Lecture Two: Flowchart definition and its symbols

1- Flowcharting Definition:

It is a method of documenting (charting) the flow (or paths) that a program would execute. There are four main categories of control structures:

a. Sequence: Simply do one instruction then the next and the next. Just do them in a given sequence or in order list.

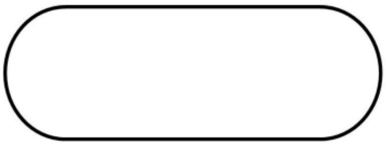
b. Selection: This is where you select or choose between two or more flows. The choice is decided by asking some sort of question. The answer determines the path (or which lines of code) will be executed.

c. Iteration: also known as repetition, it allows some code (one or many lines) to be executed (or repeated) several times. The code might not be executed at all (repeat zero times), executed a fixed number of times or executed indefinitely until some conditions has been met. Also known as looping because the flowcharting shows the flow looping back to repeat the task.

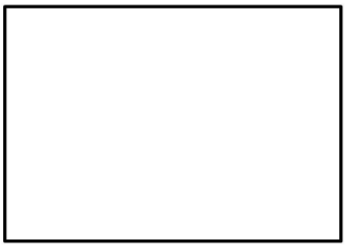
d. Branching: a control structure that allows the flow of execution to jump to a different part of the program.

2- Flowchart Symbols

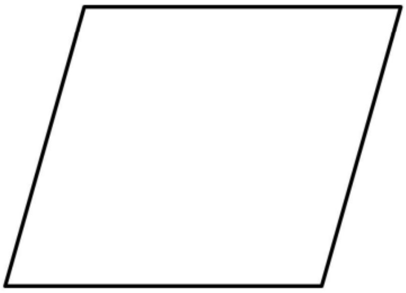
- **Terminal:** the rounded rectangle, or terminal points, indicate the flowchart’s starting and ending points.



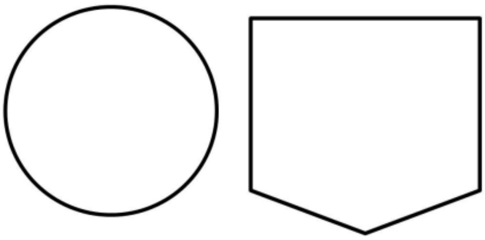
- **Process:** the rectangle depicts a process such as a mathematical computation, or a variable assignment.



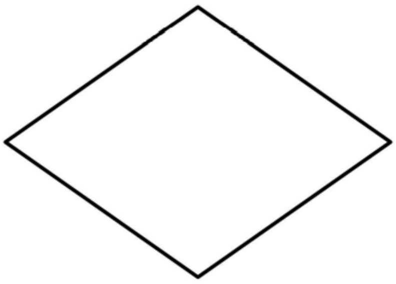
- **Input/Output:** the parallelograms designate input or output operation, in C++ language equivalent to cin or cout.



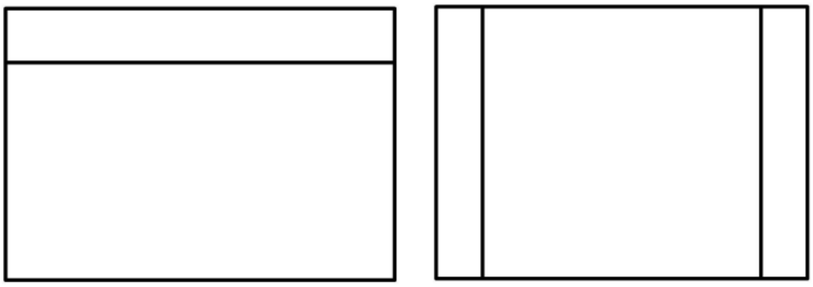
- **Connectors:** sometimes a flowchart is broken into two or more smaller flowcharts. This is usually done when a flowchart does not fit on a single page, or must be divided into sections. A connector symbol, which is a small circle whit a letter or number inside it, allows you to connect two flowcharts on the same page. A connector symbol that looks like a pocket on a shirt, allows you to connect to a flowchart on a different page.



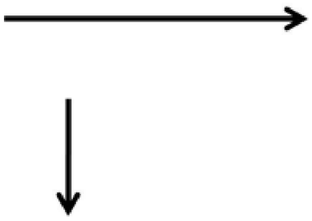
- **Decision:** the diamond is used to represent the true/false statement being tested n a decision symbol**.**



- **Module Call:** a program module is represented in a flowchart by rectangle with some lines to distinguish it from process symbol. Often programmers will makes a distinction between program control and specific task modules as shown below.



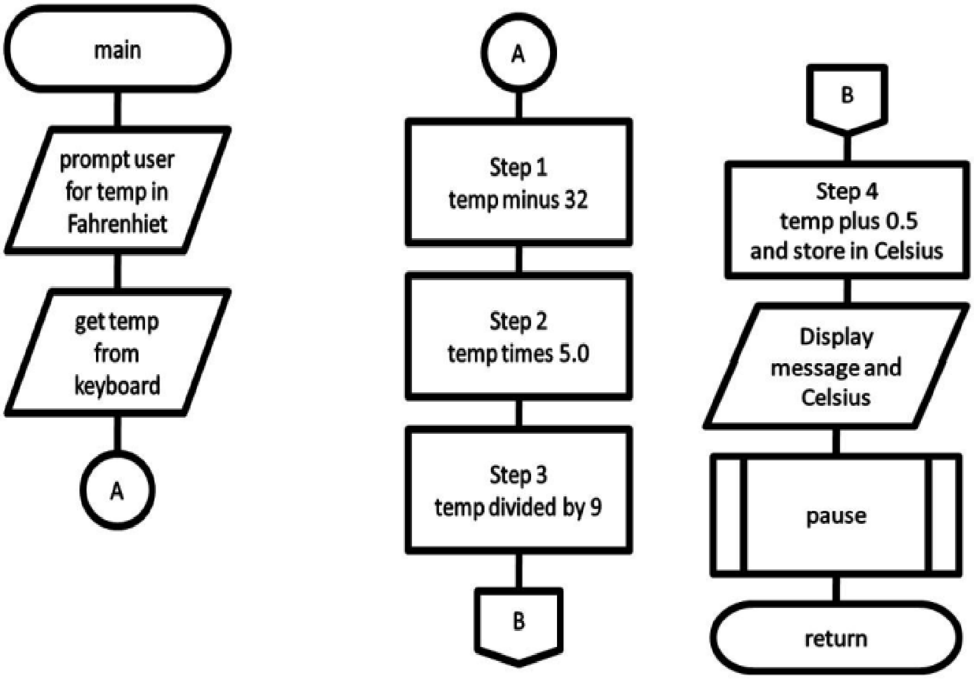
- **Flow lines:** the default flow is left to right and top to bottom (the same way you read English)**.** To save time arrowheads are often only drawn when the flow lines go contrary the normal.



3- Pseudocode & Flowcharts Example

The next item is pseudocode for a simple temperature conversion program. This demonstrates the use of both the on-page connectors. It also illustrates the sequence control structure where nothing usually happens. Just do one instruction after another in the sequence listed.

```
Input
        Display a message asking user for the temperature in Fahrenheit.
        Get the temperature from the keyboard.
Processing
        Calculate the Celsius by adding subtracting 32 from the Fahrenheit
        Temperature then multiply the result by 5 then.
        Divide the result by 9. Round up or down to the whole number.
Output
        Display the Celsius with an appropriate message.
```



Developing Test Data: Test data consists of the user providing some input values and predicting the outputs. This can be quite easy for a simple program and the data can be used twice.

1. To check the model to see if it produces the correct results (model checking).
2. To check the coded program to see if it produces the correct results (code checking)

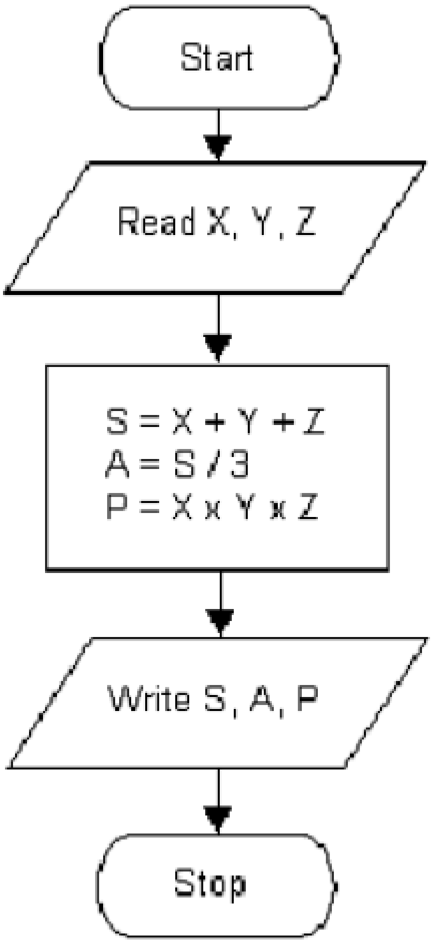
Pseudocode & Flowchart Examples

1. Write a pseudocode & design a flowchart to print the sum, average and product of 3 numbers given by the user.

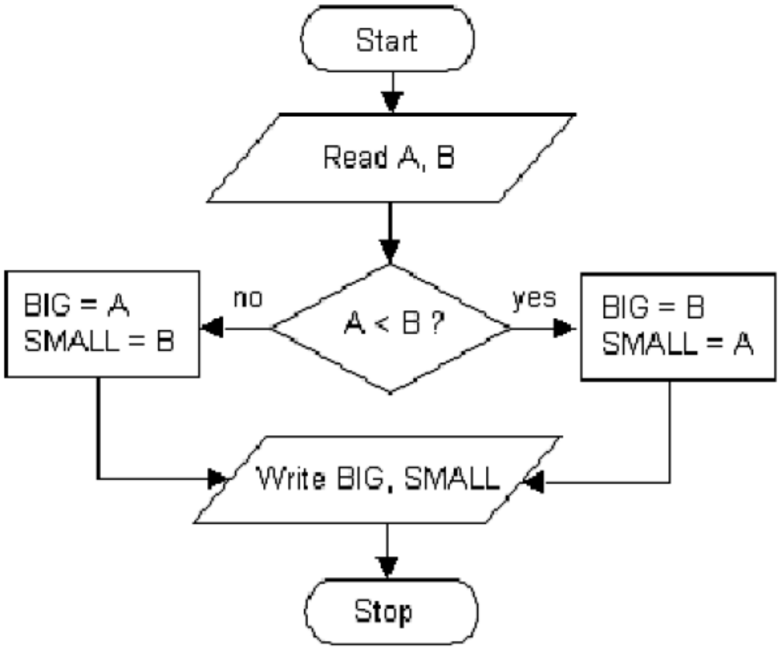
Pseudocode

```
Input
        Read X, Y, Z
Processing
        Compute Sum (S) as X+Y+Z
        Compute Average (A) as S/3
        Compute Product (P) as X*Y*Z
Output
        Write (Display) the Sum, Average, and Product.
        Pause so the user can see the answer
```

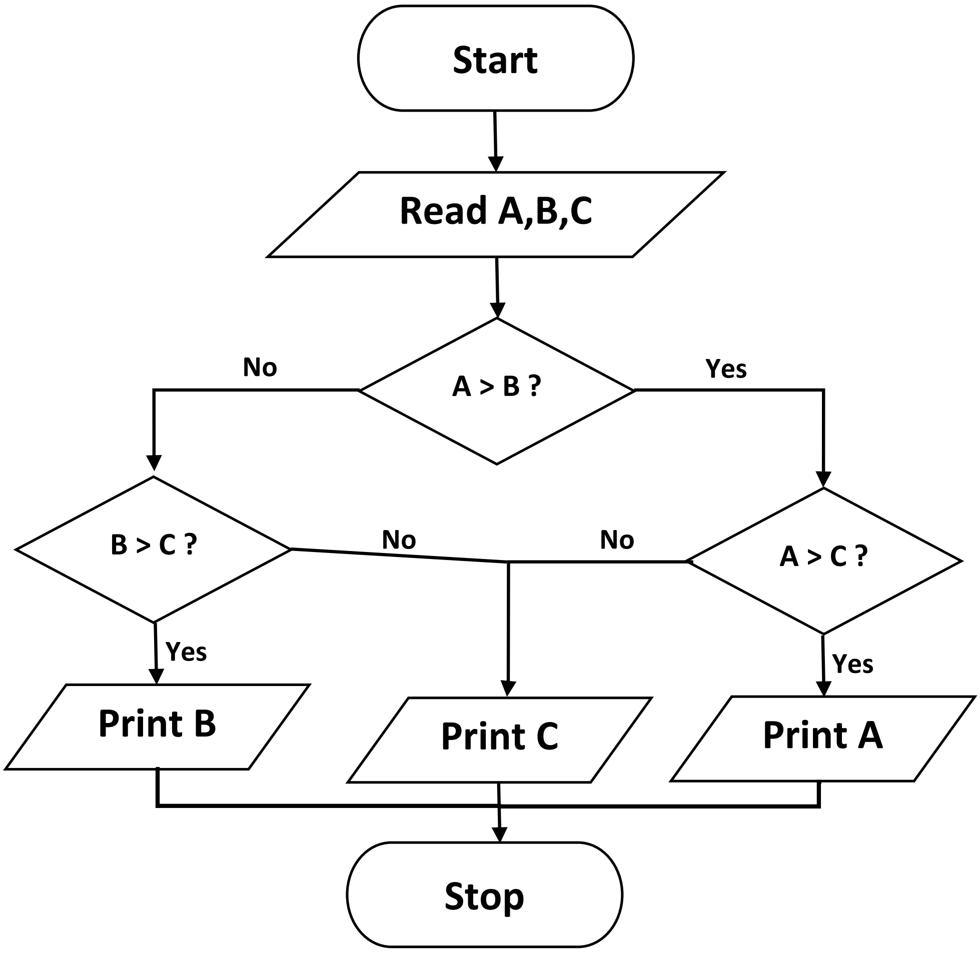
Flowchart



2. Design a flowchart to find the largest number between two non-equal numbers A and B.



3. Design a flowchart to find the largest number among three (A,B,C) non-equal numbers given by the user.


Start
Read A,B,C
A > B ?
No
Yes
B > C ?
No
No
A > C ?
Yes
Yes
Print B
Print C
Print A
Stop