

# Control Flow Testing

# Basis Path Testing

- **Basis path testing** is a specific **white box testing technique** that uses the program's control flow graph to identify a set of independent paths through the code.
- The goal is to create a set of test cases that guarantees the execution of every statement and branch at least once.
- This technique is based on a concept called the **cyclomatic number**

# Basis Path Testing

- Cyclomatic Number
  - The **cyclomatic number** is a quantitative measure of the logical complexity of a program unit.
  - It's calculated from the CFG and is used to determine the minimum number of independent paths that must be tested to achieve **basis path coverage**.
  - A higher cyclomatic number indicates a more complex program, which requires more testing
- $M=E-N+2P$

# Basis Path Testing

- **Selection of Minimum Number of Test Cases**
  - This refers to the core objective of basis path testing.
  - The cyclomatic number provides the **minimum number of test cases** required to guarantee that every independent path is executed.
  - This makes testing more efficient and thorough by focusing the effort on the most critical paths.

# Basis Path Testing

- **Steps to Generate Basis Paths**
  - **Draw the Control Flow Graph (CFG):** First, you create a visual representation of the program's logic
  - **Calculate the Cyclomatic Complexity:** Next, you determine the minimum number of independent paths needed for a complete test suite.
  - **Identify a Basis Set of Paths:** Now, you use the cyclomatic number to find a set of linearly independent paths. The number of paths in this set must equal the cyclomatic number.
    - Start by choosing a simple path through the graph that includes a few decision points.
    - Then, incrementally add new paths to the set. Each new path must introduce at least one new edge that was not included in the previously chosen paths.
    - Continue this process until you have a set of paths equal to the cyclomatic number, ensuring that every edge in the CFG is covered at least once.
  - **Derive Test Cases:** Finally, for each path in your basis set, you create a test case.