**Data Flow Testing** is a type of structural testing. It is a method that is used to find the test paths of a program according to the locations of definitions and uses of variables in the program. It has nothing to do with data flow diagrams.  
It is concerned with:

* Statements where variables receive values,
* Statements where these values are used or referenced.

To illustrate the approach of data flow testing, assume that each statement in the program assigned a unique statement number. For a statement number S-

DEF(S) = {X | statement S contains the definition of X}

USE(S) = {X | statement S contains the use of X}

If a statement is a loop or if condition then its DEF set is empty and USE set is based on the condition of statement s.

Data Flow Testing uses the control flow graph to find the situations that can interrupt the flow of the program.  
Reference or define anomalies in the flow of the data are detected at the time of associations between values and variables. These anomalies are:

* A variable is defined but not used or referenced,
* A variable is used but never defined,
* A variable is defined twice before it is used

**Advantages of Data Flow Testing:**  
Data Flow Testing is used to find the following issues-

* To find a variable that is used but never defined,
* To find a variable that is defined but never used,
* To find a variable that is defined multiple times before it is used,
* Deallocating a variable before it is used.

**Disadvantages of Data Flow Testing**

* Time consuming and costly process
* Requires knowledge of programming languages

**Example:**

**1.** read x, y;

**2.** if(x>y)

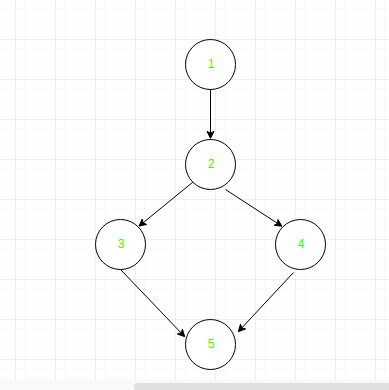
**3.** a = x+1

else

**4.** a = y-1

**5.** print a;

**Control flow graph of above example:**



**Define/use of variables of above example:**

| Variable | Defined at node | Used at node |
| --- | --- | --- |
| x | 1 | 2, 3 |
| y | 1 | 2, 4 |
| a | 3, 4 | 5 |

A definition use path (du-path) for a variable is a path from the statement in which the variable is defined to the statement in which it is used.

A definition clear path (dc-path) for a variable is a du-path for the variable having a node for defining the variable and a node for using the variable, such that there is no other node on the du-path path which is a defining node of the variable (that is, there is no other assignment statement on that path).

The du-paths and dc-paths describe the flow of data across source statements from points at which the values are defined to points at which the values are used. The du-paths which are not definition clear are potential trouble paths for a program.

For the above example, the du-paths are:

|  |  |  |
| --- | --- | --- |
| **Variable** | **du-paths** | **Is the path definition clear?** |
| x | 1, 2  1, 3 | Yes  Yes |
| y | 1, 2  1, 4 | Yes  Yes |
| a | 3, 5  4, 5 | No  Yes |