Testing

• Why bother testing?
• What are differences between testing and debugging?
Types of Testing

- **Unit testing**: tests (smallest) individual units of source code, usually methods and functions
  - car: screws, steering wheel, windshield, etc.
- Integration testing: Looks for errors existed in the connection of multiple parts/modules
  - car: engine, transmission, A/C, etc.
- System testing: tests the entire system in terms of behavior and performance
  - car: 0 to 60 mph acceleration, fuel consumption in real road condition, etc.
- Others: reliability testing, regression testing, UI testing...
JUnit

• A simple (unit) testing framework for Java
• Current version: JUnit 4
Assertion Methods
(org.junit.Assert)

- `assertEquals(message, expected, actual)`
- `assertTrue(message, condition)`
- `assertNull(message, object)`
- `assertSame(message, expected obj, actual obj)`
- `fail(message)`
- ‘message’ is optional
- `assertNotEquals`, `assertFalse`, `assertNull`, `assertNotSame`...
Annotations ‘@’ JUnit 4

• What if you want to test whether a method throws an exception or not when it should?
  – 1. try/catch + fail or
  – 2. Use ‘expected’ annotation:
    ```java
    @Test(expected=SomeException.class)
    ```

• What if you want to set a timeout for a method?
  – Use ‘timeout’ annotation:
    ```java
    @Test(timeout=500) [i.e. 500 ms]
    ```

• Other annotations:
  – `@Before/@After, @BeforeClass/@AfterClass`