1. All code outside my code being in test works as expected.
2. Testing with code flow: Just flow with code(method) and follow all the paths in the code and make sure that we traversed all the paths(You can think of path coverage and branch coverage here)
3. Test with input : for any method in java we can have method input
   1. By parameter: Test for all possible parameter inputs. And also for their **combinations**
   2. By output from method call on external objects(passed in or referenced): output from external objects can be decision maker(sometimes code breaker)
   3. All possible values of instance variables used in my method.

For this we need to check for boundary conditions (edge cases) and one/few from failure and success ranges.

1. Testing with business conditions: Here we need the permutation of all the external and internal factors affecting our code/method being tested.

In our system we need to go on tweaking our strategies and also need to flow through external/outside code.

JMock comes handy when you need to set lot of expectations 🡪 try to avoid but it’s tough avoid.

JDummy comes in when we don’t have interface for some class 🡪 this should be our last weapon.

Thanks to ApplicationTestHelper 🡪 this is used when we are taking module from Application using static method. And also to start Application in junit testing mode (I am not sure about any such Application running mode but we call it this way)

In our system there is support for white box testing for testing private methods (you need to search in code base) 🡪 For private method we always prefer to flow junit through them in the testing of public methods they are being called.