Cognizant | Academy
Passion for making a difference

Exploring JUnit 4.x

Targeted at: Entry Level Trainees

Session 1: Testing Fundamentals

Academy

About the Author

Created By:	B Sai Prasad, (105582)
Credential Information:	Sun Certified Java Programmer, Microsoft Certified Technology Specialist, PMI-certified Project Management Professional
Version and Date:	JUnit/PPT/1110/1.0

Cognizant Certified Official Curriculum

Icons Used

Questions

Tools

Hands-on Exercise

Coding Standards

Test Your Understanding

Reference

Try it Out

A Welcome Break

Contacts

Session 01: Testing Fundamentals overview

- **Introduction:**
  - Testing is an *integral part* of the system development function; testing starts with the requirements, not the code
  - Testing can only establish that the system *does not function* properly under specific conditions
  - Testing can *never* completely establish the correctness of the system software
  - In this chapter, associates would know the answers of what is software testing and why it is important

Session 01- Testing Fundamentals: Objective

- **Objective:**

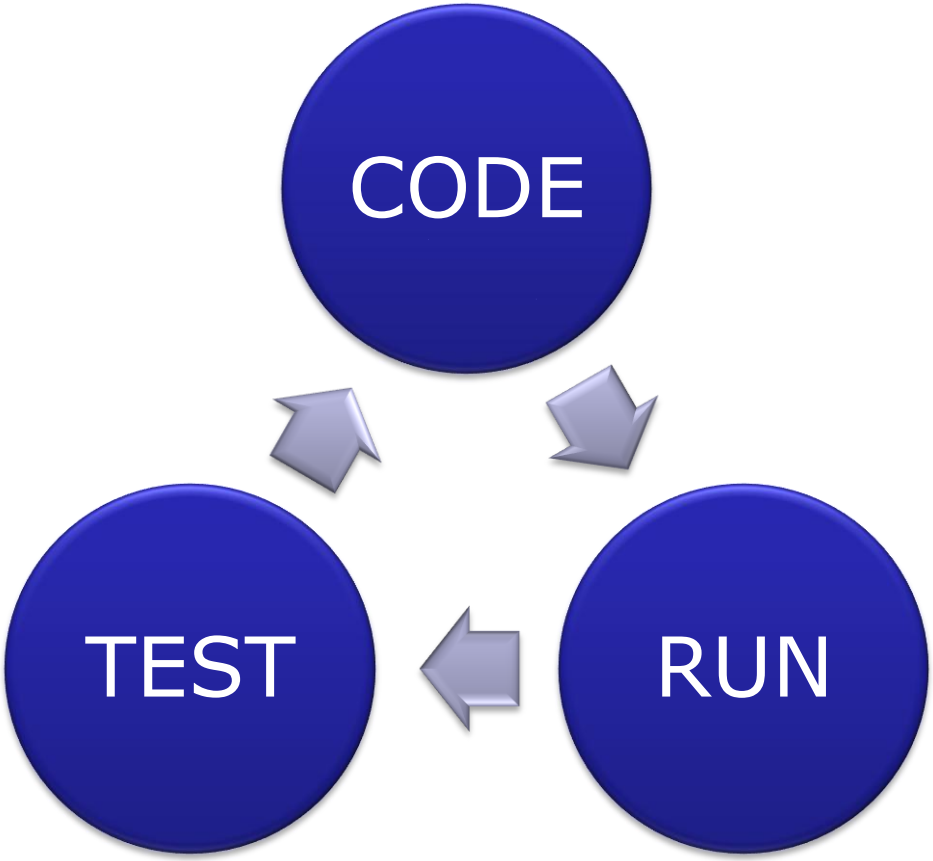
After completing this chapter, associates will be able to:

- Define the role of software testing
- Describe why faults should be found early
- Explain unit testing
- Know why unit testing alone is not important
- List the five flavors of software testing

Testing objectives

- The objective of testing includes:
  1. Testing is a process of executing a program with the *intent of finding an error*
  2. A good test case is one that has a *high probability* of finding an *undiscovered error*
  3. A successful test is one that *uncovers* an undiscovered error

Testing in Development Cycle



Finding faults early

- It is commonly believed that, the earlier a defect is found, the cheaper it is to fix

		TIME DETECTED				
		Requirements	Architecture	Construction	System Test	Post Release
TIME INTRODUCED	Requirements	1x	3x	5-10x	10x	10-100x
	Architecture	-	1x	10x	15x	25-100x
	Construction	-	-	1x	10x	10-25x

Unit Testing

- A unit test examines the *behavior* of a distinct *unit of work*
- Unit of work is a task that is *not directly dependent* on the completion of any other task
- Unit tests should be *fine grained*, testing small numbers of closely-related methods and classes
- Unit tests focus on testing whether a method is following the terms of its *API⁺ contract*
- API⁺: Application Programming Interface - a formal agreement between the caller and the called

Writing Unit Tests

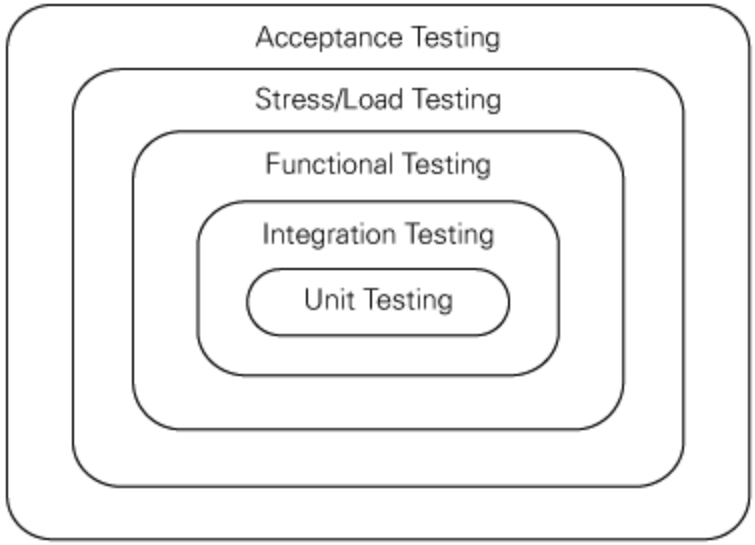
- Main Goal: To verify that the application works and try to catch bugs early
- Unit tests are more powerful than functional testing because
  - They allow greater *test coverage*
  - They enable *teamwork*
  - They prevent *regression* and limits the need for debugging
  - They give the courage to *re-factor*
  - They improve the implementation design
  - They serve as *developer's documentation*

Unit testing

- Why is unit testing alone not sufficient?
  - What happens when different units of work are combined into a workflow?
  - Will the end result of the workflow meet what the associates expect?
  - How well will the application work when many people are using it at once?
  - Will the application meet everyone's needs?

Five flavors of Software Testing

Five Flavors



- Innermost software tests are narrowest in scope
- Outermost software tests get more functional

Definition:

- Unit testing: It should examine the behavior of a distinct *unit of work*
- Integration software testing: It should examine the *interaction* between components in their target environment
- Functional software testing: It should test the application *use cases*
- Stress/load testing: It should test the *performance* of the application
- Acceptance testing: It should ensure that the application has met the *customer's goals*

- Allow time for questions from participants

Test Your Understanding

- What are the key objectives of testing?
- What is unit testing?
- Why is unit testing alone not sufficient?
- What are the five flavors of software testing?

Testing Fundamentals Session 1 : Summary

- Testing is an integral part of the system development function; testing starts with the requirements, not the code
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an undiscovered error
- A unit test examines the behavior of a distinct unit of work

Testing Fundamentals Session 1 : Summary (Contd..)

- Unit testing alone is not sufficient as the application needs to be examined for:
  - Interaction between components
  - Compliance with the software requirements specification
  - Performance
  - Meeting the customer's goals

Testing Fundamentals Session 1: Source

- Book:
  - JUnit in Action *by Vincent Massol; Ted Husted*
- Web:
  - Wiki: http://en.wikipedia.org/wiki/Software_testing

Cognizant | Academy
Passion for making a difference

You have completed the Session 1 Testing Fundamentals

Academy