Data Flow Testing

Data Flow Testing is a white-box testing technique that focuses on the flow of data within a program. It analyzes how variables are defined, used, and killed (destroyed or out of scope) to detect:
Uninitialized variables
Incorrect data usage
Unused variables

DU Path (Definition-Use Path)

A Definition-Use (DU) Path is a fundamental concept in data flow testing, which focuses on how variables are defined and used in a program. It helps in detecting potential uninitialized variables, redundant computations, and incorrect variable usage by analyzing the flow of data within a program.

Key Terms:
1. Definition (D): A variable is assigned a value.
2. Use (U): A variable's value is used in computations or conditions.
    1. Computational Use (C-Use): Used in arithmetic operations.
    2. Predicate Use (P-Use): Used in decision-making conditions (e.g., `if`, `while`).
3. DU Chain: A sequence of execution where a defined variable is used before being redefined or going out of scope.

Types of DU Paths:

1. Simple DU Path (Direct Definition to Use)

A variable is defined and later used without any redefinition in between.

Example:
```
int x = 5;   // Definition of x
int y = x + 2; // Use of x (C-Use)
```

DU Chain: x → y = x + 2

2. Extended DU Path (Definition to Use Across Multiple Statements)

A variable is defined in one block and used in another without redefinition.

Example:
```
int x = 10;  // Definition of x
if (x > 5) {  // Predicate Use (P-Use) of x
    int y = x * 2;  // Computational Use (C-Use) of x
}
```

DU Chains:
- **x → if (x > 5) // P-Use**
- **x → y = x * 2 // C-Use**

3. Loop DU Path (Definition Used in a Loop)

A variable is defined outside or within a loop and used repeatedly.

Example:

```
int sum = 0;  // Definition of sum
for (int i = 1; i <= 5; i++) {
    sum = sum + i;  // Use of sum (C-Use)
}
```

DU Chain: sum → sum = sum + i