Brief:

1. Data flow testing is related to path testing.
2. Node can be either a defined node or a used node.
3. DU path is the path.

Use Node:

Use node has been further categorized as:

1. P-use: Predicate use i.e a node which deals with conditions, e.g if-else etc.
2. C-use: Called as computation use, e.g sum = sum + 2.
3. O-use: Output use, e.g print i.
4. L-use: Location use, e.g a[i] as used in an array.
5. I – Use: Iteration use, e.g i = i + 1.

i = i + 1 is an example of both define and use. It is getting defined at left side and is being used at right.

DU Testing Process:

1. Select the set of paths and test them to find bugs, if any. Bugs can be in any of the following forms: a variable has been defined but is not used; or a variable has been used but not defined.
2. To ensure that all the paths, edges and nodes have been tested.
3. Further bugs can be any of following: variable is defined multiple times before being used; deallocating a variable before it is being used.

Example (for practice):

1. int x, y, z
2. if (x > y)
3. a = x + 2
4. print a

   else
5. a = y + 4

6. print m

Program Slices:

1. If a program is large or complex, it can be further split into various sub-programs.
2. Slicing can be achieved upto any point in the program, but the impact of control statement, if any must be taken in consideration, e.g effect on line number which lies between a loop must be taken care in slice.