**Software Unit Tests**

1. Boundary Testing

Need to test:

* Nth element
* N+1 element
* 0 element
* Halfway element
* Ith iteration
* i+1 iteration
* just below 0 element (-1 or -2)

1. Error-handling path testing

Need to test:

* Error handling descriptions
* Error handling correspondence
* Exception condition processing
* Error location

1. Basic path testing

Need to test:

* Test if statements such that each statement is executed at least once
* Test else statements such that each statement is executed at least once
* Predicted flow, notarised flow and actual flow needed (derive graphs0
* Test while statements such that each statements executes at least once
* Test each case statement such that it executes at least once

1. Independent path testing

Need to test:

* From flow graphs, test each independent path for current execution

1. Control structure testing

Need to test:

* Simple loops: skip loop; one pass through loop, two passes through loop, m passes through loop where m<n, n-1, N and N+1 passes through loop.
* Nested loops: Test innermost loop- all other loops set to minimum values, perform simple loop tests. Move to next loop outwards keeping all other loops at minimum values.

1. Random testing (OO-specific)

* Isolate class
* Determine operations possible for class
* Determine minimum class behaviour
* Determine where other behaviour introduces variety
* Generate test cases of different operation sequences

1. State-based partitioning testing (OO-specific)

For each individual class:

* Identify state changing methods
* Identify nonstate changing methods
* Derive test cases where state operations are exercised
* Derive test cases where nonstate operations are exercised
* Always include minimum operation sequence within test cases.