What abilities should you have:

1. Big picture understanding: find the most important things. E.g. to test amazon system, payment is in the first place.
2. Know how the pieces fit together: integration. E.g. do not forget to do integration test
3. Organization: break down the parts. Before state something, organize them well
4. Practicality: do reasonable test. Not just restart and reinstall…

Test a real world object:

1. Who are the users and for what reasons?
2. What are the use cases? How to use it?
3. What are the bounds of use? Consider the environmental factors
4. What are the stress/failure conditions? Under what conditions, it will be broken?
5. How would you perform the testing? Organize the well before stating. Both intended and unintended.

Test a piece of software:

Manually vs automated? Black box vs white box

1. Are we doing black box testing or white box testing?
2. Who are the users and for what reasons?
3. What are the use cases? How to use it?
4. What are the bounds of use? Consider the environmental factors
5. What are the stress/failure conditions? Under what conditions, it will be broken?
6. How would you perform the testing? Organize the well before stating. Both intended and unintended.

Test a function:

1. Discuss assumptions and how to handle specific situations.
2. Define the test cases. (normal cases, the extremes, nulls and illegal, strange input)
3. Define the expected results.
4. Write test code. assertEquals()

Troubleshooting:

1. Understand the scenario (How long it lasts? What tools they use? How often it happens? Any error reports?)
2. Break down the problem
3. Create specific manageable tests