Basic Programming Concepts

Some Terminologies

- **Algorithm / Flowchart / Pseudo-code**
  - A step-by-step procedure for solving a particular problem.
  - Should be independent of the programming language.
- **Program**
  - A translation of the algorithm/flowchart into a form that can be processed by a computer.
  - Typically written in a high-level language like C, C++, Java, etc.

First Look at a C Program

```
/* Program to compute the area of a circle */
#include  <stdio.h>    /* Compulsory, for library files */
main()                 /* Function heading */
{
      float  radius, area;     /* Declare variables */
      scanf ("%f", &radius);   /* Read radius */
      area = 3.14159 * radius * radius;
      printf ("Area = %f", area); /* Output to screen */
}
```

Input: 10
Output: Area = 314.158997

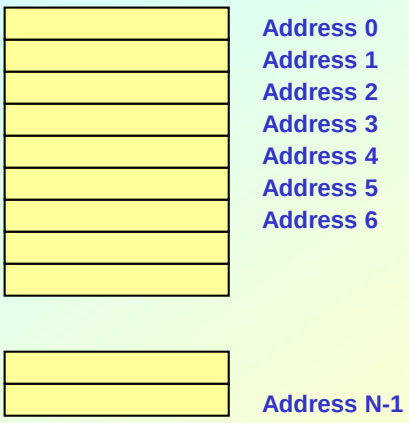
Variables and Constants

- Most important concept for problem solving using computers.
- All temporary results are stored in terms of *variables* and *constants*.
  - The value of a variable can be changed.
  - The value of a constant do not change.
- Where are they stored?
  - In main memory.

Contd.

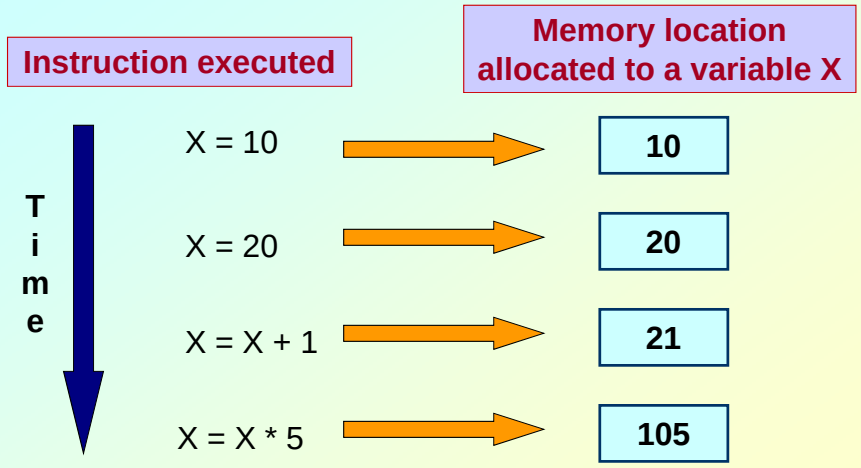
- How does memory look like (logically)?
  - As a list of storage locations, each having a unique address.
  - Variables and constants are stored in these storage locations.
  - Variable is like a *house*, and the name of a variable is like the *address* of the house.
    - Different people may reside in the house, which is like the *contents* of a variable.

Memory map

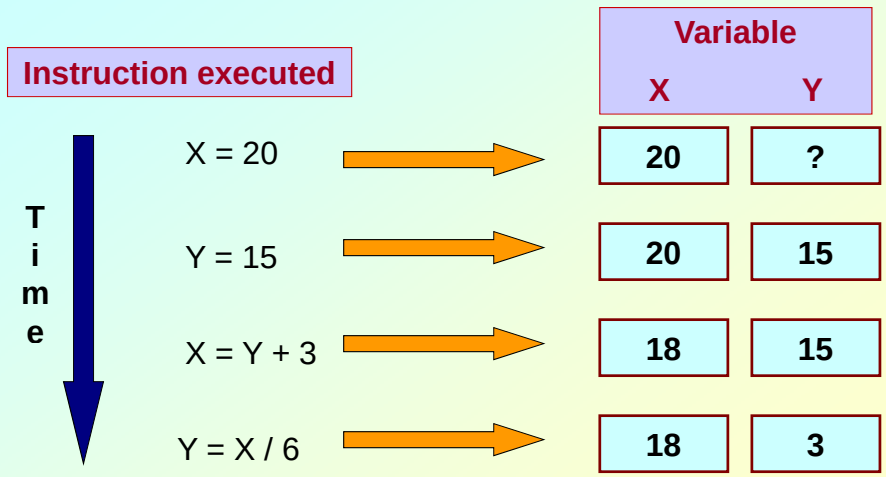


Every variable is mapped to a particular memory address

Variables in Memory



Variables in Memory (contd.)



Data types

- **Three common data types used:**
  - **Integer :: can store only whole numbers**
    Examples: 25, -56, 1, 0
  - **Floating-point :: can store numbers with fractional values.**
    Examples: 3.14159, 5.0, -12345.345, 2.65E12
  - **Character :: can store a single character**
    Examples: 'A', 'a', '*', '3', ' ', '+'

Data Types (contd.)

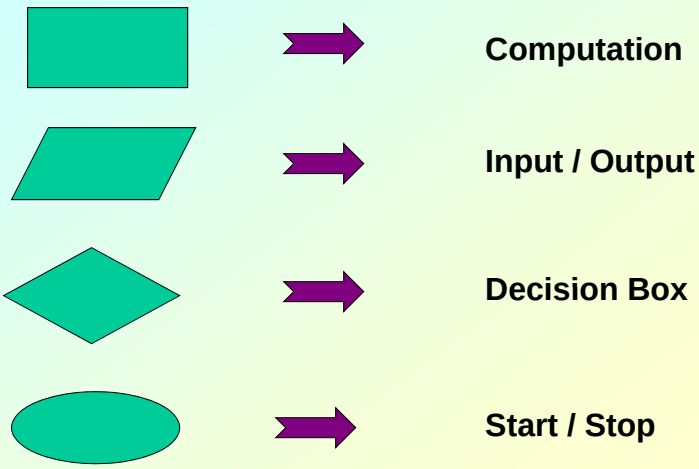
- **How are they stored in memory?**
  - **Integer ::**
    - **16 bits**
    - **32 bits**
  - **Float ::**
    - **32 bits**
    - **64 bits**
  - **Char ::**
    - **8 bits (ASCII code)**
    - **16 bits (UNICODE, used in Java)**

Actual number of bits varies from one computer to another

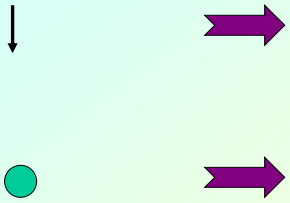
Problem solving (Typical Flow)

- **Step 1:**
  - Clearly specify the problem to be solved.
- **Step 2:**
  - Draw flowchart or write algorithm.
- **Step 3:**
  - Convert flowchart (algorithm) into program code.
- **Step 4:**
  - Compile the program into object code.
- **Step 5:**
  - Execute the program.

Flowchart: basic symbols



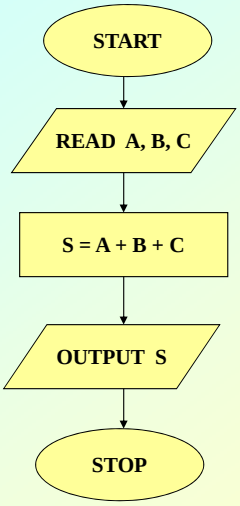
Contd.



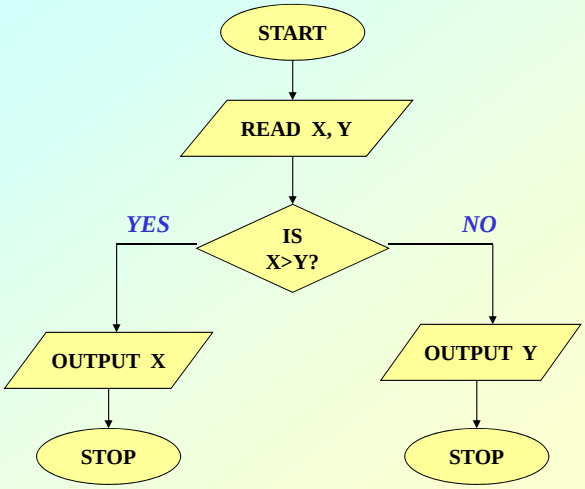
Flow of control

Connector

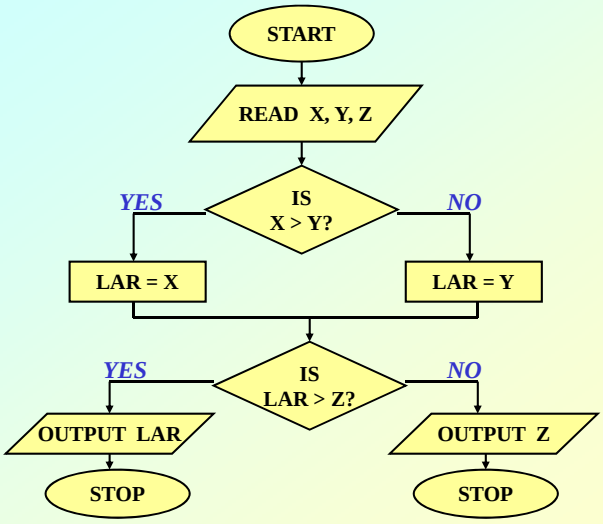
Example 1: *Adding three numbers*



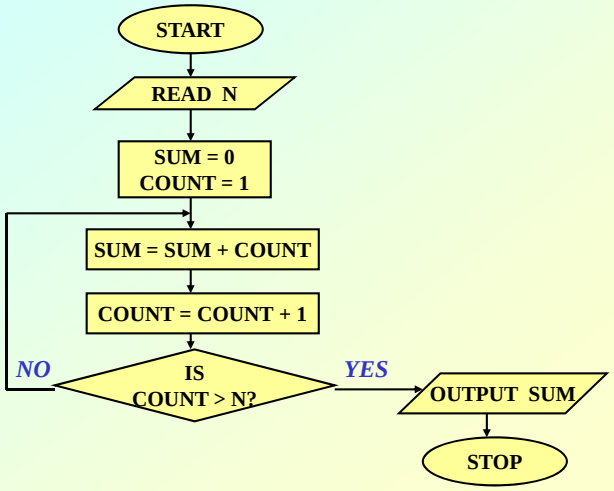
Example 2: *Larger of two numbers*



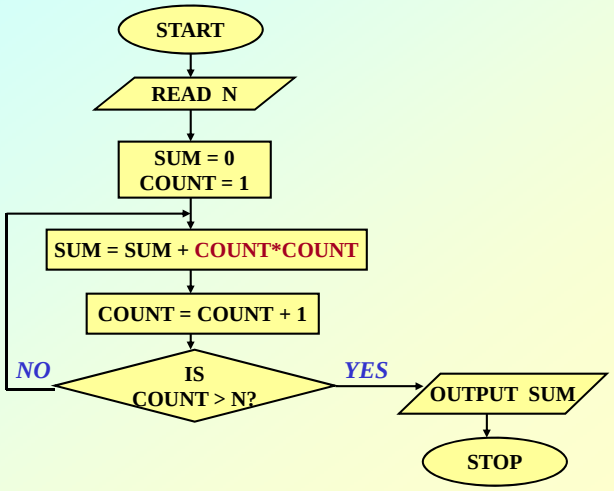
Example 3: *Largest of three numbers*



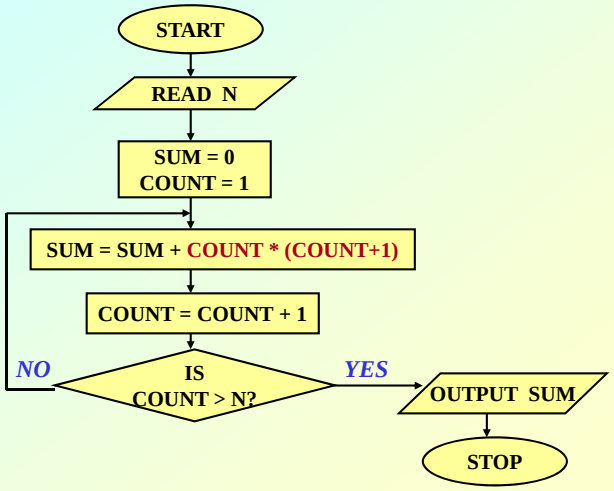
Example 4: *Sum of first N natural numbers*



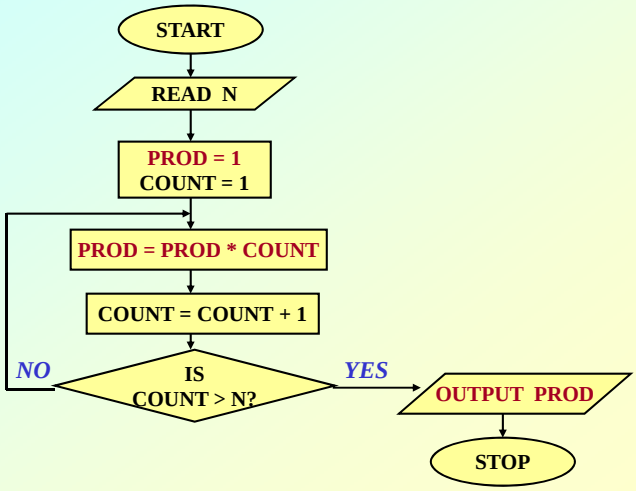
Example 5: *SUM = $1^2 + 2^2 + 3^2 + N^2$*



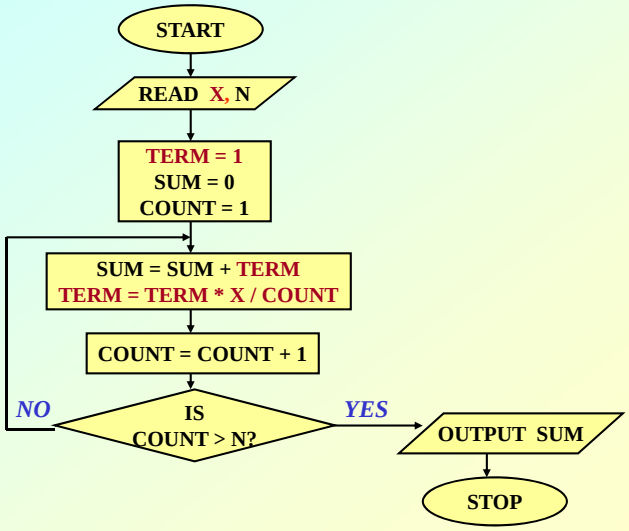
Example 6: *SUM = 1.2 + 2.3 + 3.4 + to N terms*



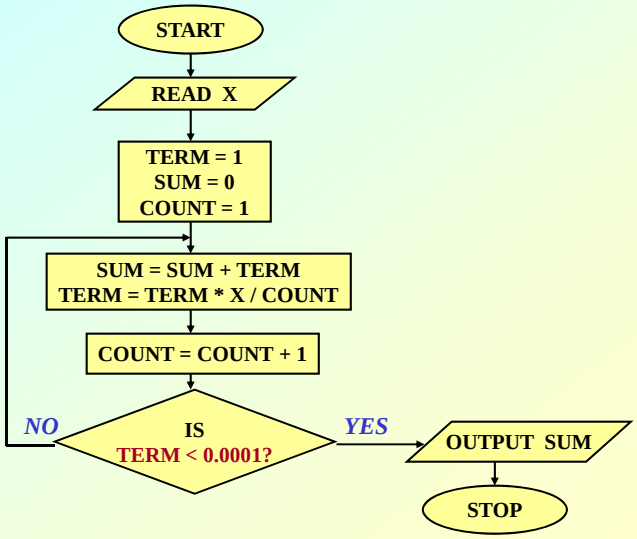
Example 7: *Computing Factorial*



Example 8: *Computing e^x series up to N terms*



Example 9: *Computing e^x series up to 4 decimal places*



Example 10: *Roots of a quadratic equation*

$$ax^2 + bx + c = 0$$

TRY YOURSELF

Example 11: *Grade computation*

MARKS ≥ 90 ➔ Ex
89 ≥ MARKS ≥ 80 ➔ A
79 ≥ MARKS ≥ 70 ➔ B
69 ≥ MARKS ≥ 60 ➔ C
59 ≥ MARKS ≥ 50 ➔ D
49 ≥ MARKS ≥ 35 ➔ P
34 ≥ MARKS ➔ F

Grade Computation (contd.)