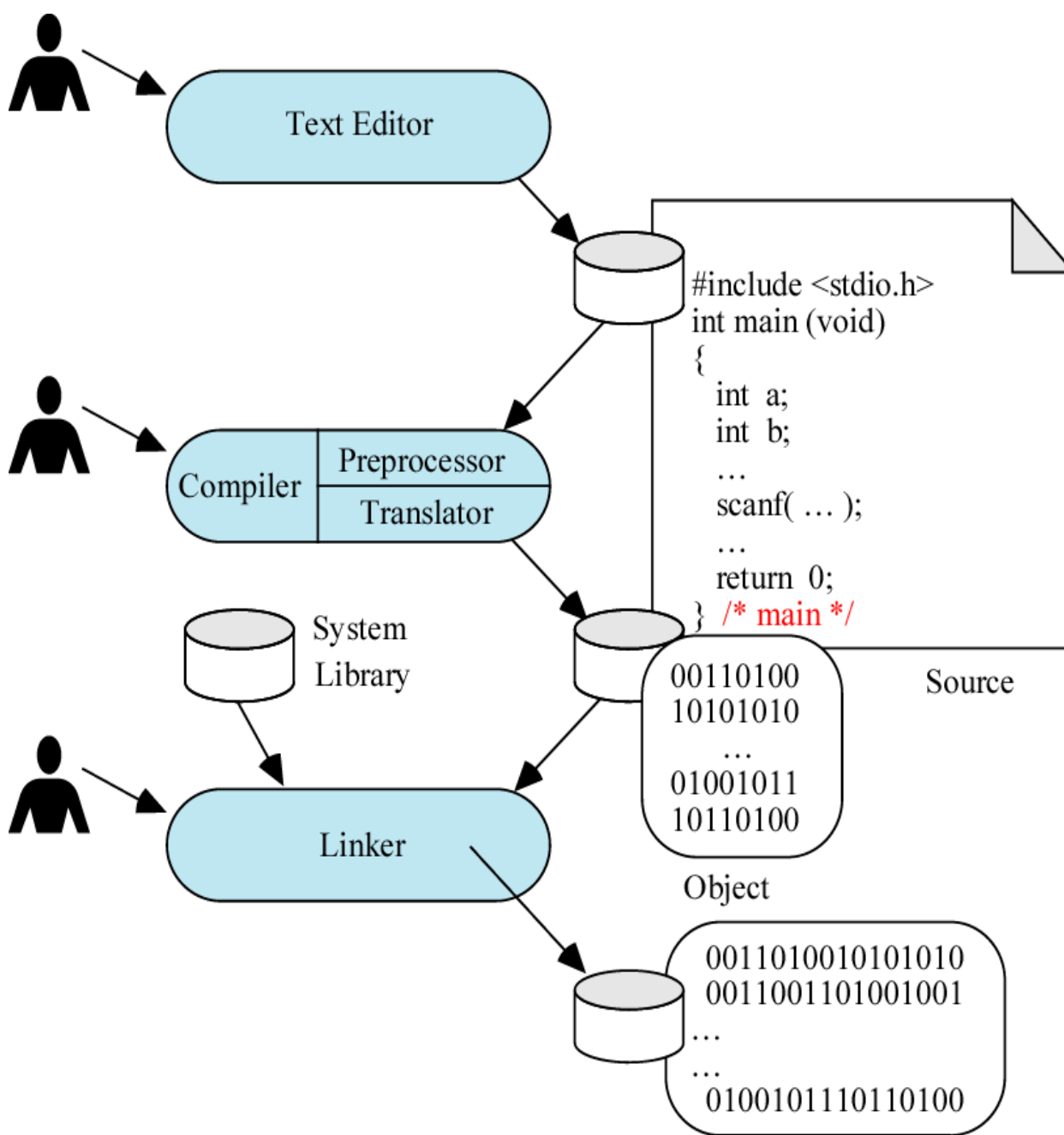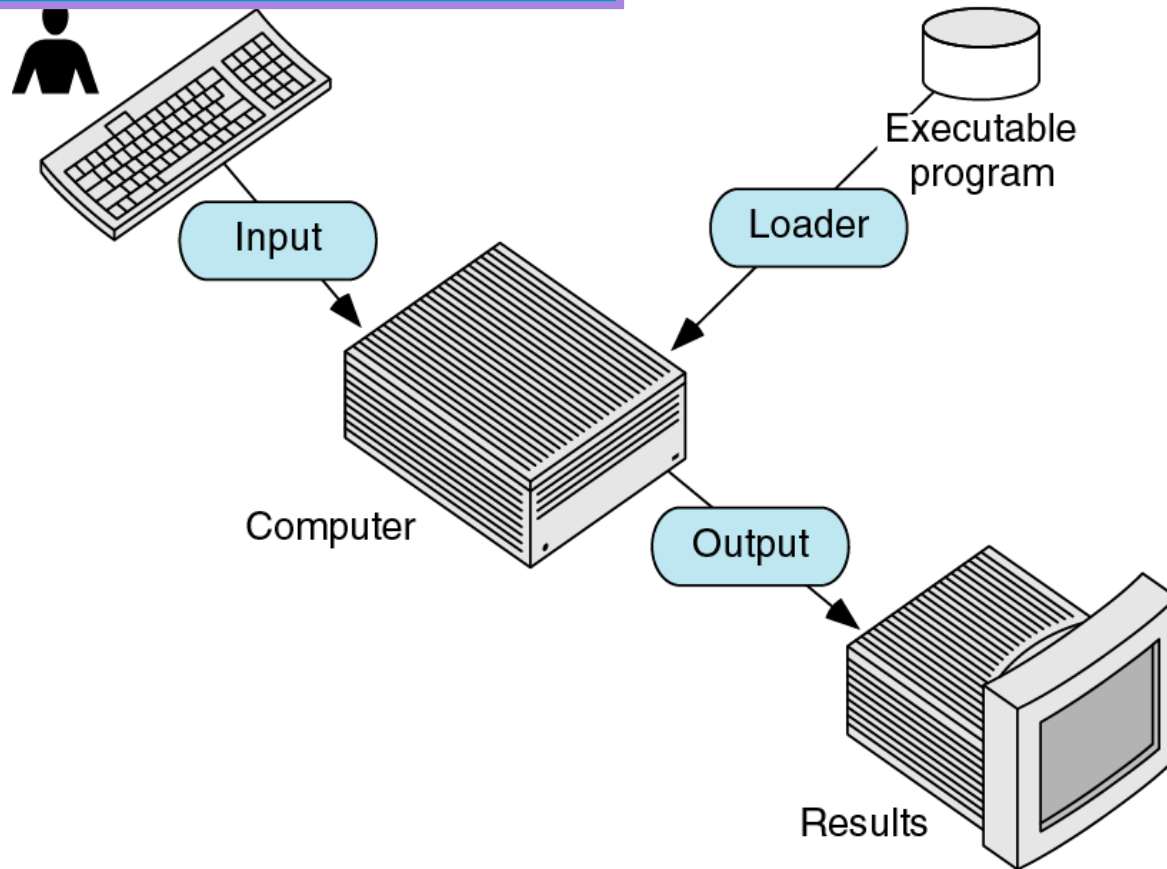
Figure Source: A Structured Programming Approach using C: Behrouz A Forouzan

Executable program

Loader

Input

Computer

Output

Results

To execute your program, use an operating system, load your program in the primary memory and execute it. The program is loaded into the primary memory using a system program called Loader.

Figure Source: A Structured Programming Approach using C: Behrouz A Forouzan

PROBLEM SOLVING

DEFINE THE PROBLEM

LOOK AT POTENTIAL CAUSES FOR THE
PROBLEM
SELECT AN APPROACH TO RESOLVE THE
PROBLEM

IDENTIFY ALTERNATIVES FOR APPROACHES
TO RESOLVE THE PROBLEM

PLAN THE IMPLEMENTATION OF THE BEST
ALTERNATIVE

MONITOR IMPLEMENTATION OF THE PLAN

VERIFY IF THE PROBLEM HAS BEEN
RESOLVED OR NOT

TOOLS TO DEVELOP THE SOLUTION

STRUCTURE CHARTS

Used to design the whole program, It is also known as hierarchy chart. This chart shows the functional flow of the inter-related components. The structure chart shows how you are going to break your program into logical steps; each step will be a separate module. It also shows the interaction between various modules.
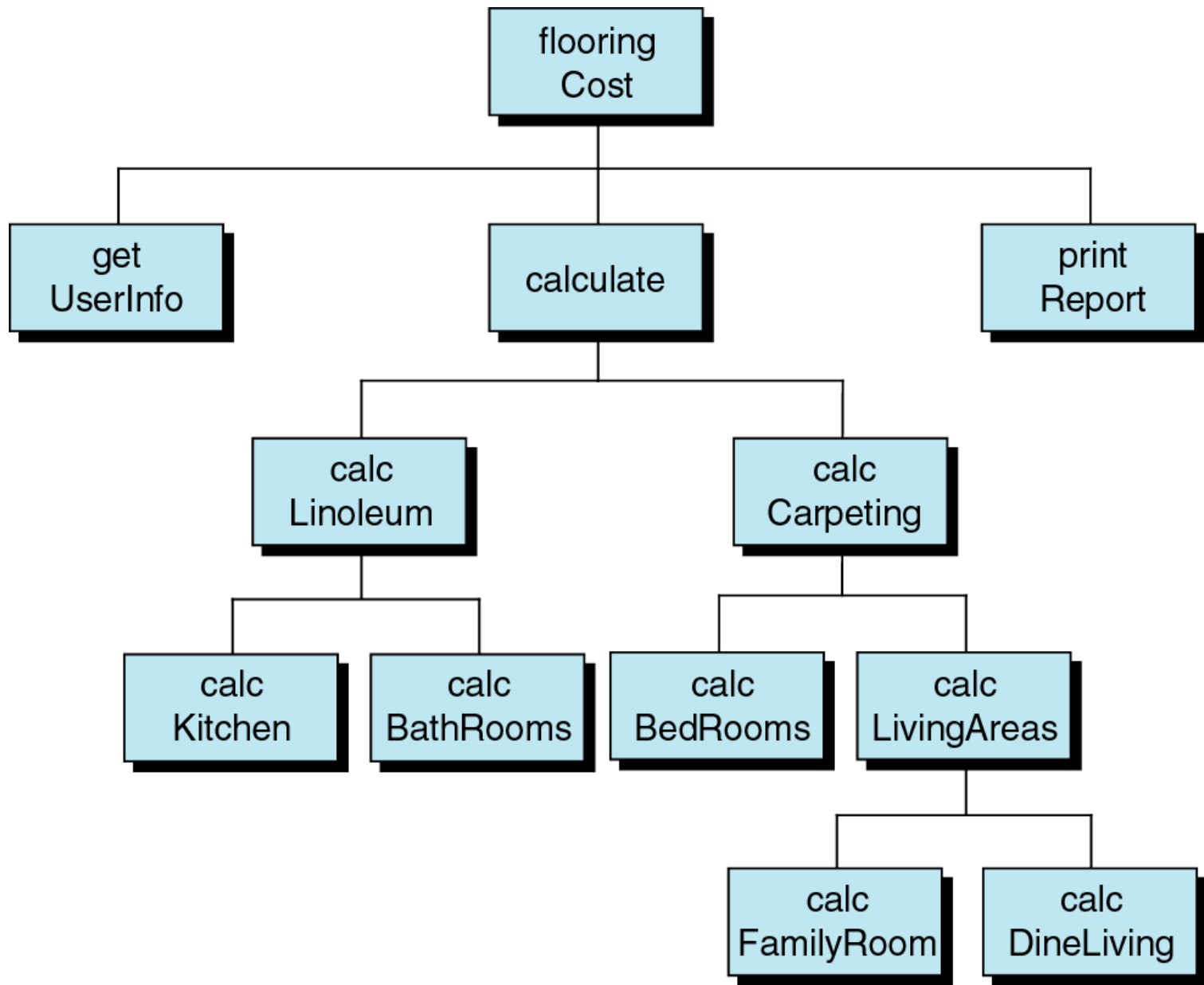
Figure Source: A Structured Programming Approach using C: Behrouz A Forouzan

PSEUDO CODE

English – like statements that follow a loosely defined syntax are used to convey the design of the algorithm.
Example: Pseudo code to add three number and to display the average:
Step 1: Input three numbers into the computer.
Step 2: Add all the three numbers and divide it by 3.
Step 3: Display the result as average.

Example:

input A
input B
C = A + B
print C

FLOW CHART

It is a program design tool in which standard graphical symbols are used to represent the logical flow of data through a function.

A flowchart is a diagrammatic representation that illustrates the sequence of operations to be performed to get the solution of a problem.

Flowcharts are generally drawn in the early stages of formulating computer solutions.
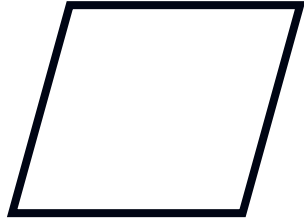
Flowcharts facilitate communication between programmers and business people. These flowcharts are quite helpful in understanding the logic of complicated and lengthy problems. Once the flowchart is drawn, it becomes easy to write the program in any high level language.

Terminal	Predefined Process
Input/Output	Connector
Process	
Flow Lines	
Decision	
Loop	
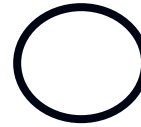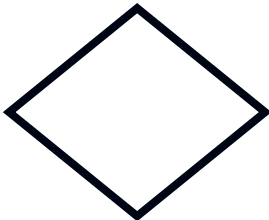
Flow Chart Example 1:

The Best Way Home

- Leaving the Office
- Check the Time and Weather
- Weather Clear? — No → (left path)
- Before 5:00pm? — No → (left path)
- Yes → Check for congestion on primary route
- Primary congested? — No → Take the Primary Route Home
- Yes → Divert to Alternate "B"
- (No path) → Take Alternate "A" Home
- Arrive Safely
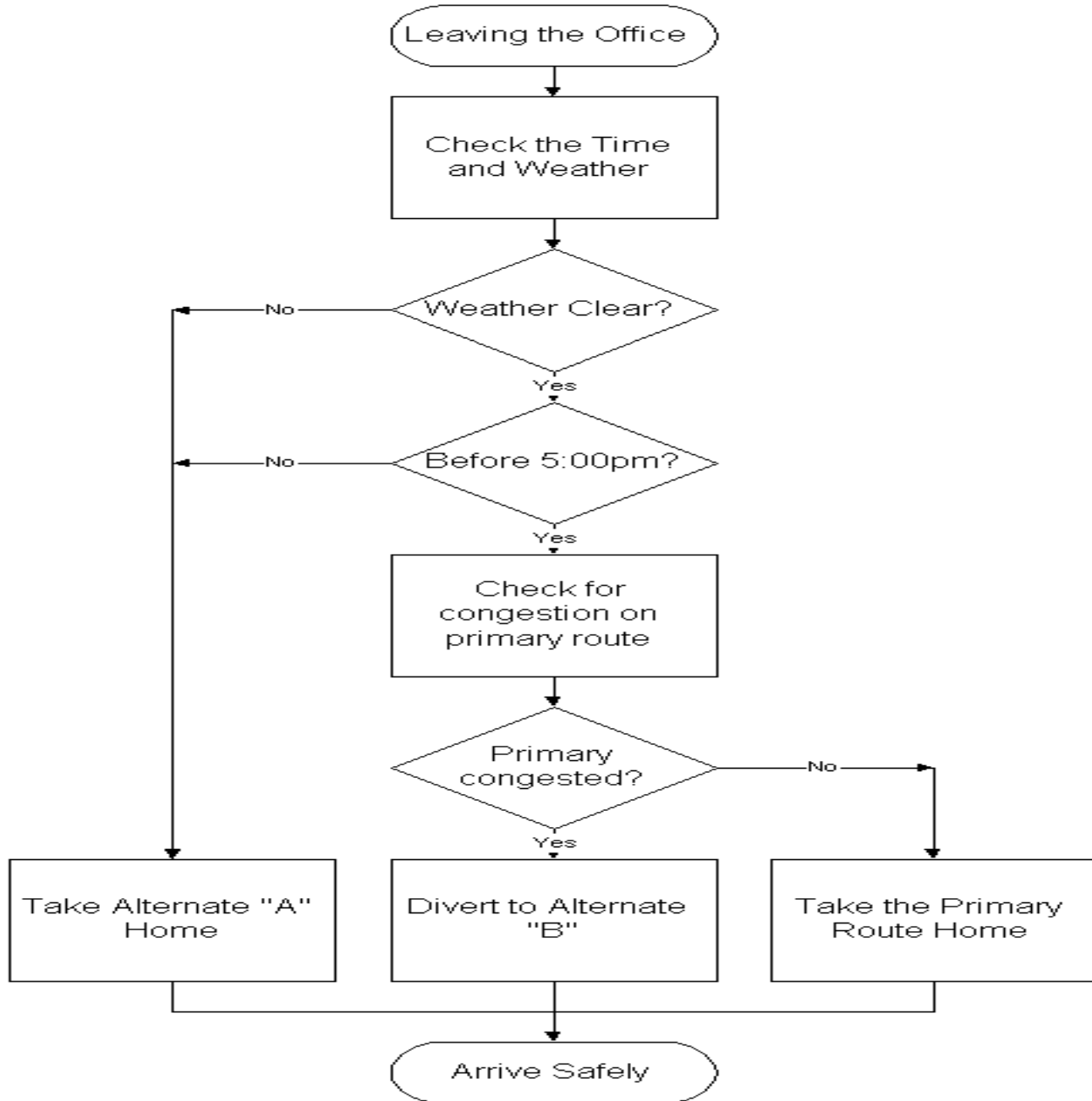
Example 2: Largest of Three Numbers

```
                    Start
                      |
                      v
              Read A, B, C
                      |
                      v
         FALSE      A > B      TRUE
          |                      |
          v                      v
    TRUE  B > C  FALSE    FALSE  A > C  TRUE
     |          |          |            |
     v          v          v            v
B is Greater  C is Greater      A is Greater
     |            |                |
     +----------> End <------------+
```
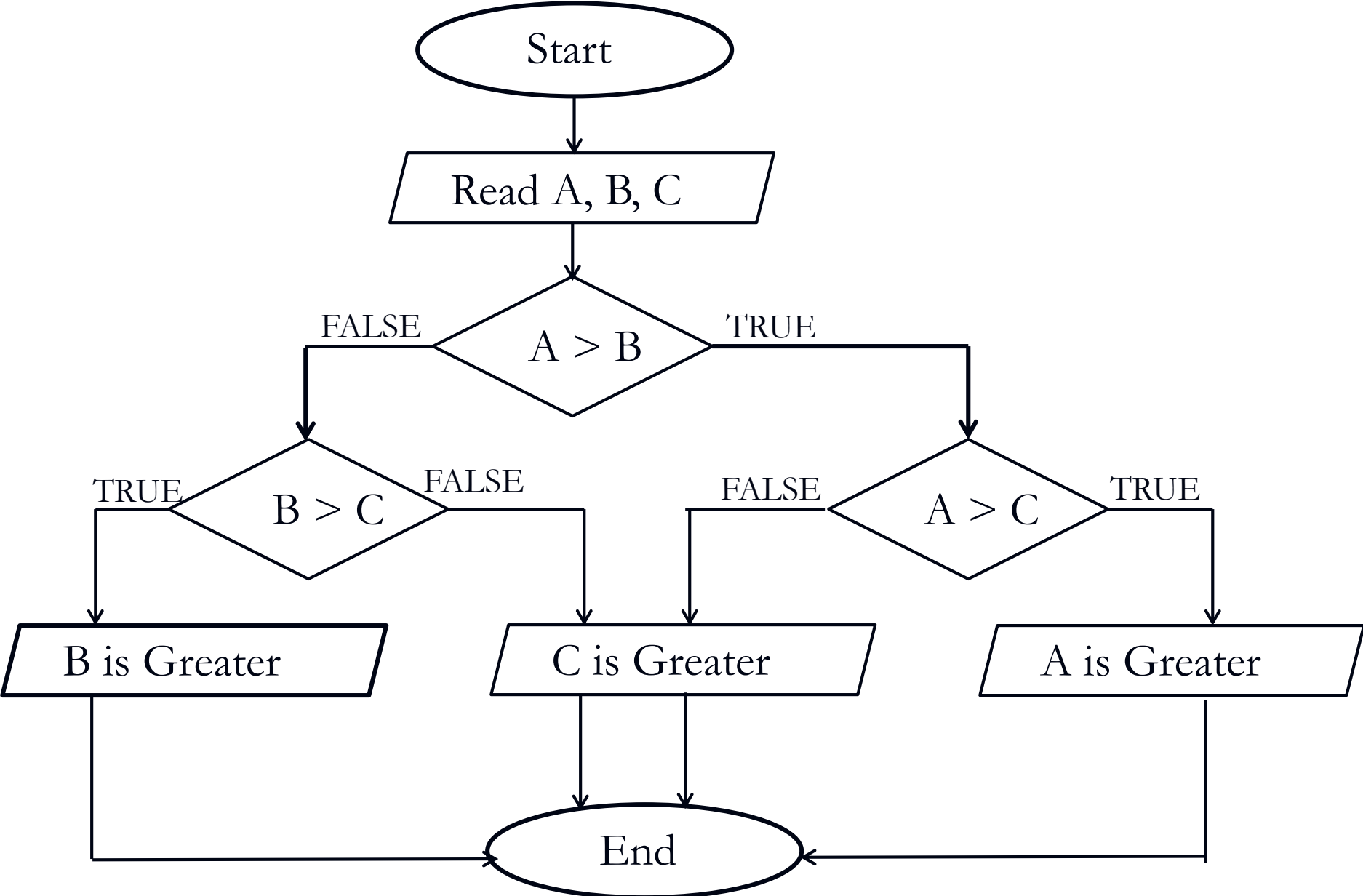
Example 3: To find the sum of 1+2+3+…+50

```
          ┌─────────────┐
          │    START    │
          └──────┬──────┘
                 │
                 ▼
          ┌─────────────┐
          │   SUM = 0   │
          └──────┬──────┘
                 │
                 ▼
          ┌─────────────┐
          │    N = 0    │
          └──────┬──────┘
                 │
       ┌─────────▶
       │         ▼
       │  ┌─────────────┐
       │  │  N = N + 1  │
       │  └──────┬──────┘
       │         │
       │         ▼
       │  ┌──────────────────┐
       │  │  SUM = SUM + N   │
       │  └────────┬─────────┘
       │           │
  NO   │           ▼
       │        ╱ IS N=50? ╲
       ◀───────◀            ▶
                 ╲          ╱
                   │ YES
                   ▼
            ╱ PRINT SUM ╲
                   │
                   ▼
          ┌─────────────┐
          │     END     │
          └─────────────┘
```
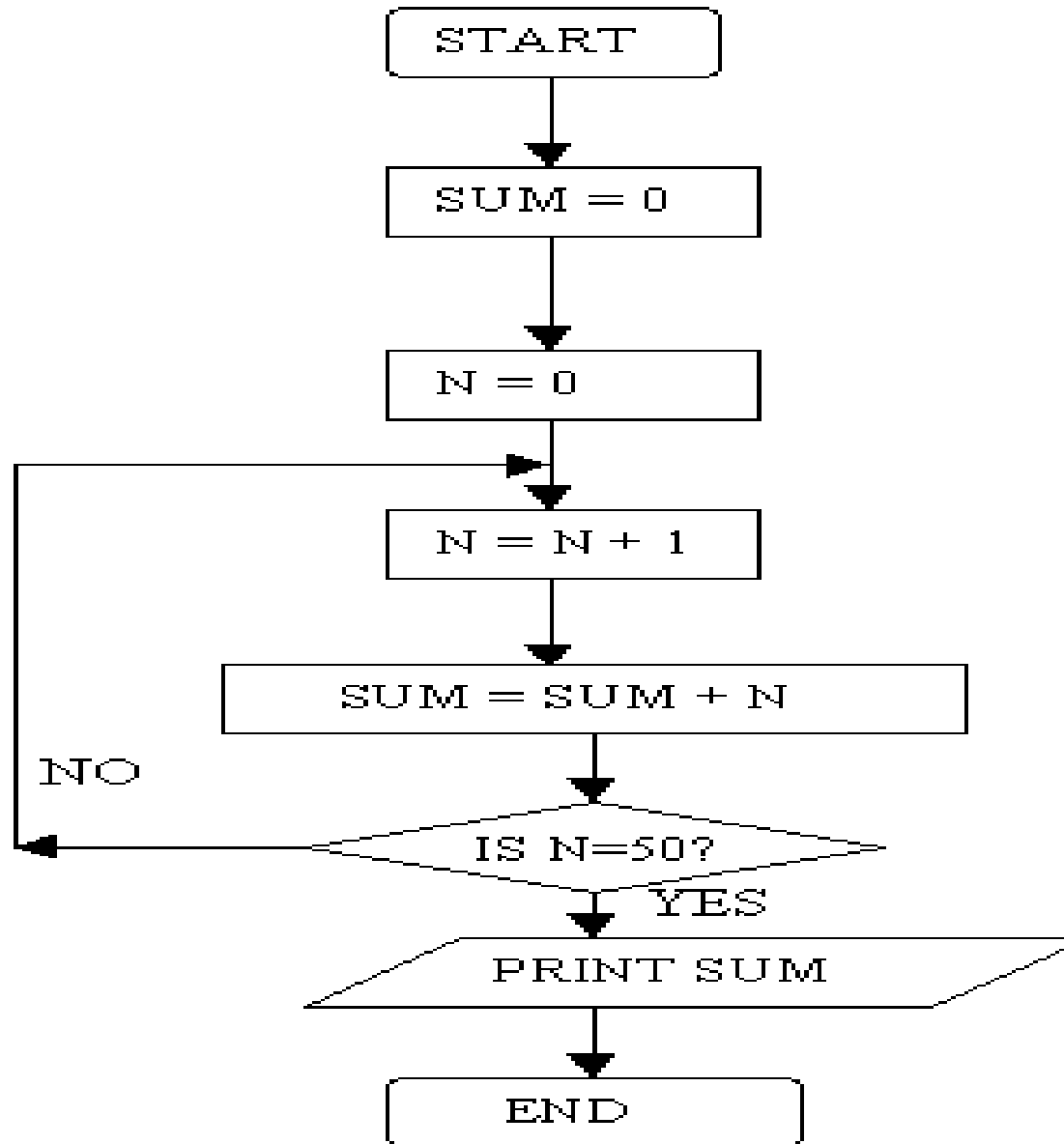
Example 3: To find the product of 1*2*3*…*50

START

READ N

M = 1
F = 1

F = F * M

IS M=N?

M=M+1

NO

YES

PRINT F

END
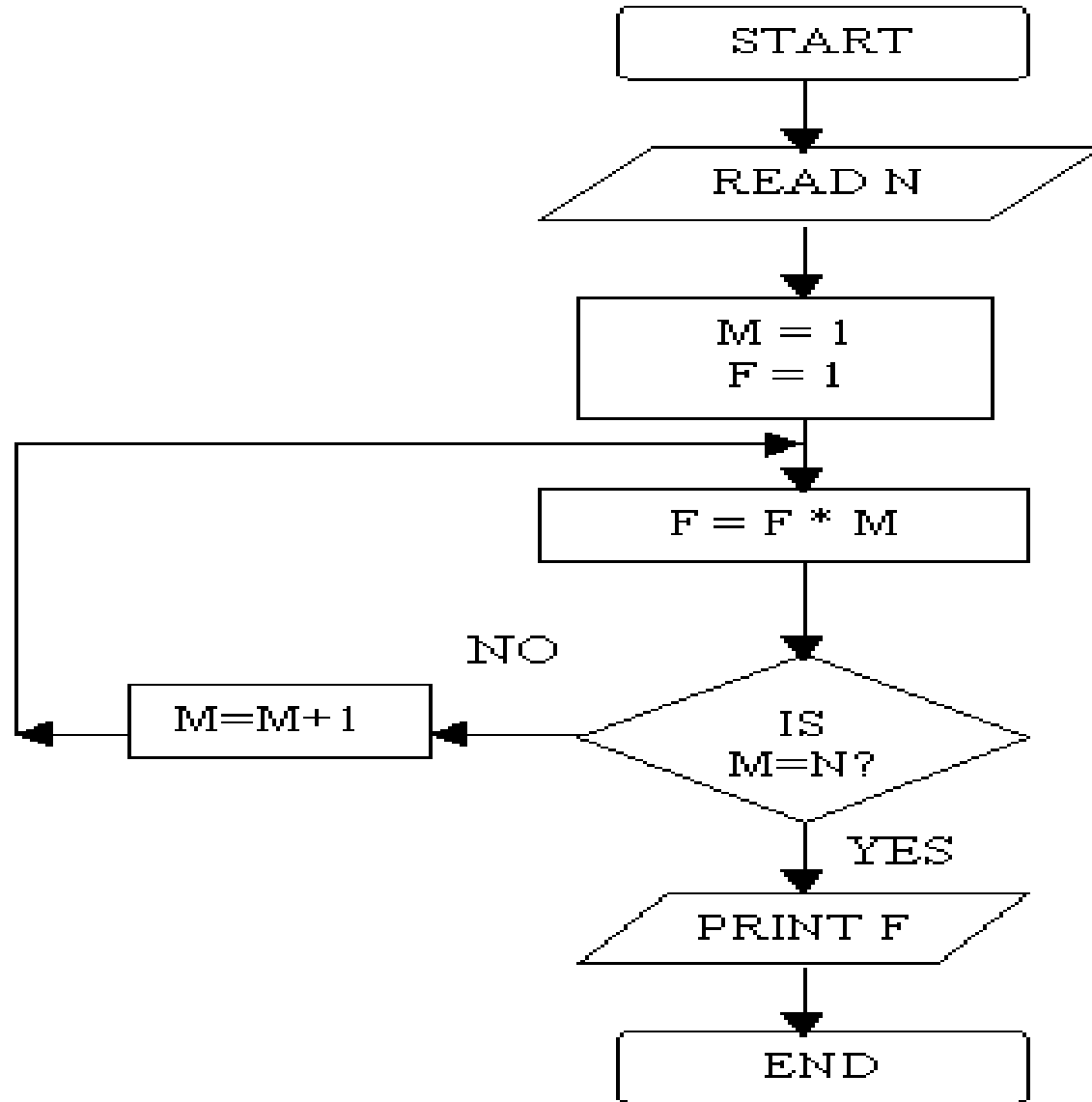
TESTING THE PROGRAMS

Program testing is very tedious and time consuming part of program development. There are two type of program testing

| BLACK BOX TESTING | In this method of testing, the program is tested without knowing what is inside the program. ie., The program is like a black box that we can not see. |

| WHITE BOX TESTING | In this method, the tester knows everything about the program. In this case program is like a glass house in which every thing is visible. |

Control Structure

- All programs could be written in terms of three control structures:
    - Sequence structure
    - Selection structure
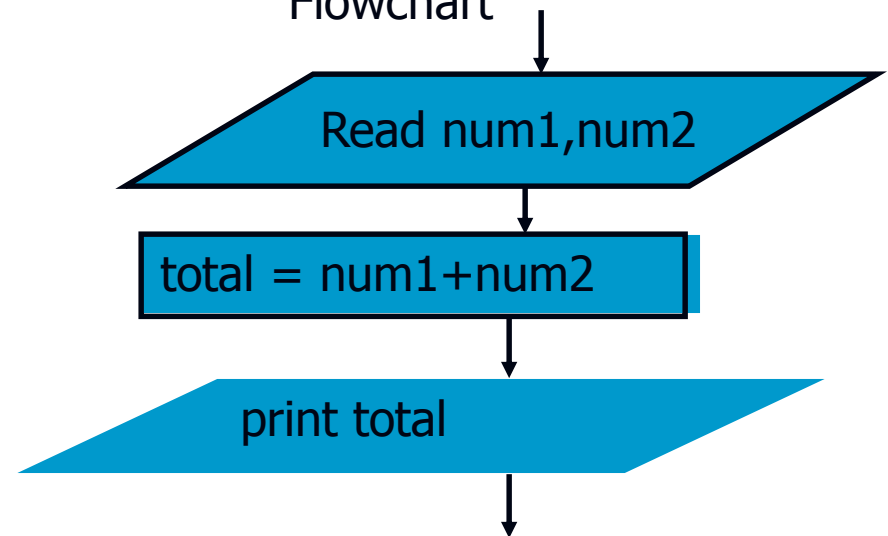   - Repetition structure

Sequence structure

- Is a series of steps executed sequentially by default

Pseudocode

Read num1,num2
Calculate total=num1+num2
Print total

Flowchart

Read num1,num2

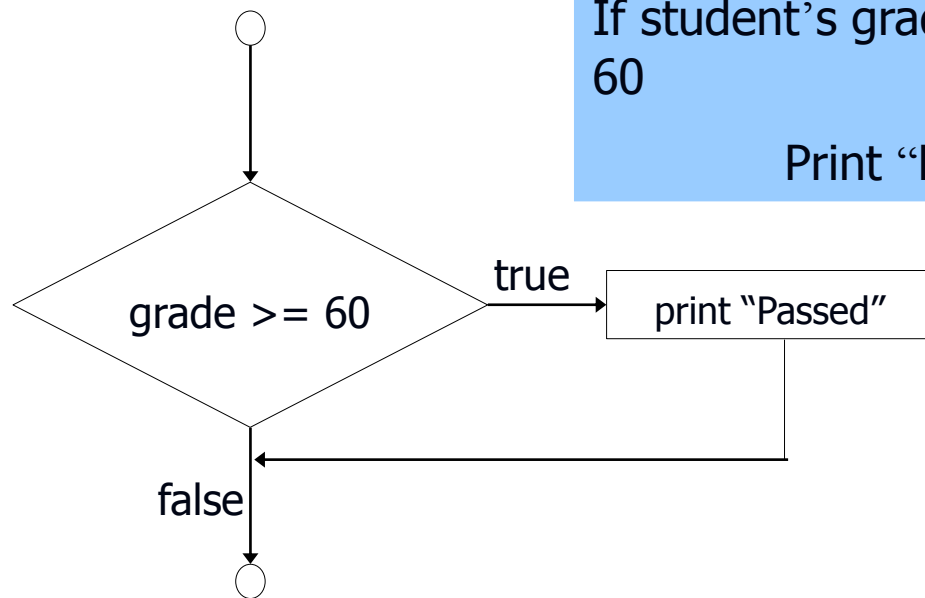total = num1+num2

print total

Selection structure

- Used to choose among alternative courses of action
- C has three types: **if**, **if**..**else**, and **switch**

The if selection structure

- **if** structure is a single-entry/single-exit structure

If student's grade is greater than or equal to 60

       Print "Passed"

```
                    ○
                    │
                    ▼
             ╱◇╲
          ╱      ╲        true
      ╱ grade >= 60 ╲ ─────────▶  ┌─────────────────┐
      ╲            ╱              │  print "Passed"  │
          ╲      ╱                └─────────────────┘
             ╲◇╱                          │
              │  ◀───────────────────────┘
        false │
              ▼
              ○
```

The if..else selection structure

- Specifies an action to be performed both when the condition is **true** and when it is **false.**

If student's grade is greater than or equal to 60
        Print "Passed"
else
        Print "Failed"

false   grade >= 60   true
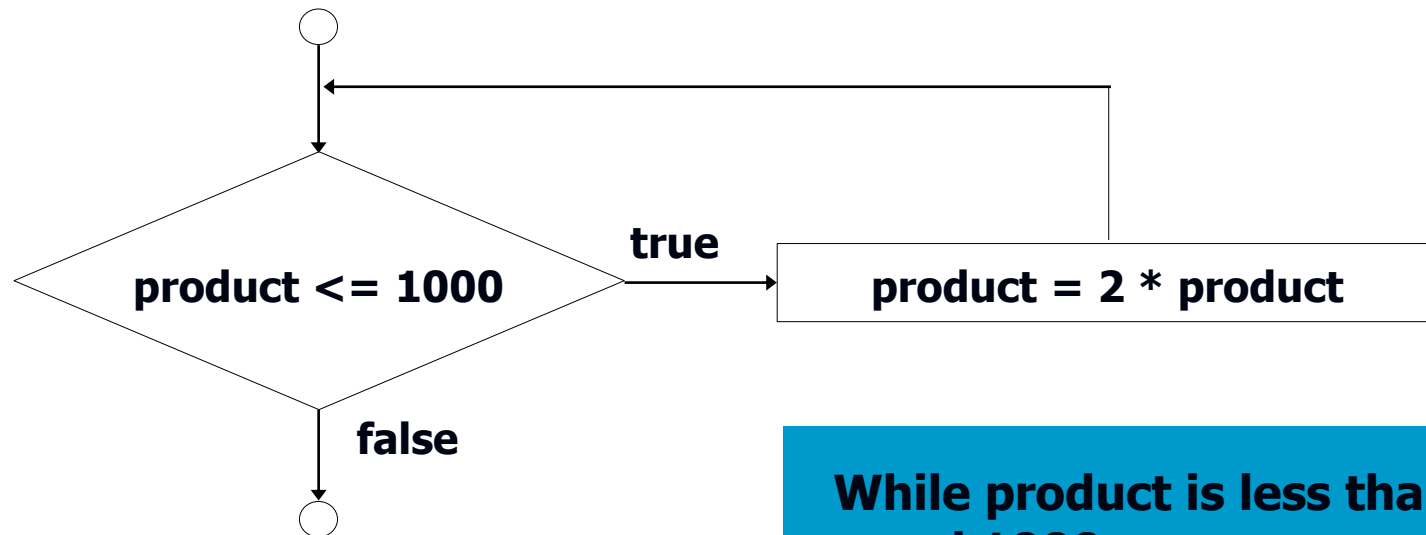
print "Failed"   print "Passed"
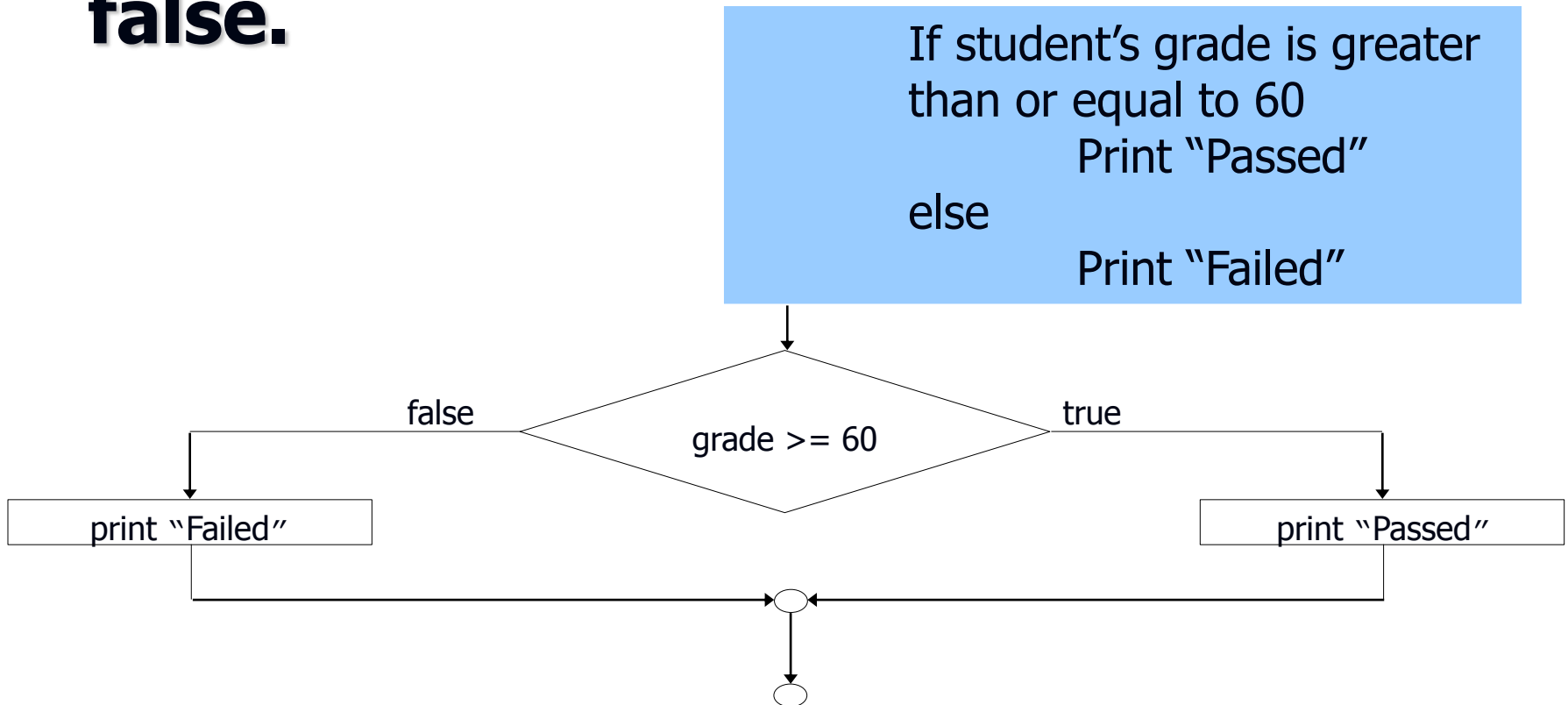
Repetition structure

- Specifies a block of one or more statements that are repeatedly executed until a condition is satisfied

- Three types : **while, for, do-while**

The while repetition structure

- Programmer specifies an action is to be repeated while some conditions remains true

product <= 1000 → **true** → product = 2 * product

false

While product is less than or equal 1000

calculate

product=2 * product

Lecture Review:

Computer Program: It is a sequence of instructions used to operate a computer to produce a specific result.

Programming: It is a process of writing the instructions in a language that the computer can understand and respond.

Programming language: The set of instructions that can be used to construct a program is called a programming language.

Algorithm: It is a step-by-step sequence of instructions that describes how to perform a computation.

Pseudo code: The description of an algorithm in a normal English phrase is known as a Pseudo code.

Flowchart: It is pictorial representation of an algorithm using different symbols.

Source Program: A sequence of instruction written in any high level language is known as a source program.

Object program: The machine language version of the source program is known as the object program.

Compiler: Compiler is a program that converts all the statements of (the entire program) a source program into a machine language before executing any one of the statement.

Interpreter: An interpreter is a program which translate each statement of a program individually and execute it immediately.
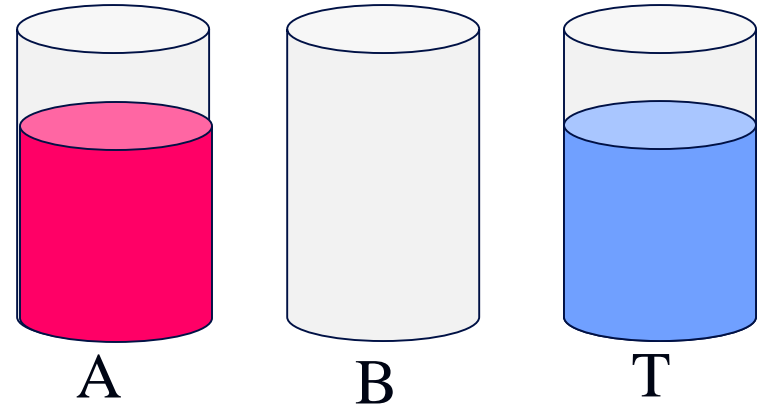
Determine and write an algorithm to interchange the contents of two cups of liquid. Assume that a third cup is available temporarily to hold the contents of either cup. Each cup should be rinsed before any new liquid is poured into it.
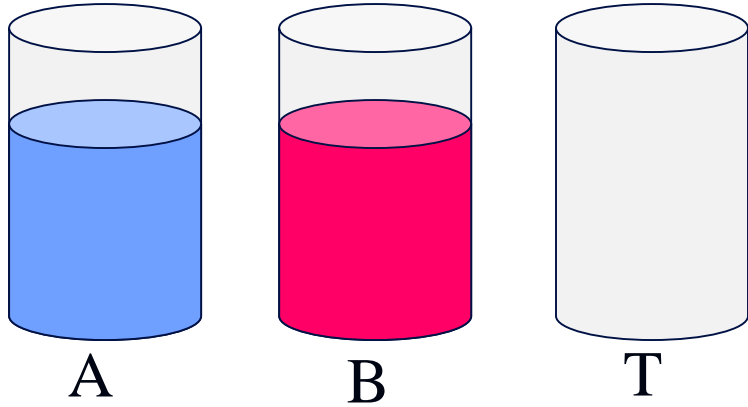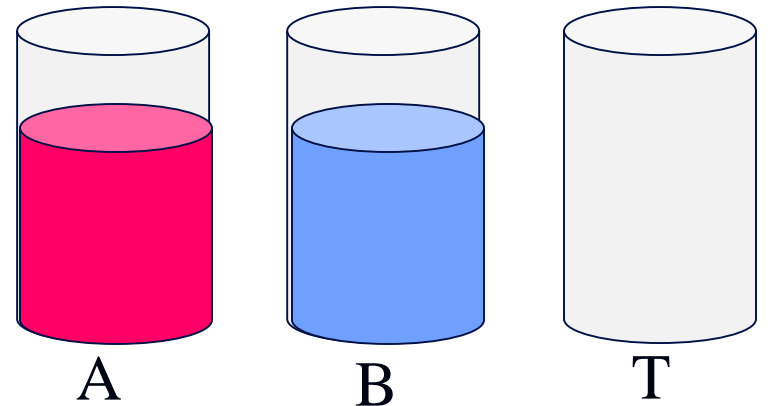

A B T
STEP 1: Transfer Cup A to Cup T

A B T
STEP 2: Rinse Cup A
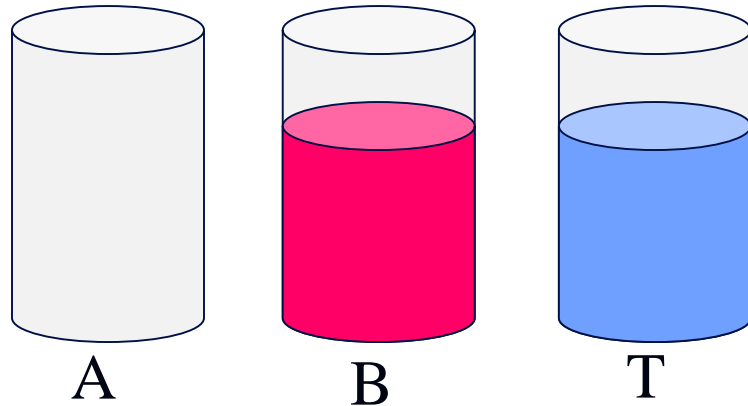
A B T
STEP 3: Transfer Cup B to Cup A

STEP 4: Rinse Cup B

A B T
STEP 5: Transfer Cup T To A

Write a detailed step-by-step instructions, in English, to find the smallest number is a group of three integer numbers.

Step 1: Compare the first number with the second number and place the smallest number in small.

Step 2: Compare the third number with the number in small. The smallest of these two numbers is the smallest number in the group.

Exercise:

1. Write an algorithm to determine the total occurrences of the letter 'e' in any sentence.

2. Write an algorithm to sort four numbers into ascending order.

3. Draw a flowchart to compute the area of a triangle given its three sides.

4. Draw a flowchart to read and check the given number is an odd or an even number.

END OF LECTURE-1

Questions? Problems? Need Directions? Ask Here!!!

FAQ

THANK YOU