ENT 189
COMPUTER PROGRAMMING

Dr. Lam Chee Kiang

Email: lckiang@unimap.edu.my

ENT 189
COMPUTER PROGRAMMING

Dr. Muhammad Izham Bin Ismail

Email: izham@unimap.edu.my

Schedule

- Pre-Lab (Week 3)
- Lab 1 (Week 4)
- Lab 2 (Week 6)
- Lab Test 1 (Week 7)
- Lab 3 (Week 10)
- Lab 4 (Week 12)
- Lab Test 2 (Week 14)

- Mid Term Exam (Week 8, 28th Oct, 8.30pm-10.00pm, Venue: TBA)

ENT 189: COMPUTER PROGRAMMING

Dr. PAULRAJ M P,
Associate Professor,
School of Mechatronic Engineering,
#42- Level 2,
Ulu Pauh New Campus
02600-Arau.
PERLIS

Email: paul@unimap.edu.my
H/P: 017 5103757
Home page: http://publicweb.unimap.edu.my/~paul/

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 Mechatronic Applications.

General Information

- Contributes 3 units:
  - 2 hours - lectures
  - 2 hours – lab and tutorial
- Main Objective:
  - Students can independently write a computer program to solve problems related to engineering.

Evaluation Contribution

Examination:

Mid Examination : 10%

Final Examination : 60%

Course Work:

Lab : 10%

Lab Test : 5%

Mini Project : 15%

Project : Team (Two/Three)

Lecture Content

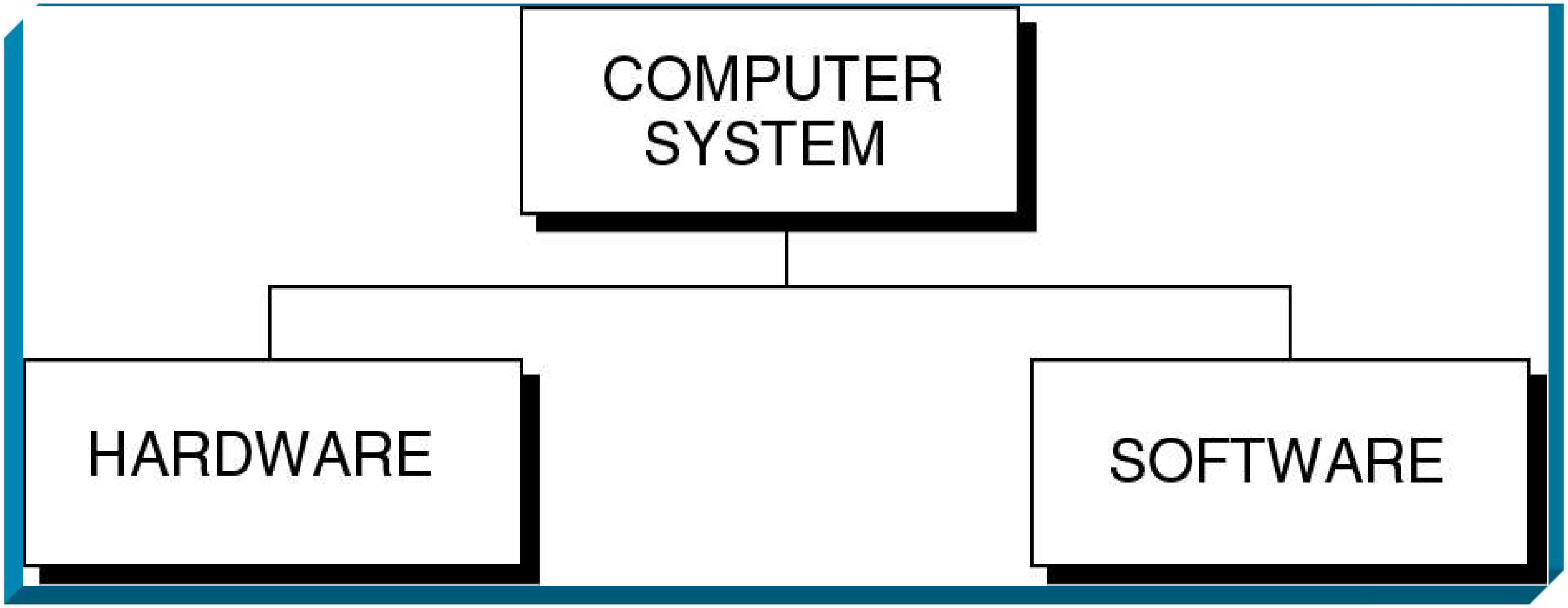
Lecture content can be accessed through
http://publicweb.unimap.edu.my/~paul/

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

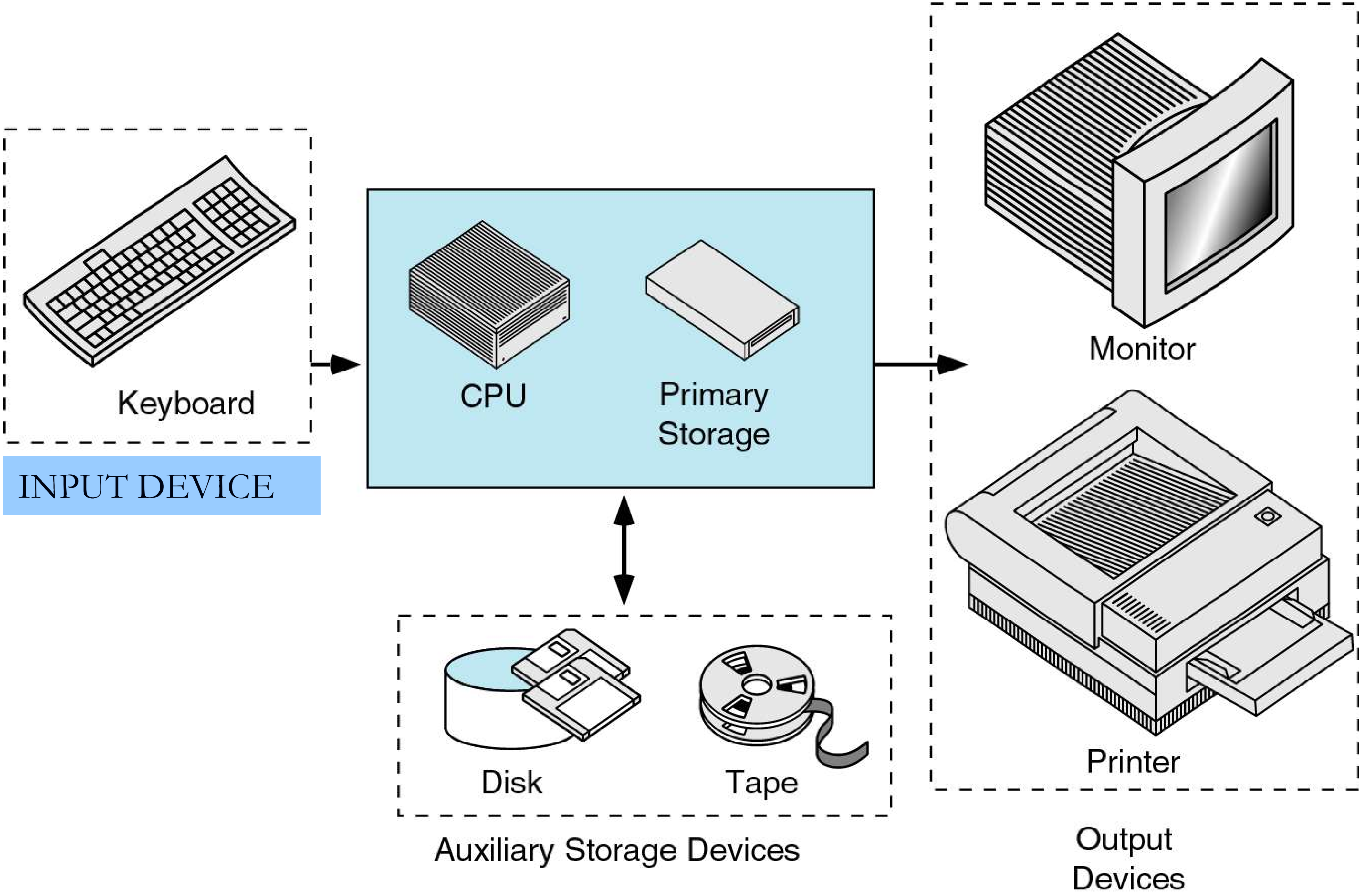
SOFTWARE

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**



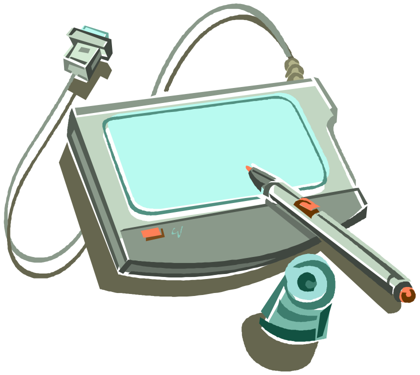
INPUT DEVICE

Hardware: Physical Component

Input Device

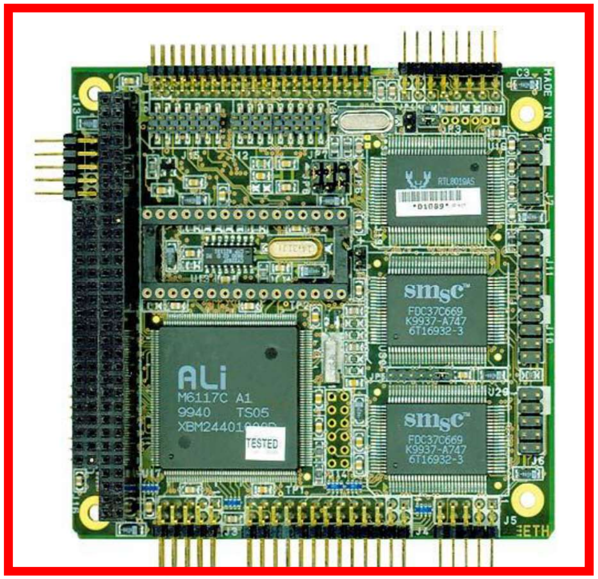
Through which programs and data are entered.

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



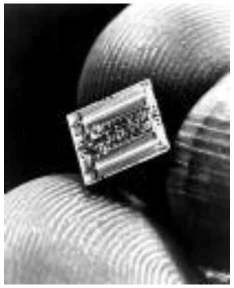
Central Processing Unit (CPU)

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.

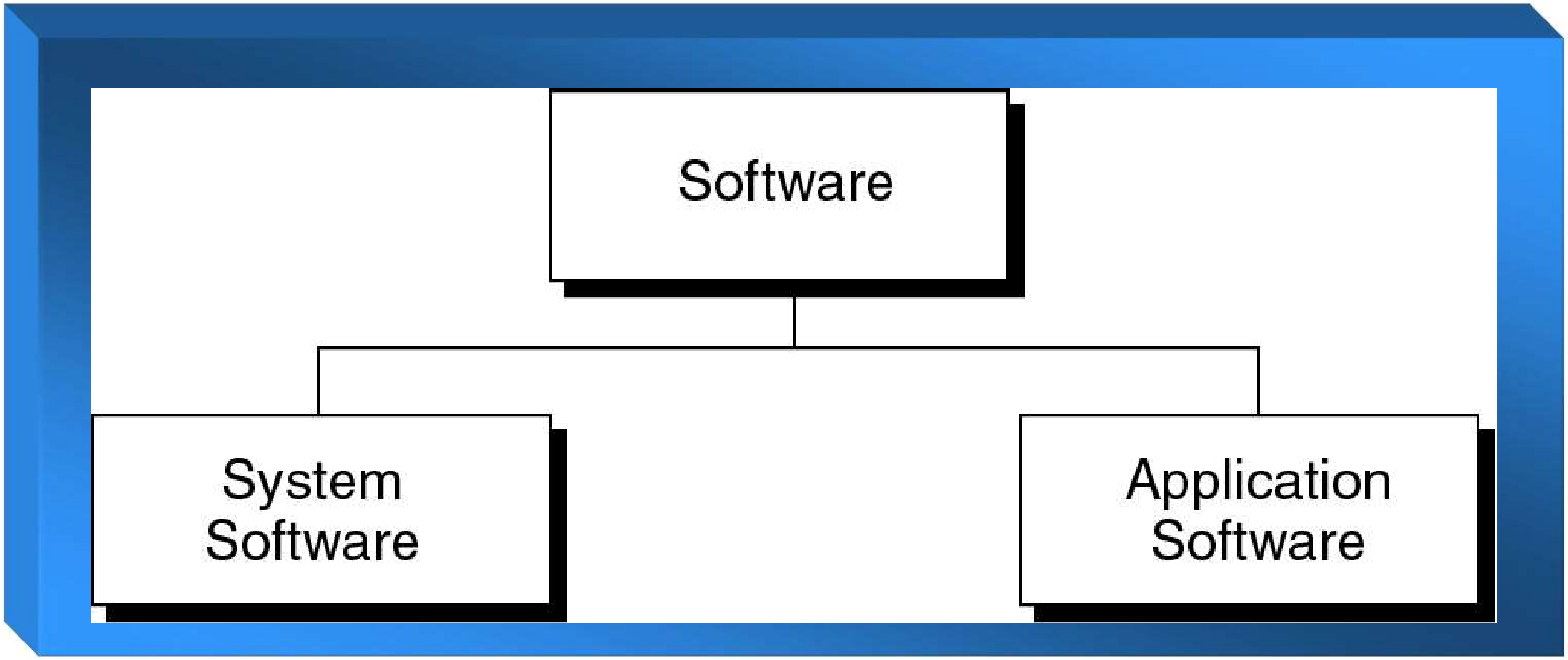
OUTPUT DEVICE

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.

AUXILIARY STORAGE

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

SOFTWARE



System Software

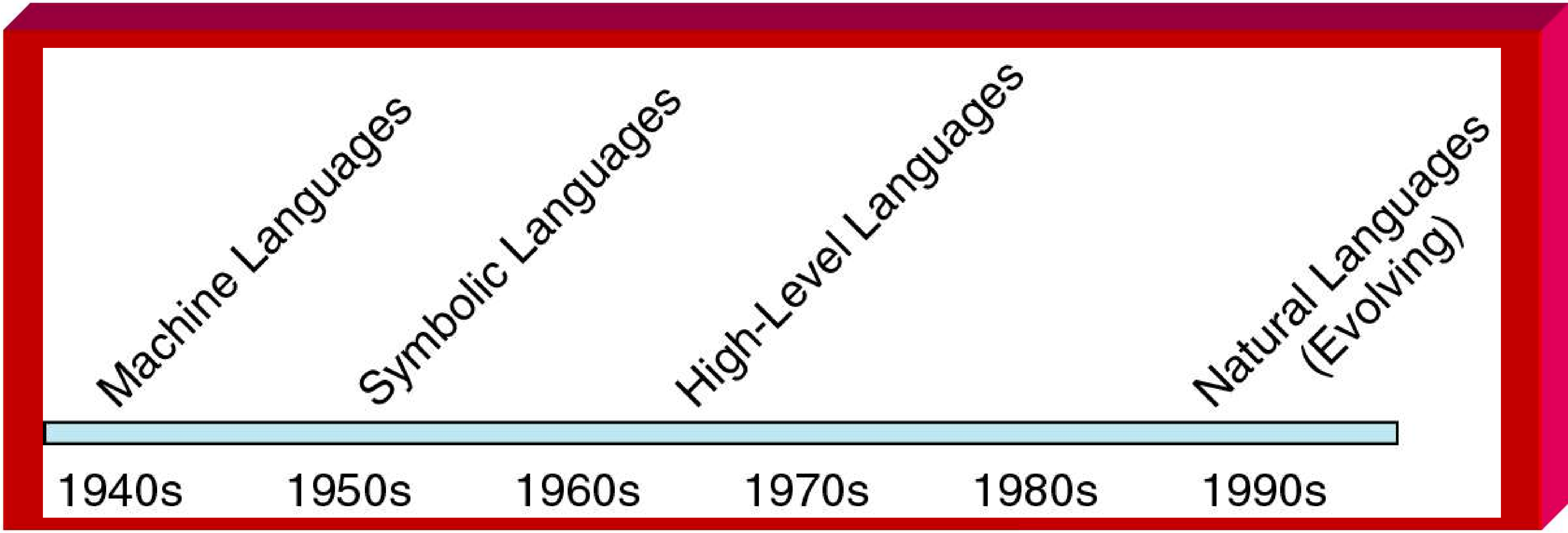
It manages the computer resources. It provides an interface between the hardware and the users but does nothing to directly serve the users.

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



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

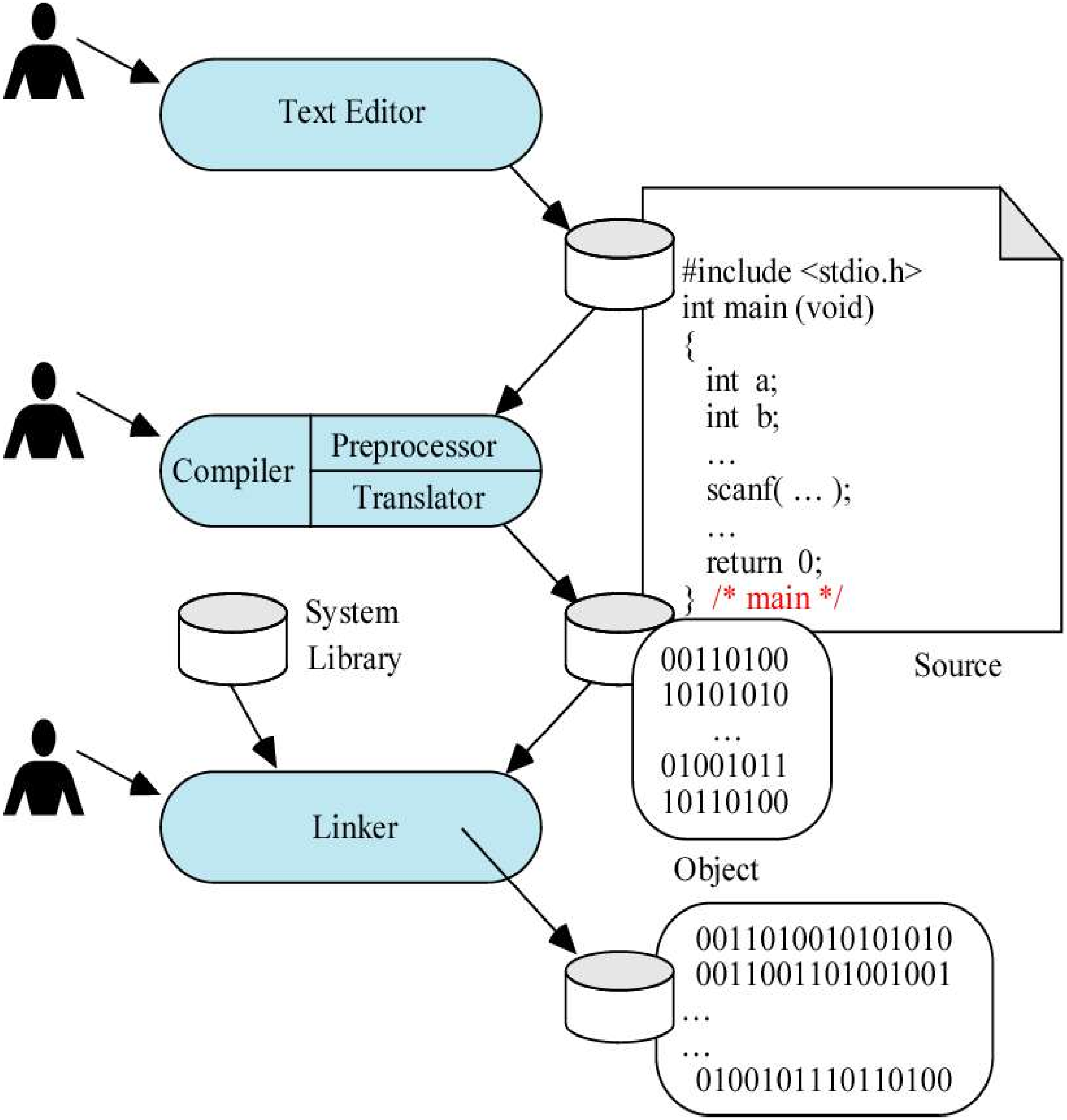
Compiler is a program that converts source program into machine language.
A C compiler contains two separate programs as : preprocessor and translator.

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.

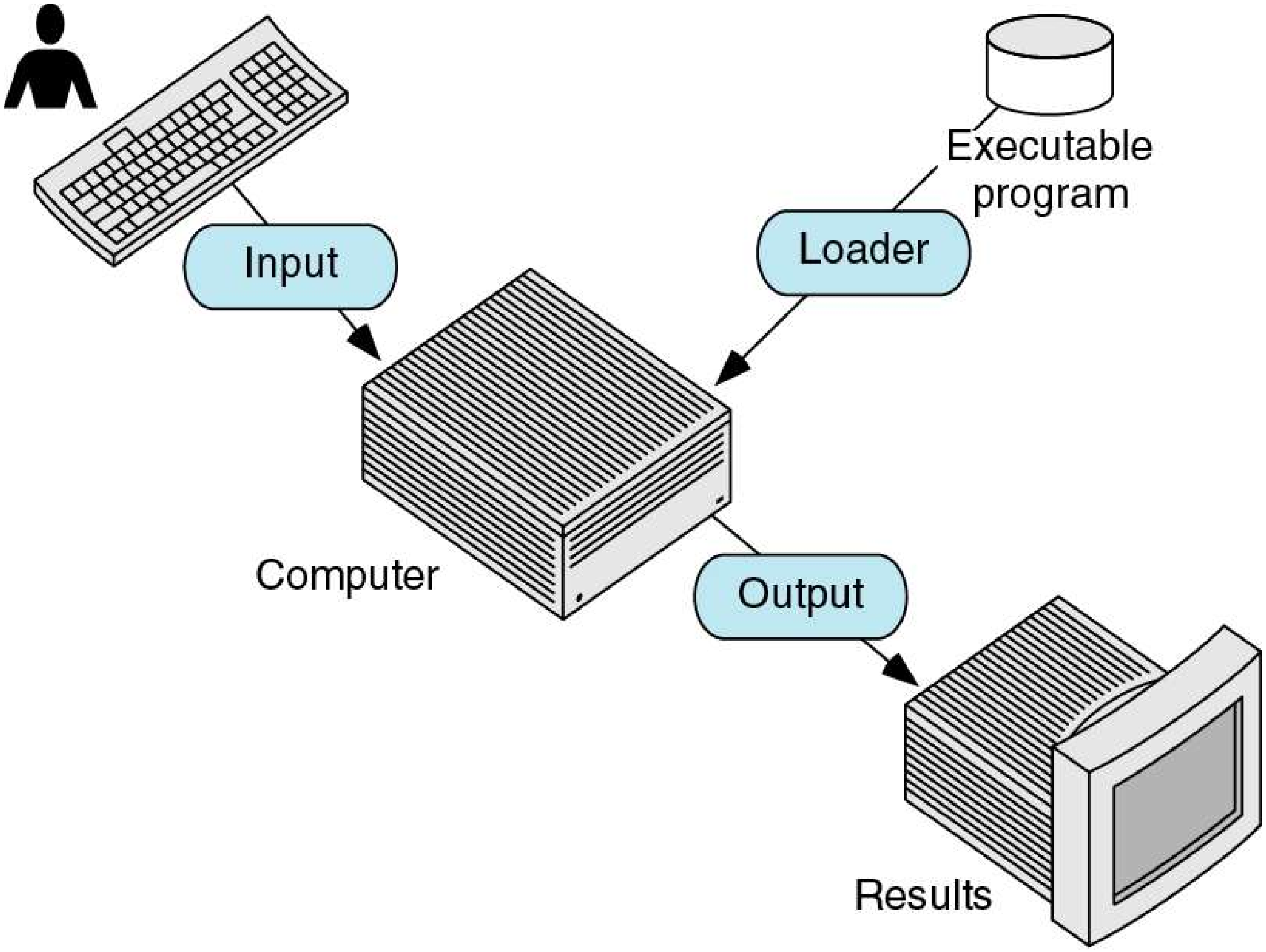


```
#include <stdio.h>
int main (void)
{
   int  a;
   int  b;
   …
   scanf( … );
   …
   return  0;
}  /* main */
```

```
00110100
10101010
   …
01001011
10110100
```

```
 0011010010101010
 0011001101001001
…
…
 0100101110110100
```

PROGRAM EXECUTION



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.

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

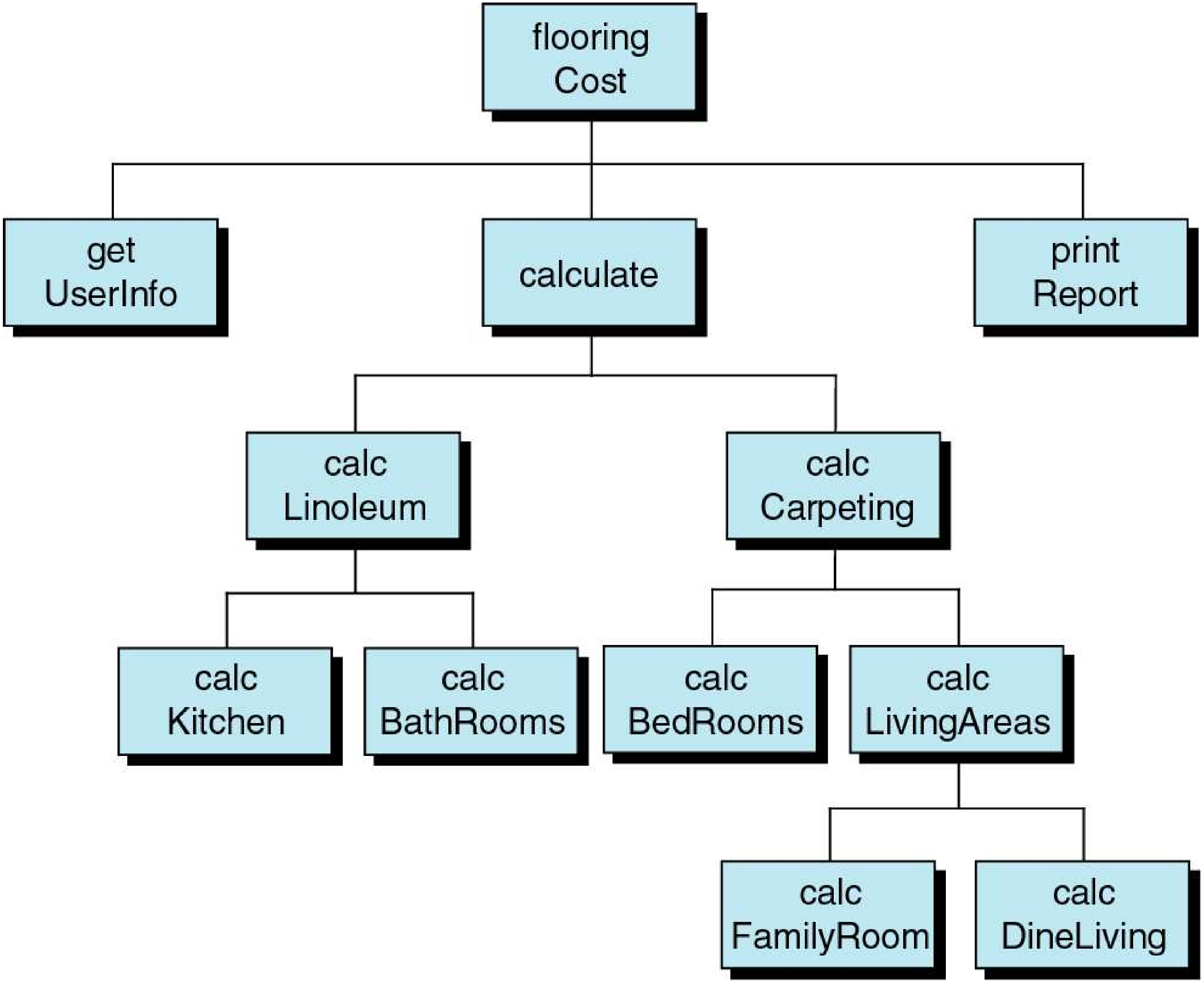
MONITOR IMPLEMENTATION OF THE PLAN

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.


flooring Cost
get UserInfo
calculate
print Report
calc Linoleum
calc Carpeting
calc Kitchen
calc BathRooms
calc BedRooms
calc LivingAreas
calc FamilyRoom
calc DineLiving

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.

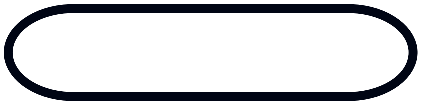
Example:

```
input A
input B
C = A + B
print C
```

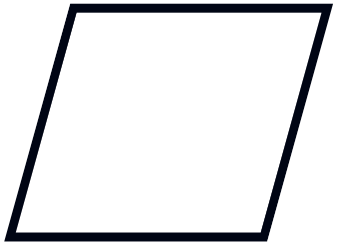
FLOW CHART

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

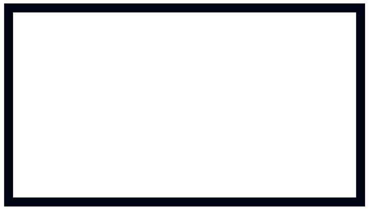
A flowchart is a diagrammatic representation that illustrates the sequence of operations to be performed to get the solution of a problem. Flowcharts are generally drawn in the early stages of formulating computer solutions. Flowcharts facilitate communication between programmers and business people. 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.



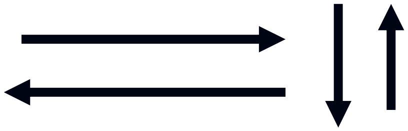
Terminal



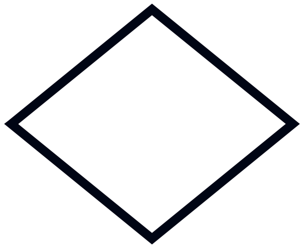
Input/ Output



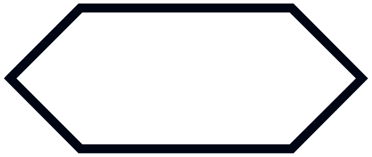
Process



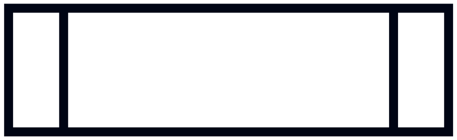
Flow Lines



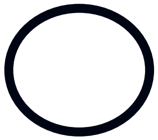
Decision



Loop

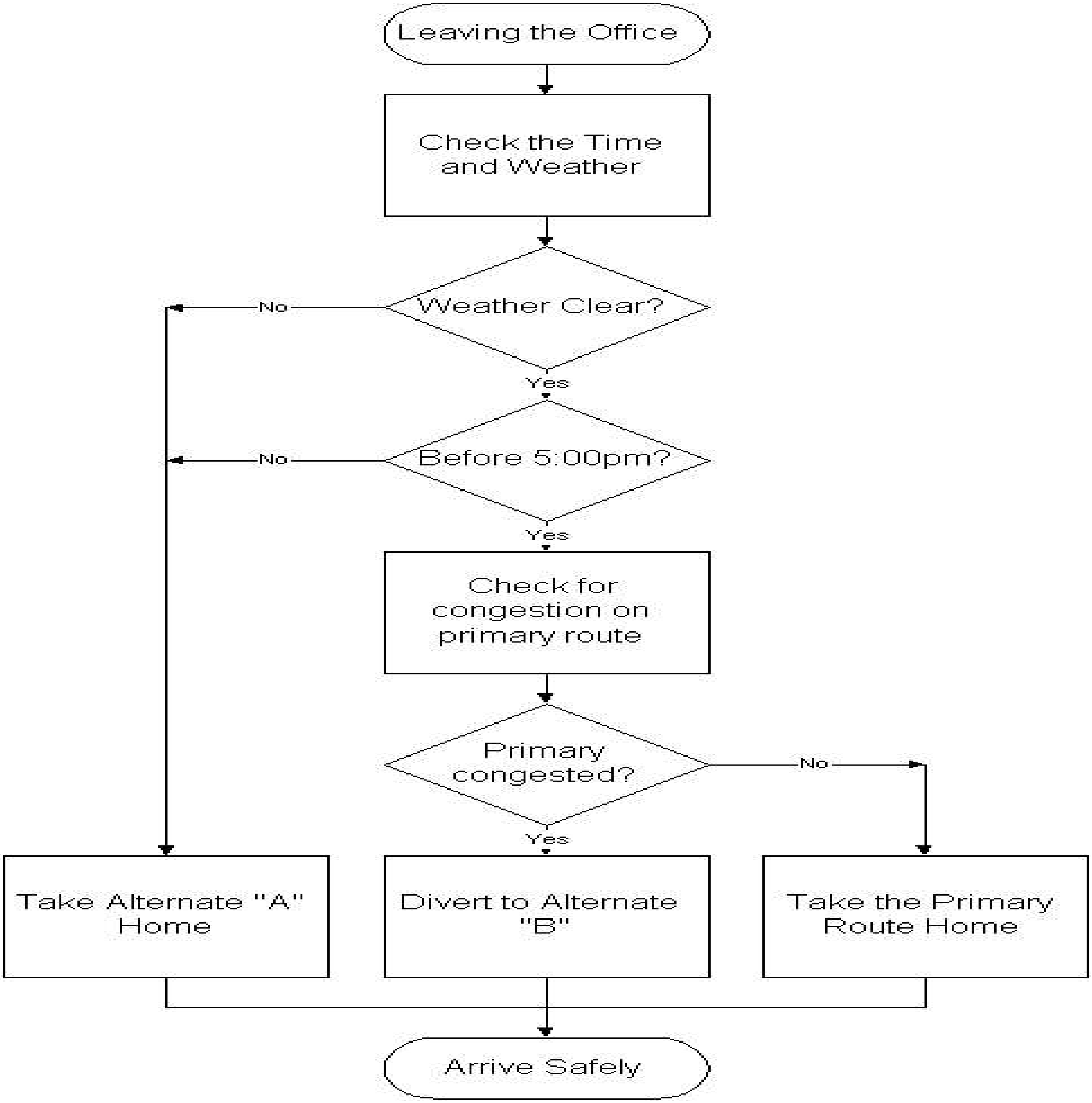


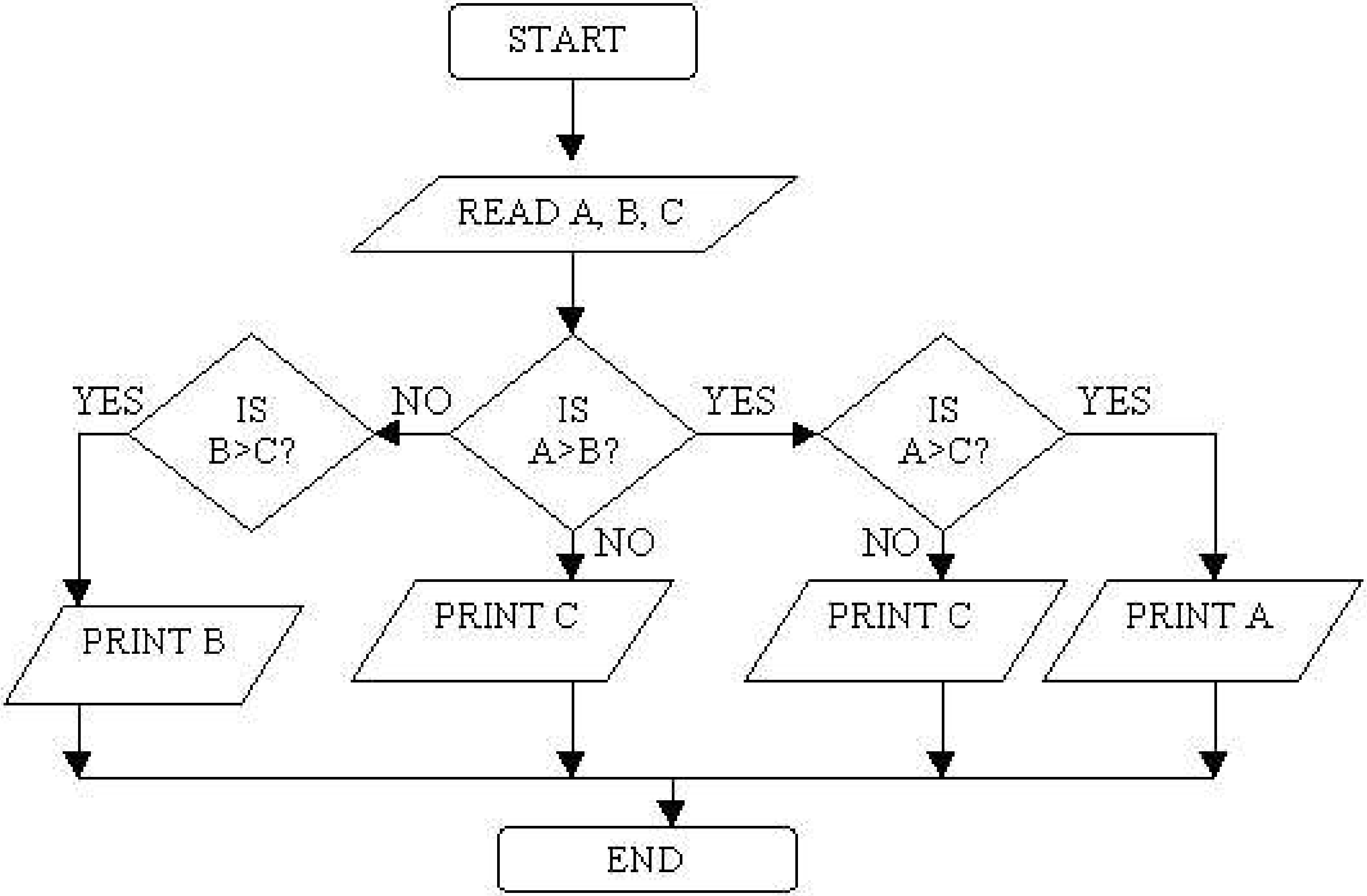
Predefined Process

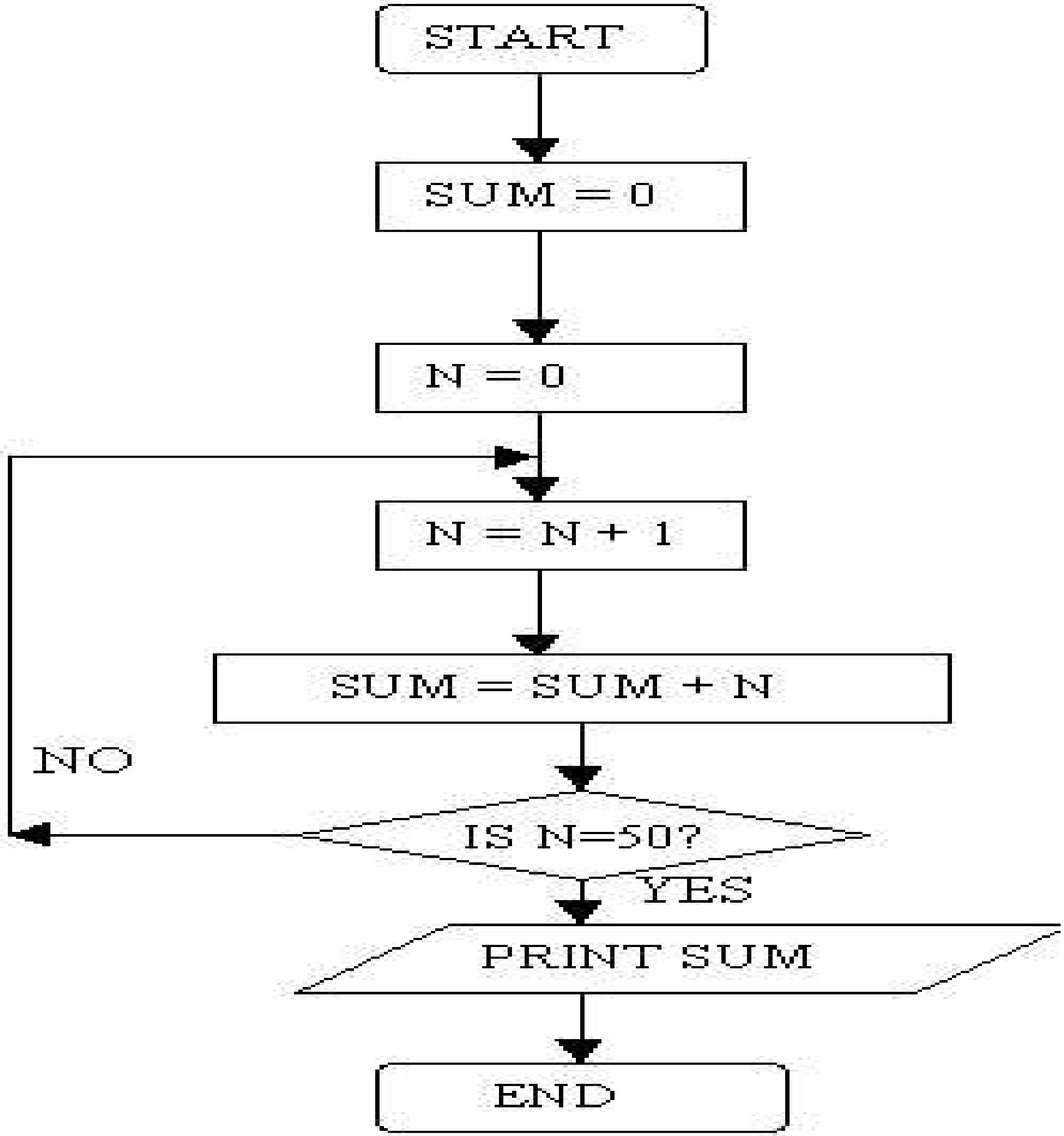


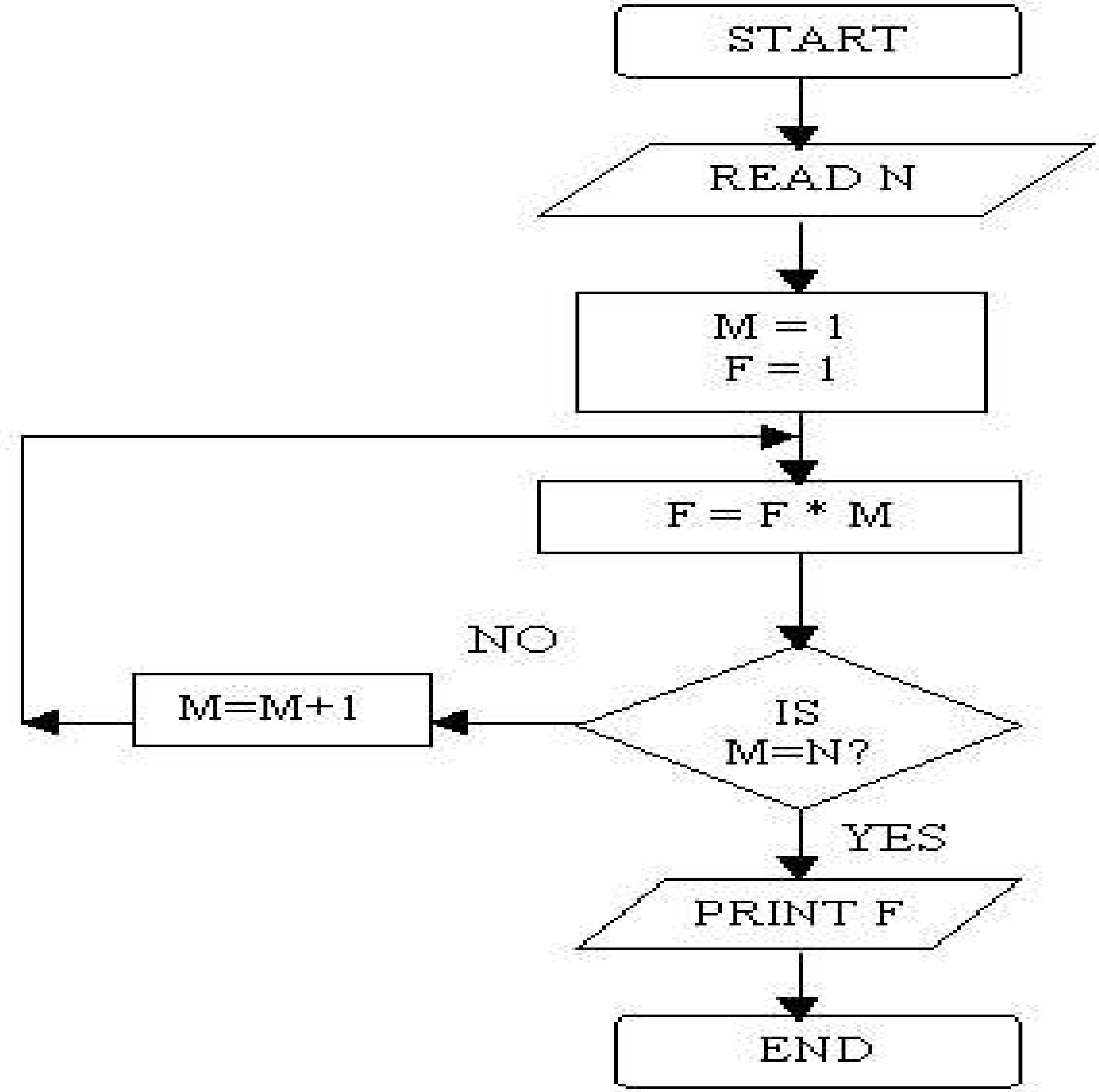
Connector

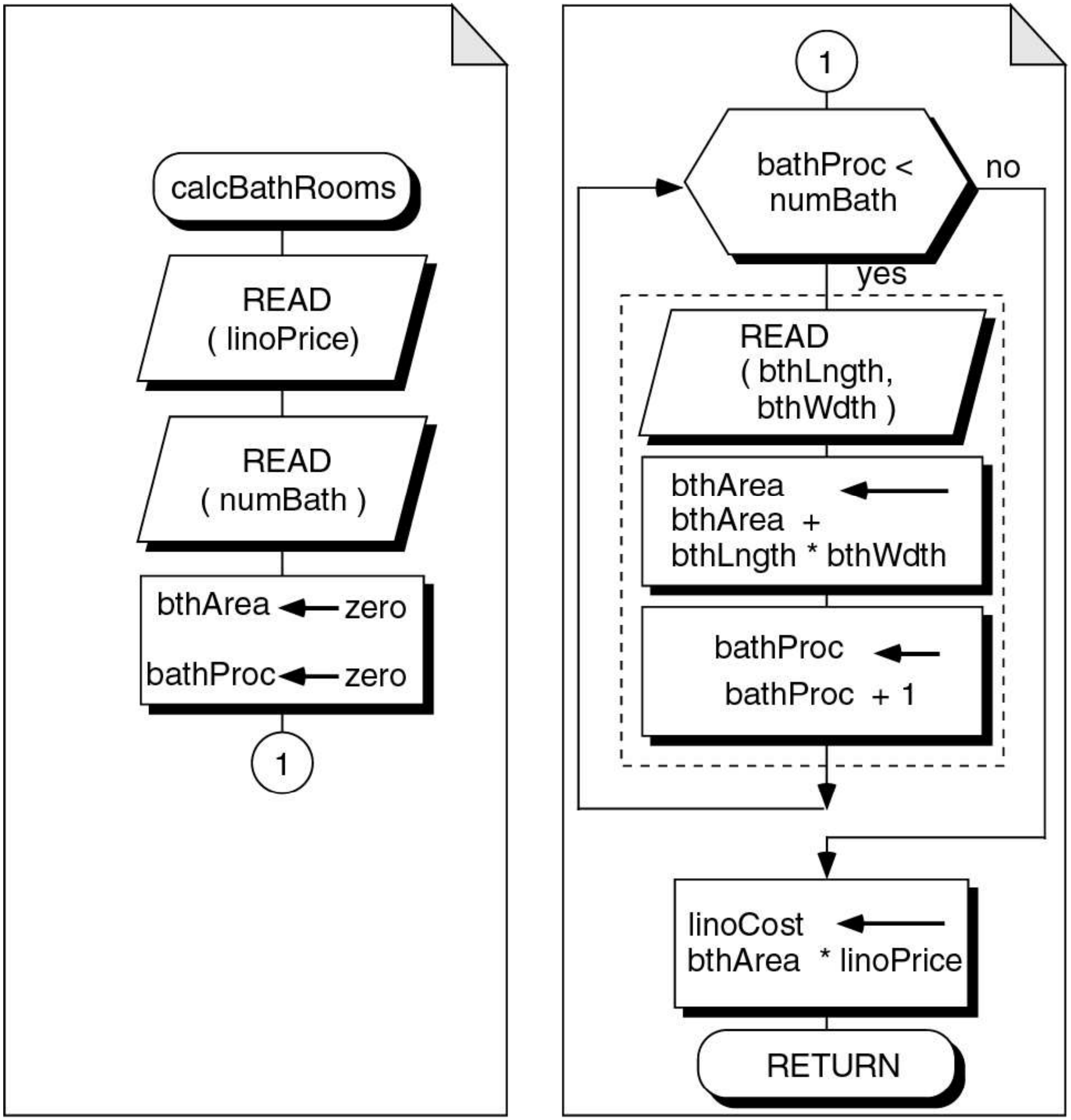
The Best Way Home




START
READ A, B, C
IS A>B?
YES
NO
IS B>C?
YES
NO
IS A>C?
YES
NO
PRINT B
PRINT C
PRINT C
PRINT A
END


START
SUM = 0
N = 0
N = N + 1
SUM = SUM + N
NO
IS N=50?
YES
PRINT SUM
END


START
READ N
M = 1
F = 1
F = F * M
NO
M=M+1
IS
M=N?
YES
PRINT F
END


calcBathRooms
READ
(linoPrice)
READ
(numBath)
bthArea ← zero
bathProc ← zero
1
1
bathProc <
numBath
no
yes
READ
(bthLngth,
bthWdth)
bthArea ←
bthArea +
bthLngth * bthWdth
bathProc ←
bathProc + 1
linoCost ←
bthArea * linoPrice
RETURN

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

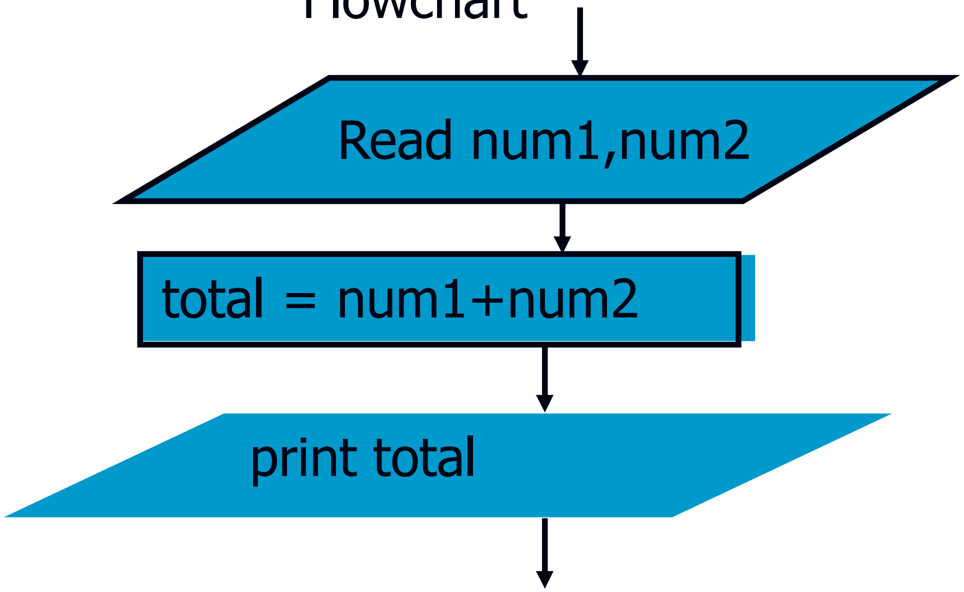
Sequence structure

- 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

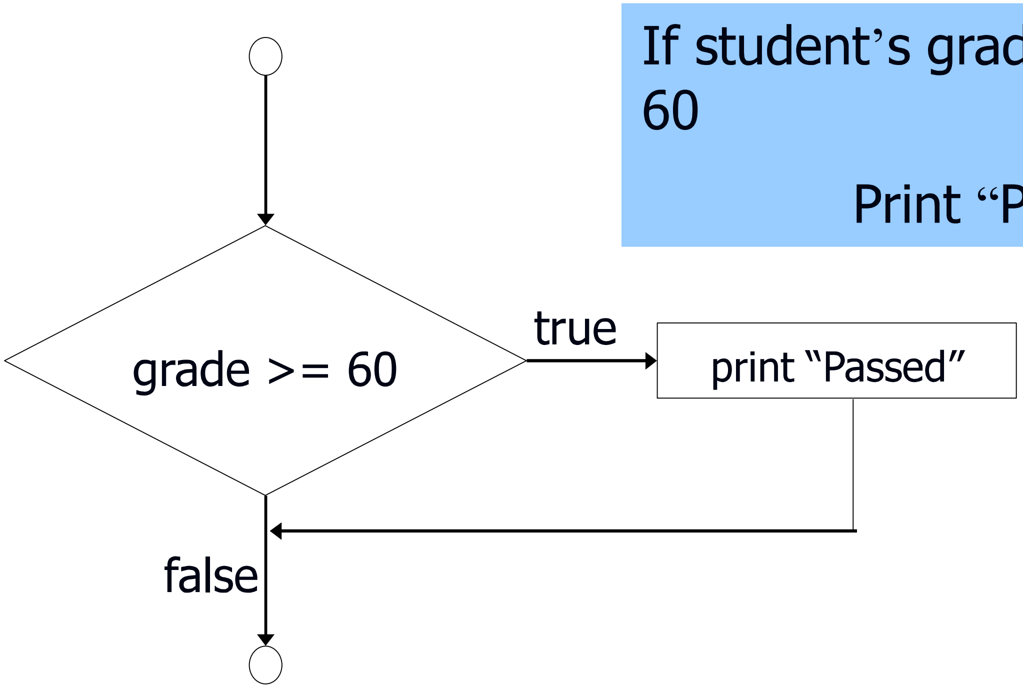
- Used to choose among alternative courses of action
- C has three types: **if**, **if**..**else**, and **switch**

The if selection structure

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

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

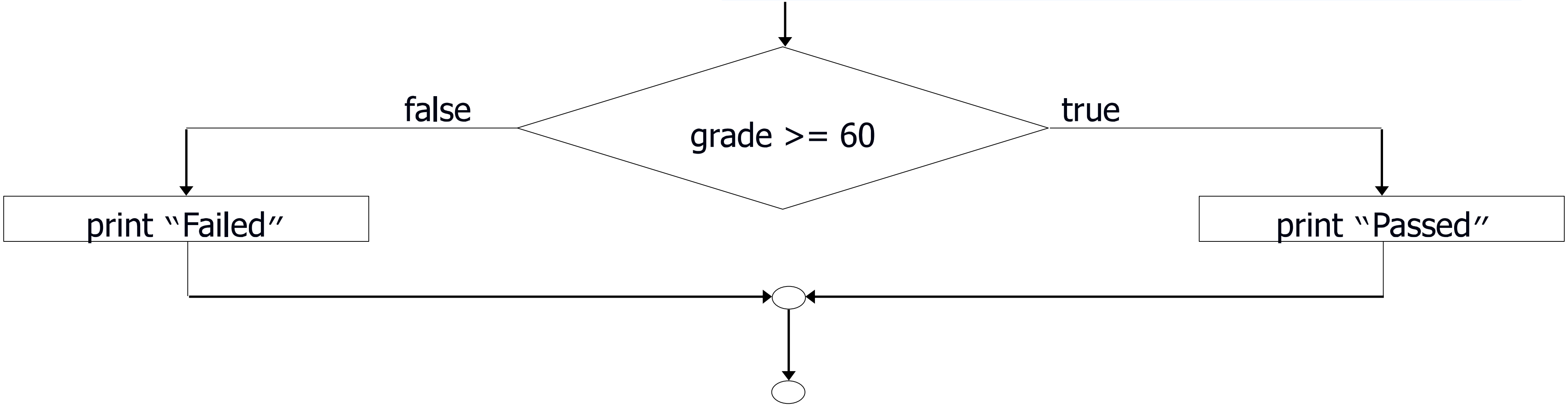
Print “Passed”



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"
```



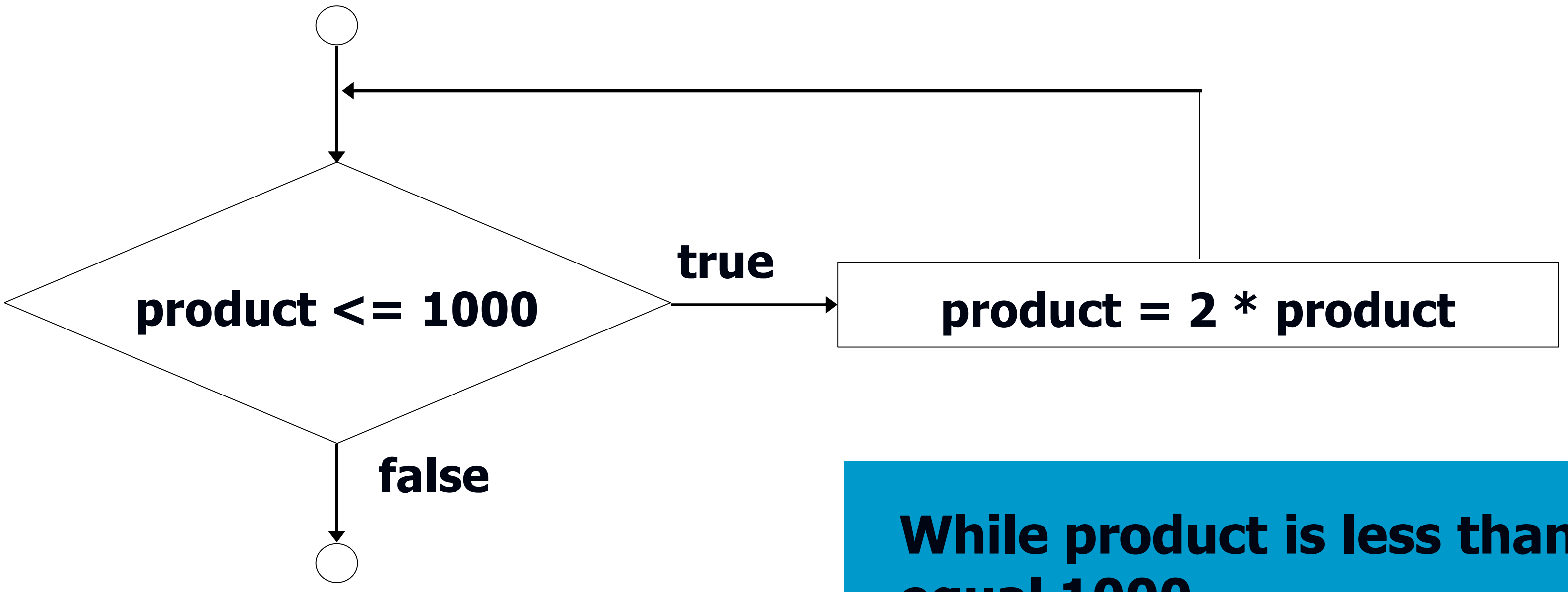
Repetition structure

- 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



While product is less than or equal 1000

calculate

product=2 * product

END OF LECTURE-1

EdTech
QUESTIONS?
PROBLEMS?
NEED DIRECTIONS?
ASK HERE!!!

FAQ

THANK YOU

- **List of Experiments**

1. **C Programming Environment.**
2. **Logical and repetitive structure statements.**
3. **Functions and Pointers.**
4. **Structure and File handling methods.**

Click to return