Sri Lanka Institute of Information Technology

SOFTWARE ENGINEERING PROCESS AND QUALITY MANAGEMENT

Lecture 4 – Software Metrics (Cyclomatic Complexity Measure

How Can Software Quality Be Measured

- Software quality metrics:
  - Code Quality
  - Reliability
  - Performance
  - Usability
  - Correctness
  - Maintainability
  - Integrity
  - Security

Why are these metrics important

- They **help developers track and improve software quality**.
- They ensure **users get a good experience**.
- They **reduce risks and costs** related to software failures.

Cyclomatic Complexity Measure

- Measures the number of linearly independent paths in a program.

$$V(G) = e-n+2$$

$$V(G) = d+1$$

Vg = No of decision statements in each method + No of methods in a class

Approaches to Measure the cyclomatic complexity

Explain some approaches that can be used to measure the Cyclomatic Complexity of a program

Calculating Cyclomatic Complexity of a Byte Code

Calculate the cyclomatic complexity of the following byte code?

Method void D0(boolean, java.lang.String)
0 iload_0
1 ifeq 11
4 getstatic #10 <Field java.io.PrintStream out>
7 aload_1
8 invokevirtual #16 <Method void println(java.lang.String)>
11 return

Calculating Cyclomatic Complexity of a Byte Code

Calculate the cyclomatic complexity of the following byte code?

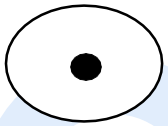
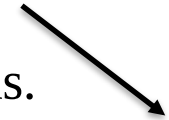
```
Method void D1(boolean, java.lang.String, java.lang.String)
 0  iload_0
 1  ifeq 14
 4  getstatic #2 <Field java.io.PrintStream out>
 7  aload_1
 8  invokevirtual #3 <Method void println(java.lang.String)>
11 goto 21
14 getstatic #2 <Field java.io.PrintStream out>
17 aload_2
18 invokevirtual #3 <Method void println(java.lang.String)>
21 return
```

Calculating Cyclomatic Complexity of a Compound Statements

You can’t expect the same cyclomatic complexity from all the approaches

The CC value obtained from the class file can be higher than CC obtained from the source file

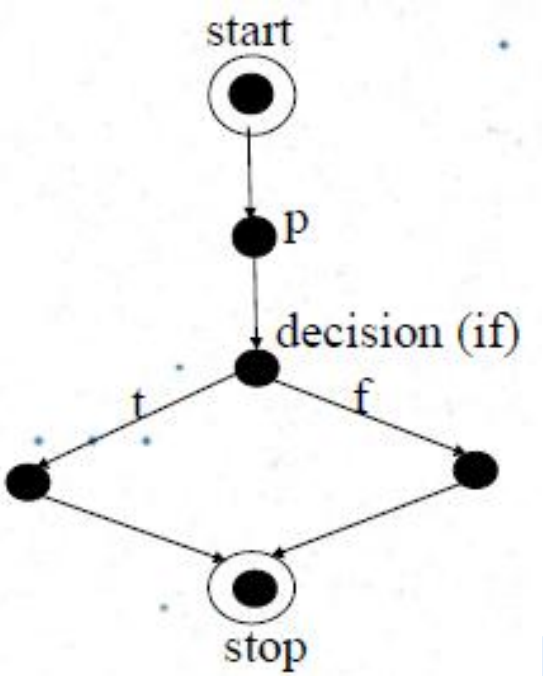
How to draw the Control Flow Graph

1. To represent a start or a stop node use the notation 
2. To represent an intermediary node use the notation 
3. Start node, stop node, decision nodes and true/false paths have to be labeled.
4. Edges should always indicate the directions. 
5. Along with a start node, procedure nodes, and decisions can also be represented.
6. A procedure node represents one or more non-decisional statements.

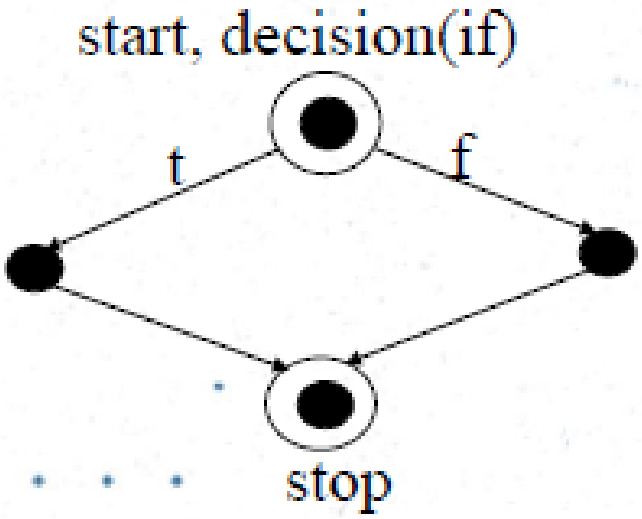
Control Flow Graph

```
int p;
if(p <10)
        System.out.println("Value of pis less than 10");
else
        System.out.println("Value of pis a grater than or equal to 10");
```

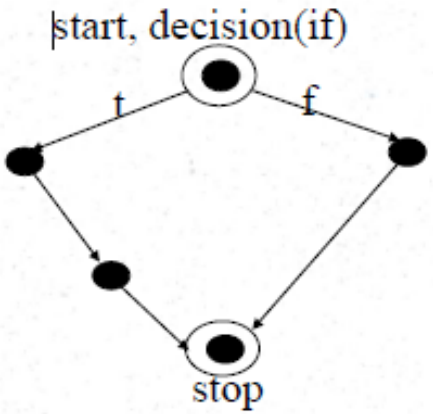
Control Flow Graph



Control Flow Graph



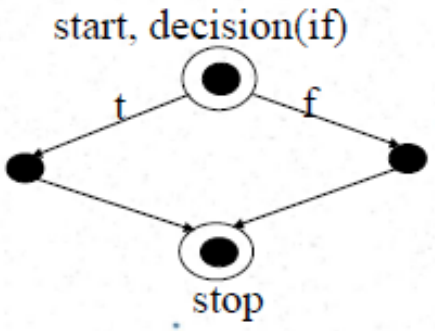
Control Flow Graph



$V(G) = e - n + 2$

$= 5 - 5 + 2$

$= 2$



$V(G) = e - n + 2$

$= 4 - 4 + 2$

$= 2$

Cyclomatic Complexity of a Class

Total Cyclomatic Complexity for a class (V_g) = Sum of the cyclomatic complexity of each method

$$V_g = \sum_{i=1}^{n} V(G_i)$$

$$V_g = \sum_{i=1}^{n} (d_i + 1)$$

$$\mathbf{V_g = n + d_i}$$

Where:

- n = Number of methods in the class
- G_i = Flow graph for method **i**
- d_i = Number of decisions in method **i**

Is the cyclomatic complexity the same from all approaches?

How has the V(G) = d+1 equation derived from V(G) = e-n+2

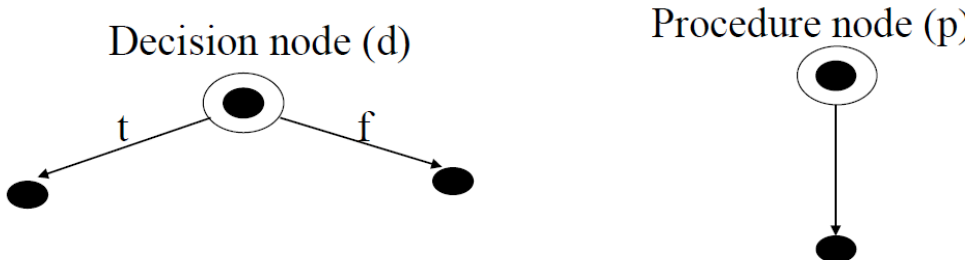
Nodes in a control flow graph:

- Decision nodes (d)
- Procedure nodes (p)
- Start node
- Stop node

Total number of nodes in a control flow graph = d + p + 1

How has the V(G) = d+1 equation derived from V(G) = e-n+2

Edges in a control flow graph:



Total number of edges in a control flow graph = 2d + 1p

How has the V(G) = d+1 equation derived from V(G) = e-n+2

e = 2d + p
Where
e = number of edges
d = decision nodes
p = procedure nodes

n = d + p + 1
Where
n = number of nodes

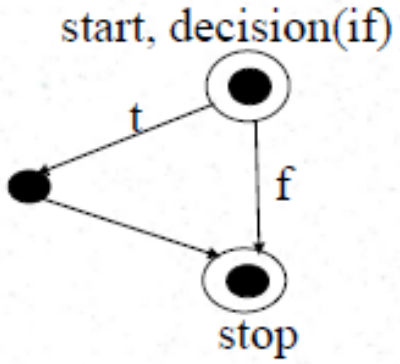
V(G)= e -n + 2
= (2d + p) -(d + p + 1) + 2
= d + 1

Question 1

```
public static void D0 (boolean a, String x){
        if(a)
                System.out.println(“x”);
        }
}
```

Question 1 - Answer

```
public static void D0 (boolean a, String x){
        if(a)
                System.out.println("x");
        }
}
```

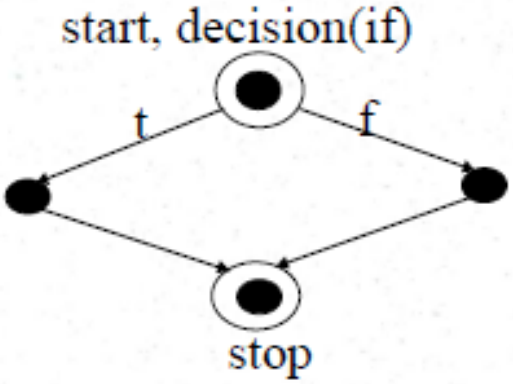


Question 2

```
public static void D1 (boolean a, String x, String y) {

        if(a)
                System.out.println("x");
        else
                System.out.println("y");
}
```

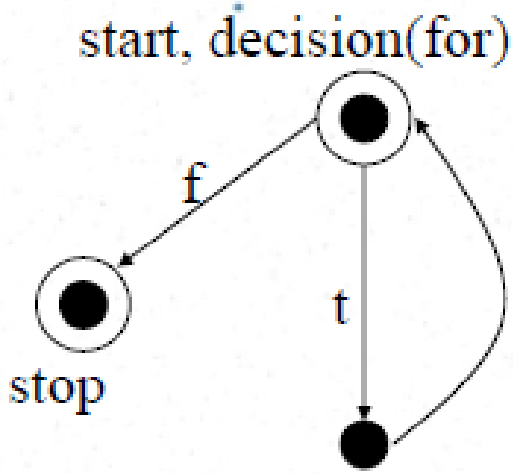
Question 2 - Answer



Question 3

```
public static void D3(int m, String x) {
        for(int i=0; i<m; i++)
                System.out.println("x");
}
```

Question 3 - Answer



$$V(G) = e - n + 2$$

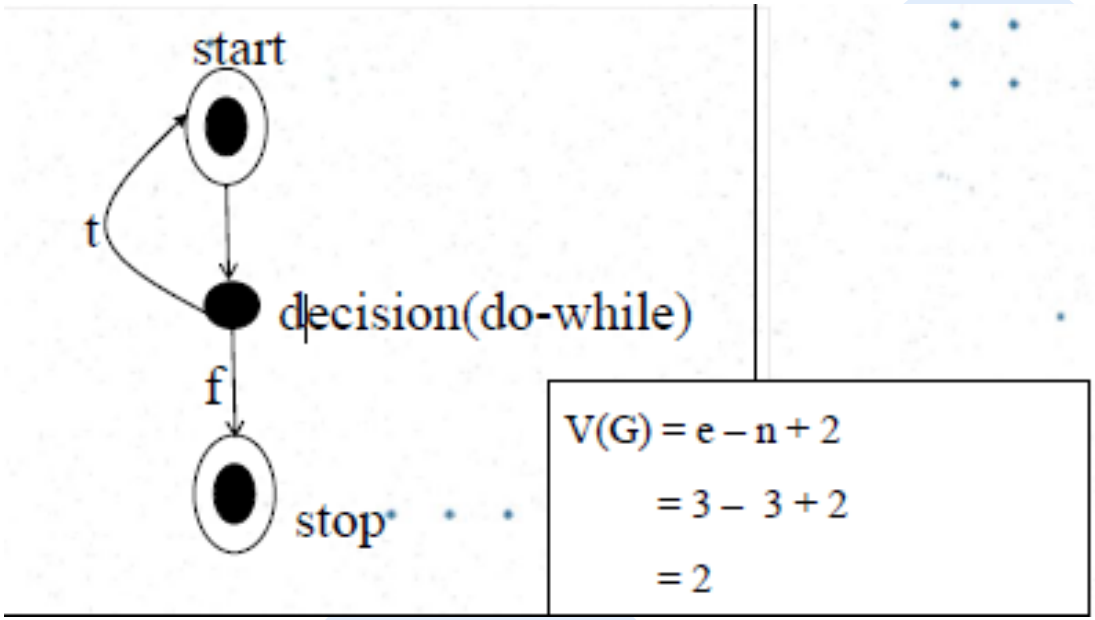
$$= 3 - 3 + 2$$

$$= 2$$

Question 4

```
public static void D3(int a, String x) {
        do
        {
                System.out.println("x");
                a++;
        } while (a <10)
}
```

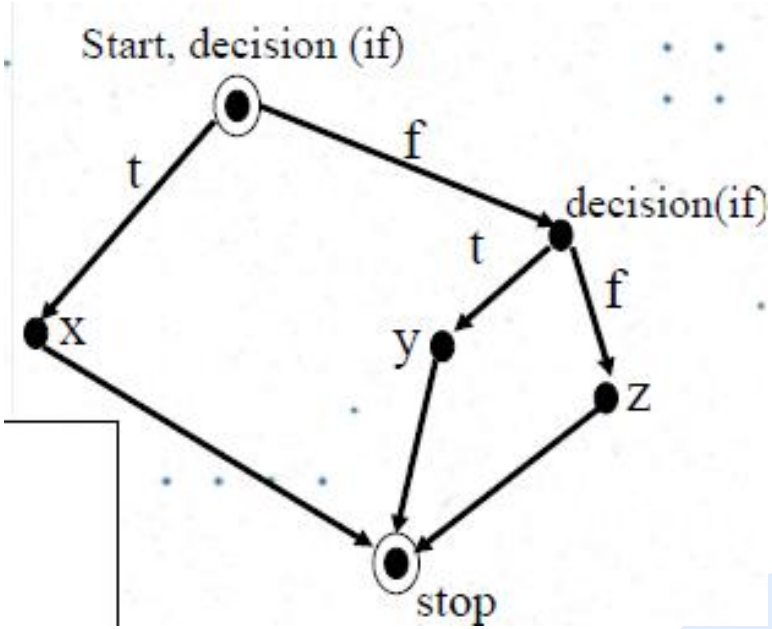
Question 4 - Answer



Question 5

```
void composite (boolean a, boolean b, String x, String y, String z) {
        if (a)
                System.out.println(x);
        else {
                if (b)
                        System.out.println(y);
                else
                        System.out.println(z);
        }
}
```

Question 5 - Answer

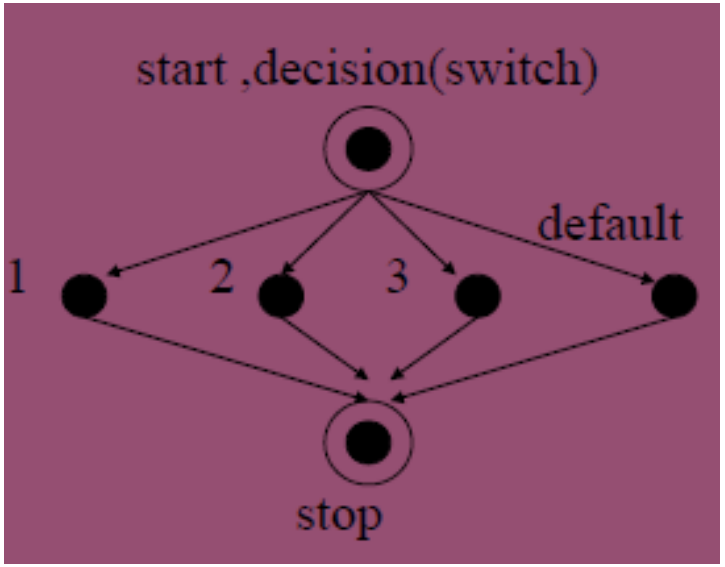


$$V(G) = e - n + 2$$
$$= 7 - 6 + 2$$
$$= 3$$

Question 6

```
public static void main (String[ ] args) {
        int i = 0;
        switch (i) {
                case 1: System.out.println("its 1");
                        break;
                case 2: System.out.println("its 2");
                        break;
                case 3: System.out.println("its 3");
                        break;
                default: System.out.println("its none");
                        break;
        }
}
```

Question 6 - Answer



$$V(G) = e - n + 2$$
$$= 8 - 6 + 2$$
$$= 4$$

THANK YOU!