

# Metrics

Saturday, April 4, 2020 12:28 PM

## Fan In:

Functions or methods that call another function or method.  
X is a function that calls other functions

## Fan Out:

Functions or methods that are called by another function.

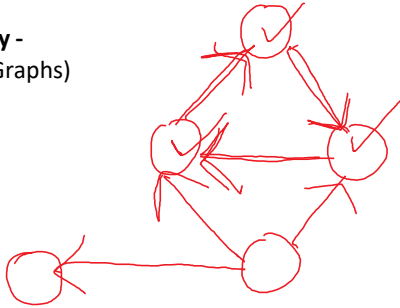
## Lines of Code/Length of Code

**Cyclometric Complexity -**  
(Can be computed on Graphs)

$$M = E - N + 2P$$

$$N = 5, E = 6, P = 3$$

$$M = 6 - 5 + 2 * 3 = 7$$



## Specialization Index (SIX)

$$SIX = \frac{NMO * DIT}{NMI + NMA + NMO} * 100\%$$

DIT = Depth of Inheritance  
NMA = Number of Methods Added (to the inheritance)  
NMI - Number of Inherited Methods  
NMO = The number of Overridden Operation

DIT = 2  
NMI = 2  
NMA = 1  
NMO = 2

$$SIX = \frac{(2 * 2)}{(1 + 2 + 2)} * 100\% = \frac{4}{5} * 100\% = 0.8 * 100\% = 80\%$$

Problem ->

```
Class Person{
    void read();
    void display();
}
Class Student extends Person{
    void read();
    void display();
    void getAverage();
}
Class GraduateStudent extends Student{
    void read();
    void display();
    void workStatus();
}
```

## Defect Removal Efficiency (DRE)

$$DRE = E / (E + D)$$

E = No. of Errors found **before delivery of the software to the end-user**  
D = No. of Defects found **AFTER delivery**

Sample:

- We found 100 defects during the testing phase and then later, say within 90 days after software release (in production), found five defects,
- Solution: DRE = E / (E + D) = 100 / (100 + 5) = 0.9524 \* 100% = 95.24%

## Fog Index

- Fog index = ( average number of words per sentence) + (number of words of 3 syllables or more) \* 0.4