INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

CSN-103: Fundamentals of Object Oriented Programming

Instructor: Dr. Rahul Thakur

Assistant Professor, Computer Science and Engineering, IIT Roorkee

Computer as a Programmed Machine

- What computers do? How they do it?
  - *Almost* like a human being
    - A task need to be done
      - Formulate a **sequence** of steps to compete the task
      - Execute these steps
  - Not like a human being
    - Vague steps/instructions
      - Natural language is a problem
      - All possible steps are “pre-defined”: Instruction Set

Machine Language

- The only language a computer can understand
- No need for are processing/translation. Directly executed by the processor
- Machine Language→ Consists of instructions

1010 1110 0011 0101 0101 0000 1011

1000 1110 1011 0001 0101 0110 1001

1011 1010 0111 0101 1111 0100 1111

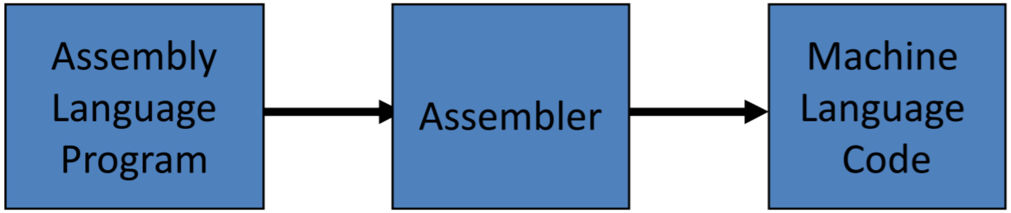
Binary String of 0s and 1s

- Machine Dependent: Each computer type (architecture) has its own machine language
  - Not Portable
- Advantage: Extremely fast
- Disadvantage: Programmer unfriendly → Programming errors

Assembly Languages

- Uses English like abbreviations such as ADD, SUB, MUL etc.
- But computers only understand Machine Language!!!
- Require a software → Assembler
  - Convert Assembly language to Machine language
- Example program:

```
MOV A, 1011
MOV B, 1100
ADD A,B
```



- Advantage: Programmer friendly
- Disadvantage: Program is still quite lengthy, slow execution, not portable

High Level Languages

- English like statements
- Much easier to learn and remember
- Need an additional program
  - Compiler: Convert high level language instructions to Machine language instructions
- Example Programming languages: BASIC, PASCAL, C++, C, JAVA, Python etc.
- Advantage: Extremely friendly to programmers
  - One high level language → Multiple machine language instructions
  - Shorter programs → Easy to debug
  - PORTABLE
- Disadvantage: Slower execution

Definition

- Algorithm: A finite sequence of unambiguous statements to solve a specific problem in finite time

- Program: An implementation of an algorithm in some programming language
  - A sequence of instructions that comply the rules of a specific programming language

- Data Structure: A data structure is a data organization, management, and storage format that enables efficient access and modification
  - Array, Linked List, Tree, Graph, Stack, Queue

Example Algorithms

- An algorithm to find the product of 2 numbers and display it on the screen

```
Step 1: Start
Step 2: Read two number $A$, $B$
Step 3: $P=A*B$
Step 4: Display $P$
Step 5 End
```

Note: Simple English statements can be considered as an Algorithm

Example Algorithms

- An algorithm to swap (exchange) two numbers

```
Step 1: Start
Step 2: Read two number $A, B$
Step 3: $temp = A$
Step 4: $A = B$
Step 5: $B = temp$
Step 6: Display $A, B$
Step 7: End
```

Flowchart

- A flowchart is a type of **diagram** that represents a workflow or process

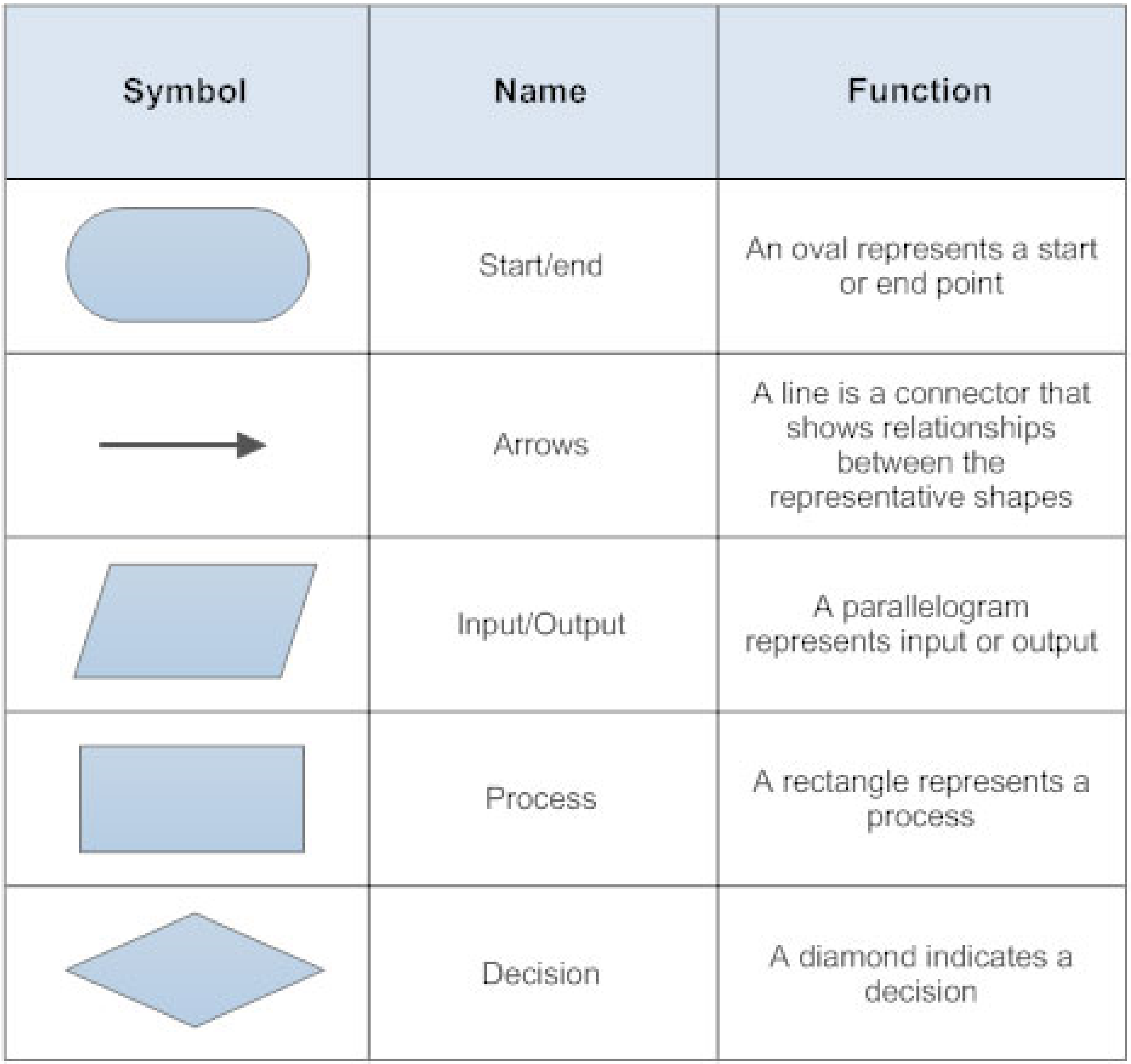
OR

Flowchart is a pictorial/graphical representation of an algorithm

- Contains different types of boxes
  - For different types of instructions
  - Boxes have standardized meanings
  - Instructions are written inside boxes

- Directional arrows indicate the exact sequence in which the instructions are to be executed

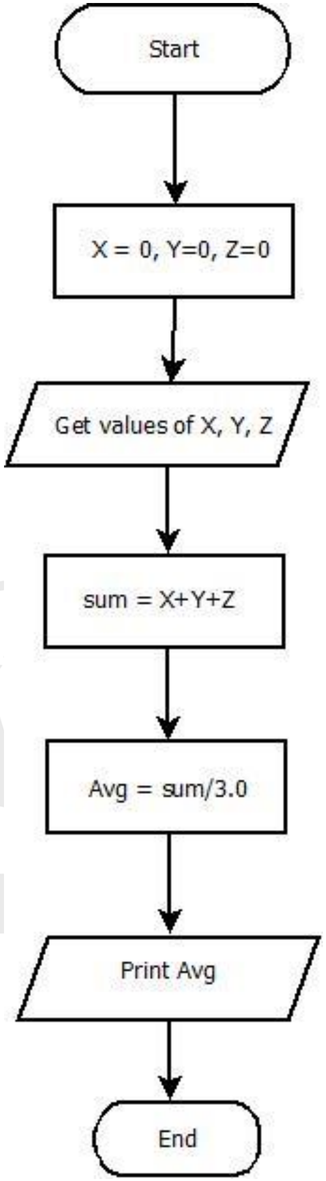
Flowchart Symbols



Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point
→	Arrows	A line is a connector that shows relationships between the representative shapes
▱	Input/Output	A parallelogram represents input or output
▭	Process	A rectangle represents a process
◇	Decision	A diamond indicates a decision

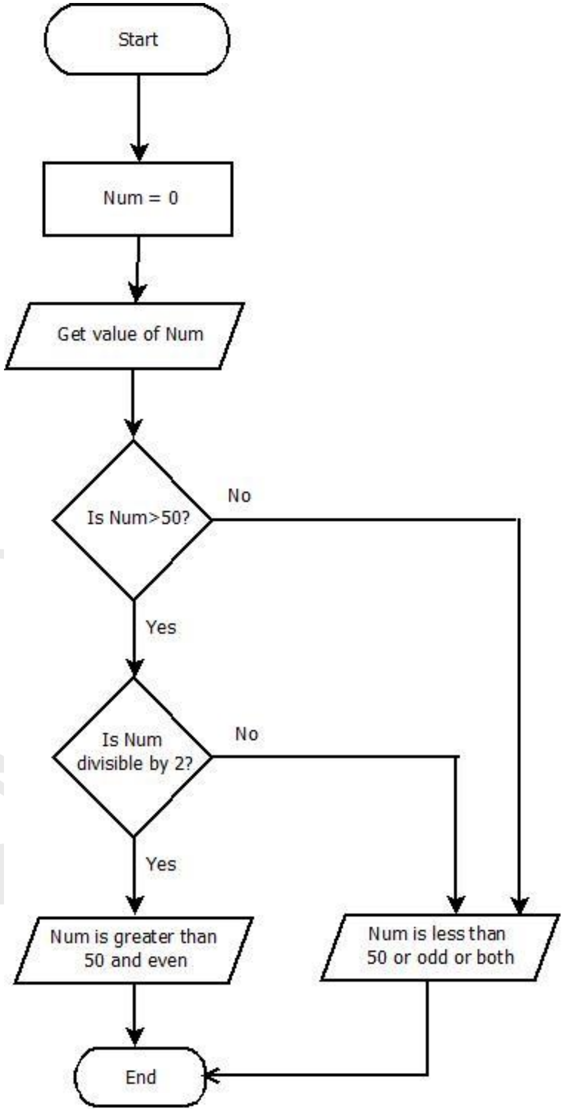
Example Flowcharts

- Finding Average of three numbers



Example Flowcharts

- Finding if a number is both greater than 50 and even.



Online Compilers

- **www.ideone.com**
- **www.tutorialspoint.com**
- **www.codechef.com**

References

- "Flowchart Symbols", SmartDraw LLC, [Online] https://www.smartdraw.com/flowchart/flowchart-symbols.htm
- Harold L Rogler, "Introduction to Computer Systems", Second Edition, August 2015