

# Software Testing Mentor

[www.softwaretestingmentor.com](http://www.softwaretestingmentor.com)

**ISTQB Foundation Level and Software Testing Training**

[www.softwaretestingmentor.com](http://www.softwaretestingmentor.com)

# **Module 3**

## **Static Test Techniques**

### **Session 3 – Static Analysis by Tools**

# What is Static Analysis?

Examination of requirements, design documents or software code without executing the artifact under test is known as static analysis

Main objective of static analysis is to find defects before coding starts and software code is actually run

Goal of static analysis is to find defects rather than failures

# Tools for Static Analysis

Static analysis tools are mostly used by developers during unit, integration testing and by designers during modeling

The tools provide details about coding standards like nesting depth, cyclomatic complexity, check against coding standard and graphic depictions of the control flow

Compiler or development IDE is also a type of static analysis tool

Static analysis tools analyze software code and the output generated by code like HTML, XML

Static analysis helps to find defects which are difficult to find during dynamic testing

# Tools for Static Analysis Cont.

Some of the typical defects found by static analysis tools are:

- Unused variables
- Unreachable code
- Security vulnerabilities
- Coding standard violations
- Syntax issues

# Typical benefits of static analysis

Defects are detected prior to test execution

It is very effective in finding defects that are difficult to find by dynamic testing

Static analysis tools give warning about any incorrect code format and structure

Using static analysis improves maintainability of code

# Features of Static Analysis tool

Static analysis tools provide many features, most common features of static analysis tools are

- Coding standards
- Code metrics
- Code structure

# Coding standards

Static analysis tool checks adherence to coding standard

- Programming rules: Warning if declared variable is not used in code
- Naming conventions: Class name should always start with capital C

It is recommended that you adopt to existing standard

- It saves lot of effort
- Because coding standard is well known there will be many checking tools available which will support this standard



# Code metrics

Code metrics is about calculating structural attributes of code like depth of nesting, cyclomatic complexity, number of lines of code etc.

These metrics are helpful to see if the design of code is becoming bigger and complex

Code metrics also help to decide about design alternatives when redesigning the existing code

# Cyclomatic Complexity

Cyclomatic complexity metric is based on the number of decisions in the program

It is very important for testing because it provides information about the more complex code which requires more testing

## Ways to calculate Cyclomatic complexity

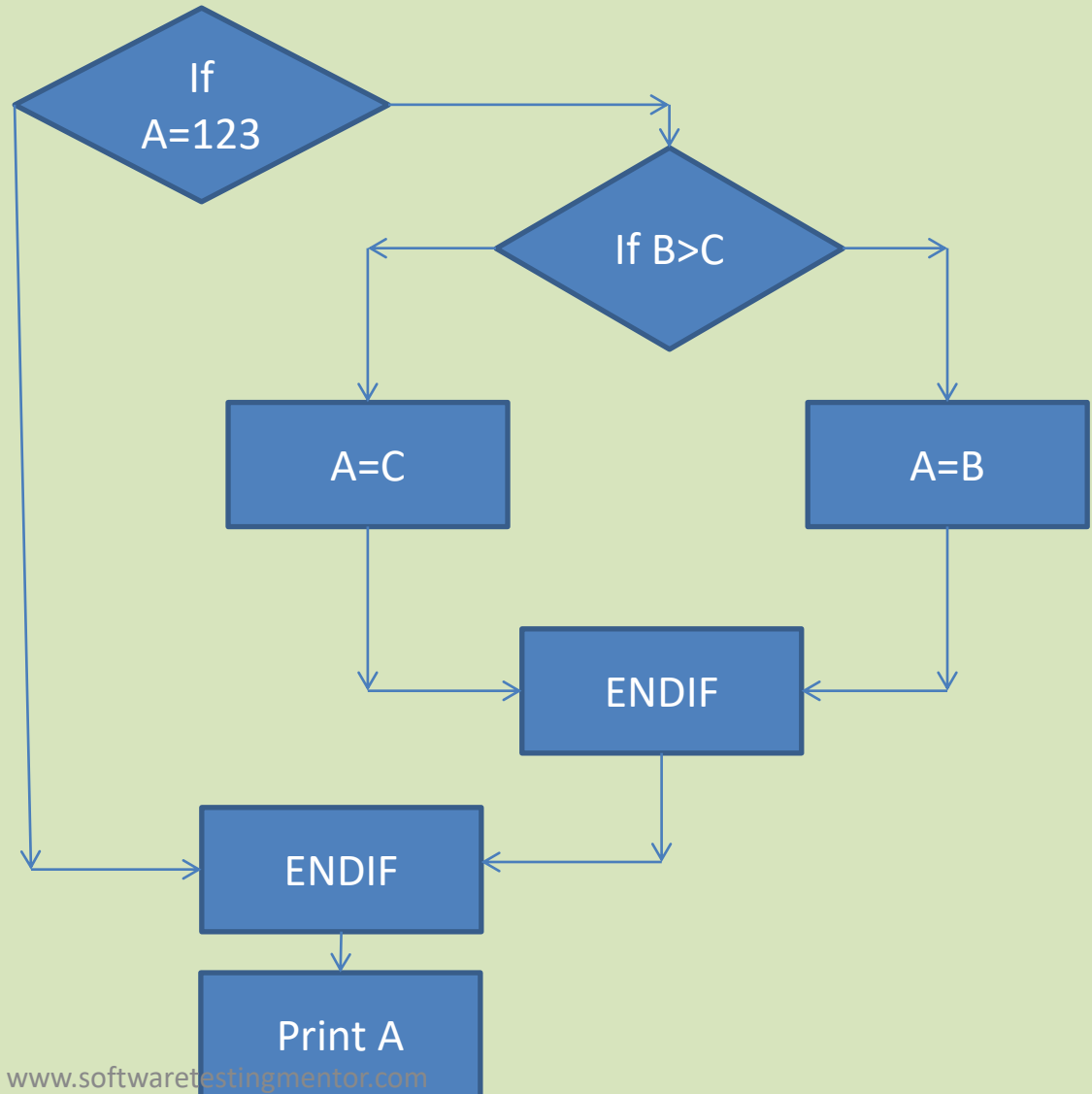
- Simplest way is to count number of binary decision statements(if,while,for etc)+1  
Cyclomatic complexity=No. of binary decisions+1
- More formal approach is to use the formulas given below:  
Cyclomatic complexity = Total no. of Edges – Total No. of Nodes + 2

# Cyclomatic Complexity example

```
IF A = 123
THEN IF B > C
THEN A = B
ELSE A = C
ENDIF
ENDIF
Print A
```

CC = No. of binary decisions + 1

CC = Total no. of Edges – Total No. of Nodes + 2



# Code Structure

Several  
aspects  
of code  
structure  
to  
consider  
are:

- control flow structure
- data flow structure
- data structure

# Control flow Structure

Control flow structure addresses the sequence in which instructions are executed

Control flow structure reflects the iterations and looks in a program design

Used to identify dead code

Code metrics like cyclomatic complexity, number of nested levels relate to control flow

# Data flow Structure

Data flow structure follows the trail of a data item as it is accessed and modified by the code

Data flow structure helps us to figure out how data is transformed by the program

Some common defects found by data flow structure are like

- Referencing a variable with undefined value
- Variable which are never used in program

# Data structure

Data structure refers to the data independent of program

It refers to algorithms for creating, modifying or deleting data

Knowledge of these algorithms provides lot of information about the difficulty in writing programs to handle the data

Sometime the program is complex because of the complex algorithm for data structure and not because of complex control flow or data flow

# Conclusion

- **What is Static analysis**
- **Typical benefits of static analysis**
- **Tools for Static Analysis**
- **Features of Static Analysis tool**
  - **Coding standards**
  - **Code metrics**
  - **Code structure**



**THANK YOU!!!**