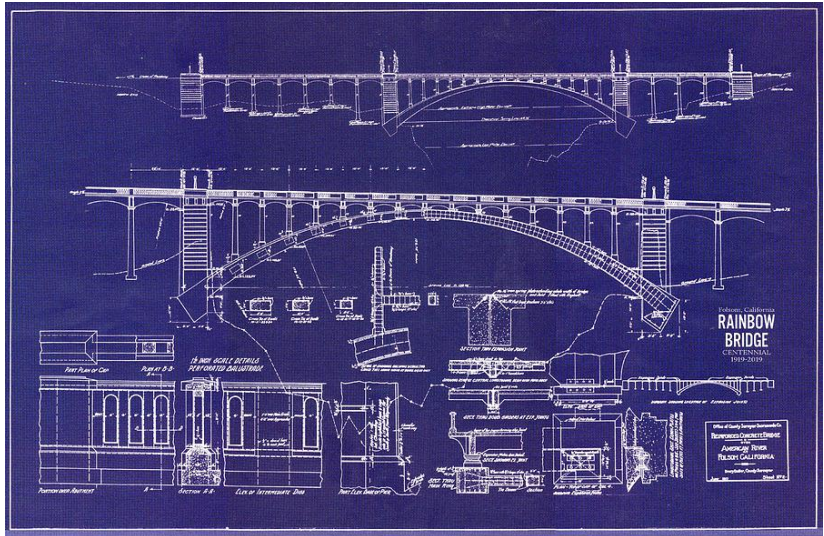


# Flowcharts

Sujit Kumar Chakrabarti

## Civil Engineering



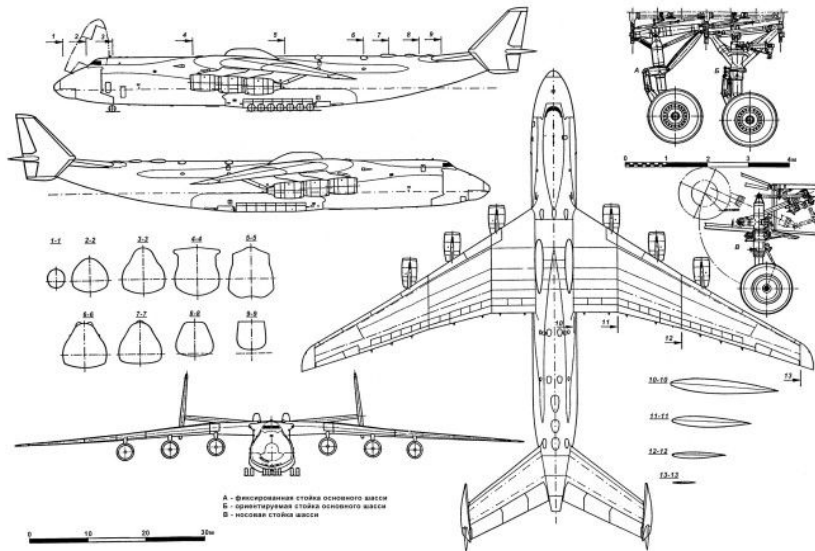
# Visual Representation

## Civil Engineering



# Visual Representation

## Aero Engineering



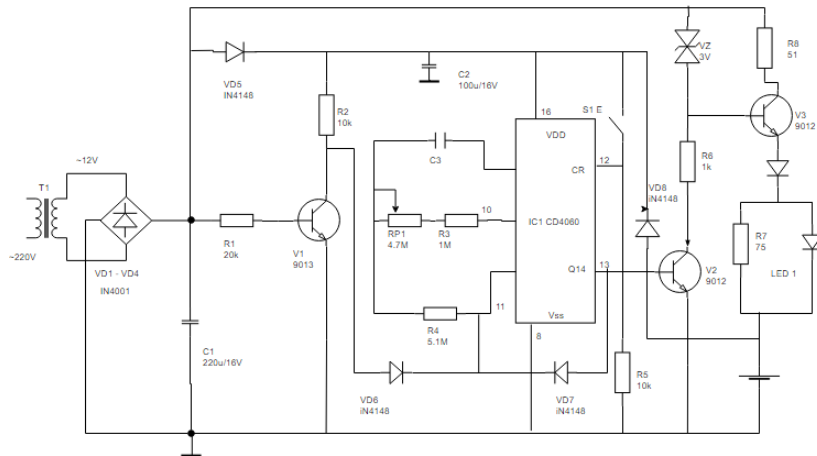
# Visual Representation

## Aero Engineering



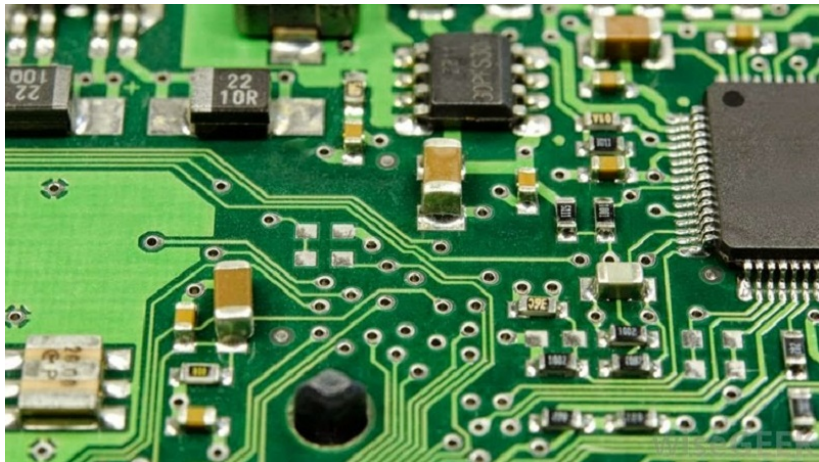
# Visual Representation

## Electrical Engineering



# Visual Representation

## Electrical Engineering



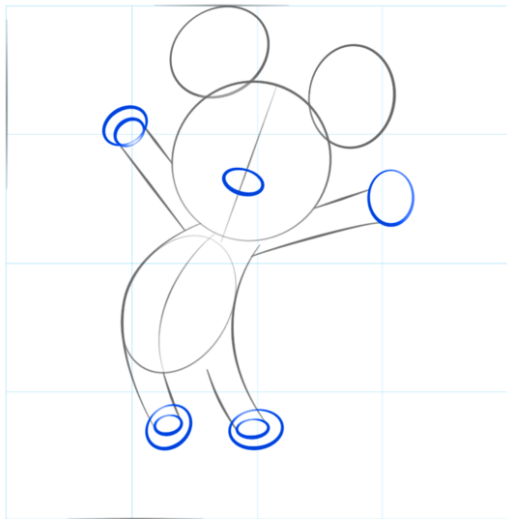
# Visual Representation

## Art



# Visual Representation

Art



SUPERCOLORING

# Visual Representation



SUPERCOLORING

# Visual Representation

## Programming

# Visual Representation

## Programming



# Visual Representation

Abstract versus Concrete

# Visual Representation

Abstract versus Concrete

Computer programs are ...

# Visual Representation

## Abstract versus Concrete

Computer programs are ... *abstract*.

# Visual Representation

## Abstract versus Concrete

Computer programs are ... *abstract*.

Can we see computer programs? –



# Visual Representation

## Abstract versus Concrete

Computer programs are ... *abstract*.

Can we see computer programs? – *No!*

# Visual Representation

## Abstract versus Concrete

Computer programs are ... *abstract*.

Can we see computer programs? – *No!*

Can we *visualise* computer programs? –

# Visual Representation

## Abstract versus Concrete

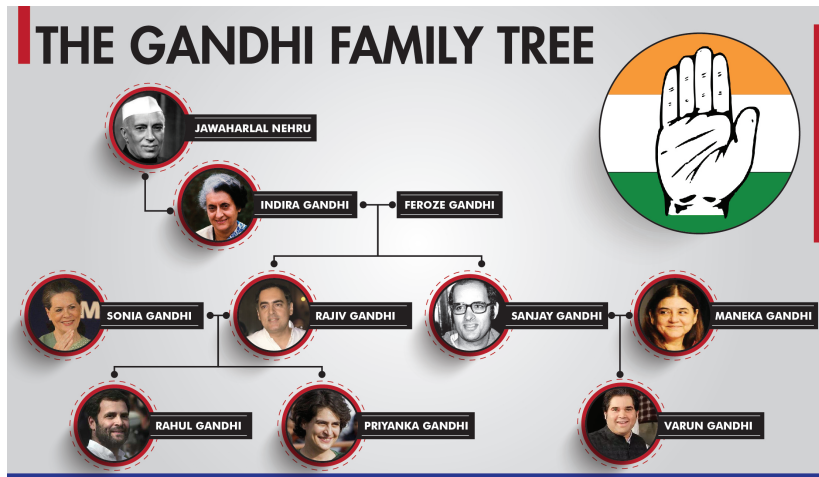
Computer programs are ... *abstract*.

Can we see computer programs? – *No!*

Can we *visualise* computer programs? – *Yes!*

# Visual Representation

## Family



# Visual Representation

## Programming

### (Dis)Connecting People!!



# Flowchart

## Sequence of Instructions

**PROBLEM 1:** You are in front of the door of your house.  
The door is already locked. Enter the house and lock the door  
from inside.

Unlock door

Open door

Walk into house

Shut door

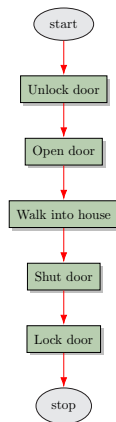
Lock door

# Flowchart

## Sequence of Instructions

**PROBLEM 1:** You are in front of the door of your house. The door is already locked. Enter the house and lock the door from inside.

Unlock door  
Open door  
Walk into house  
Shut door  
Lock door



# Flowchart

## If branch

**PROBLEM 2:** You are in front of the door of your house. It is not known earlier if the door is locked or not. Enter the house and lock the door from inside.

```
Check if the door is locked
or not
IF the check succeeds THEN DO
    Unlock door
ENDIF
```

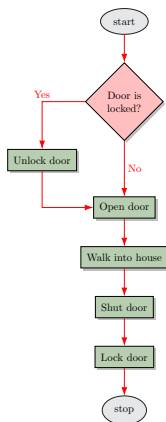


# Flowchart

## If branch

**PROBLEM 2:** You are in front of the door of your house. It is not known earlier if the door is locked or not. Enter the house and lock the door from inside.

```
Check if the door is locked
or not
IF the check succeeds THEN DO
    Unlock door
ENDIF
```



# Flowchart

## If-Else Branch

**PROBLEM 3:** We have to affix postage stamp of the following denomination to the envelope:

- ▶ Rs. 30 for speed post
- ▶ Rs. 35 for registered post

```
IF text T written on the  
given envelope E is  
"SPEED POST" THEN  
    Affix Rs. 30 stamp on E  
ELSE IF text T written on  
E is "REGISTERED POST" THEN  
    Affix Rs. 35 stamp on E  
ENDIF
```

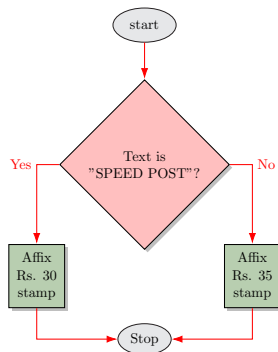
# Flowchart

## If-Else Branch

**PROBLEM 3:** We have to affix postage stamp of the following denomination to the envelope:

- ▶ Rs. 30 for speed post
- ▶ Rs. 35 for registered post

```
IF text T written on the  
given envelope E is  
"SPEED POST" THEN  
    Affix Rs. 30 stamp on E  
ELSE IF text T written on  
E is "REGISTERED POST" THEN  
    Affix Rs. 35 stamp on E  
ENDIF
```



# Flowchart

## Loop

**PROBLEM 4:** You are in front of the door of your house. The door is already locked. Enter the house and lock the door from inside. Take one step at a time.

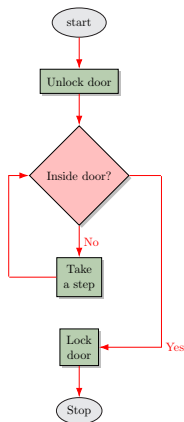
```
Unlock door
WHILE inside house is false THEN
    Take a step
DONE
Lock door
```

# Flowchart

## Loop

**PROBLEM 4:** You are in front of the door of your house. The door is already locked. Enter the house and lock the door from inside. Take one step at a time.

```
Unlock door
WHILE inside house is false THEN
    Take a step
DONE
Lock door
```



# Flowchart

## If-Else if-Else Branch

**PROBLEM 5:** Program that takes two numbers and a choice (1 for addition, 2 for multiplication) and does appropriate calculation.

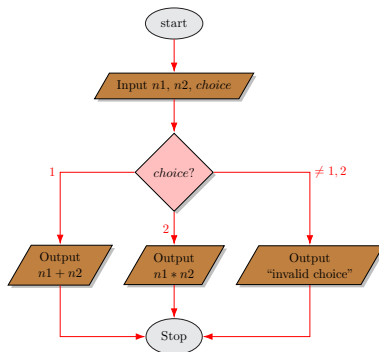
```
Input n1, n2, choice
IF choice = 1 THEN
    OUTPUT n1 + n2
ELSE IF choice = 2 THEN
    OUTPUT n1 * n2
ELSE
    OUTPUT "INVALID CHOICE."
```

# Flowchart

## If-Else if-Else Branch

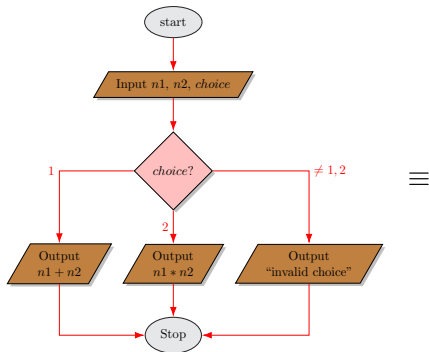
**PROBLEM 5:** Program that takes two numbers and a choice (1 for addition, 2 for multiplication) and does appropriate calculation.

```
Input n1, n2, choice
IF choice = 1 THEN
    OUTPUT n1 + n2
ELSE IF choice = 2 THEN
    OUTPUT n1 * n2
ELSE
    OUTPUT "INVALID CHOICE."
```



# Flowchart

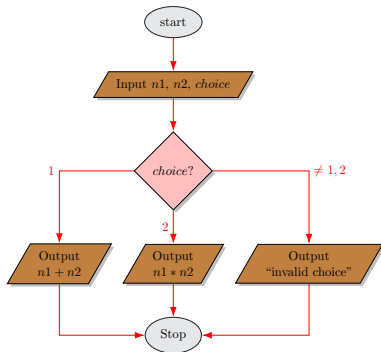
## If-Else if-Else Branch



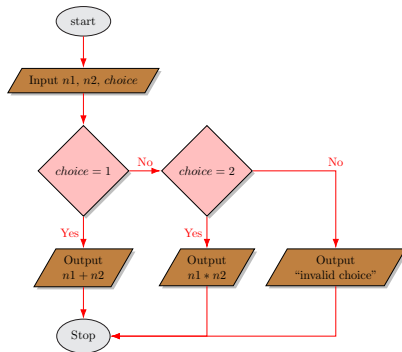


# Flowchart

## If-Else if-Else Branch

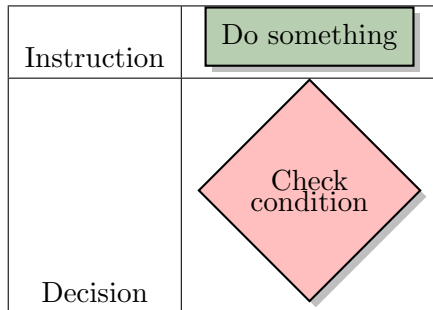


=



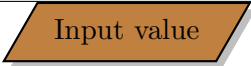

# Flowchart

## Building Blocks



# Flowchart

## Building Blocks

Input/Output	 Input value
Start/Stop	 Stop

# Flowchart

## Advantages

1. It helps think about the algorithm/process in a pictorial way.
2. It's not a formal language like a programming language (e.g. Python). Therefore, doesn't get stuck due to syntax errors.