National University of Computer & Emerging Sciences, Karachi
Quiz 2 – BCS – 6B (Spring-2023)
29th March 2023, 10:40 AM – 12:20 PM

Course Code: CS3009	**Course Name: Software Engineering**
Instructor Name: Dr. Syed Muazzam Ali Shah	
Student Roll No:	**Section No:**

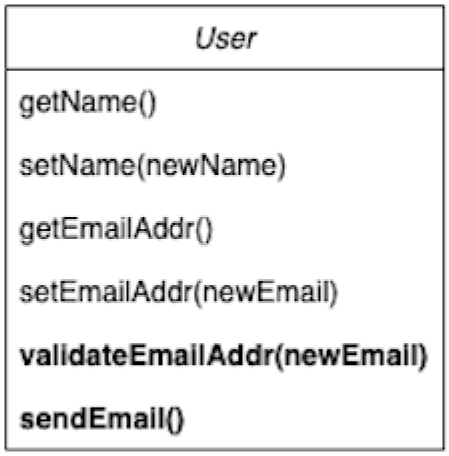
Time: 25 minutes. **Max Marks**: 10 Points

Questions 01: What is functional independence. Describe the two criteria (with example for each criteria) to assess the functional independence.

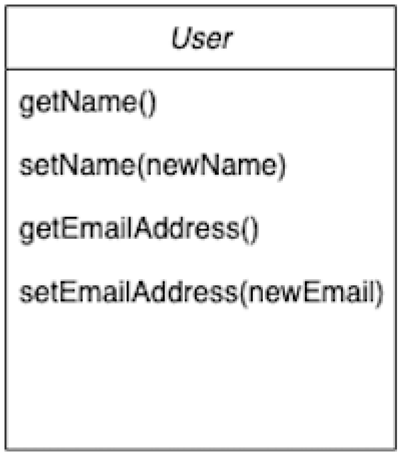
Answer:

Functional independence occurs where modules (such as a package or class) address a specific and constrained range of functionality. The modules provide interfaces only to this functionality. By constraining their functionality, the modules require the help of fewer other modules to carry out their functionality.

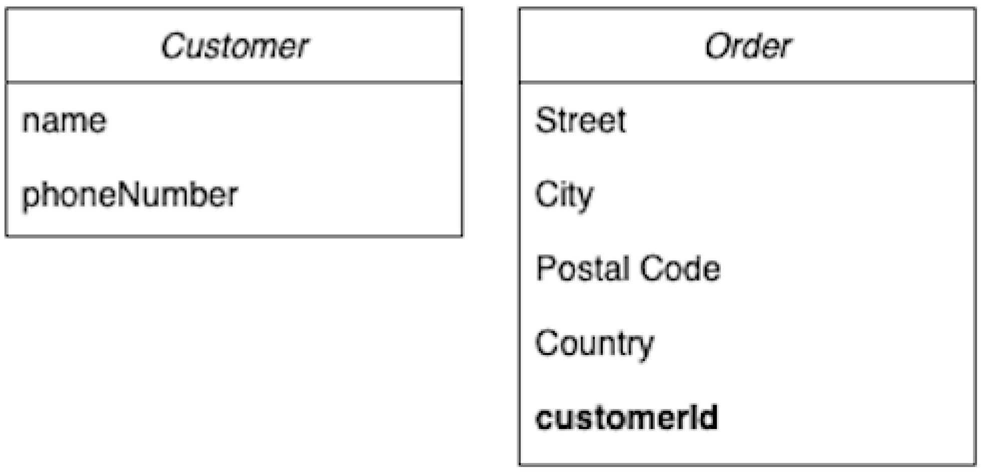
- Independence is assessed using two criteria:
- *Cohesion* is an indication of the relative functional strength of a module.
    - A cohesive module performs a single task, requiring little interaction with other components in other parts of a program. Stated simply, a cohesive module should (ideally) do just one thing.
- *Coupling* is an indication of the relative interdependence among modules.
    - Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface.



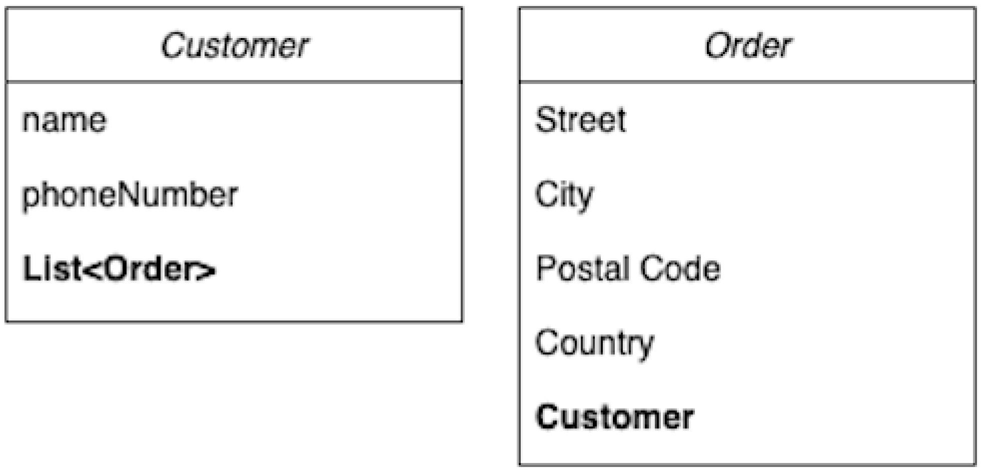
Low Cohesion



High Cohesion



Loose Coupling



Tight Coupling

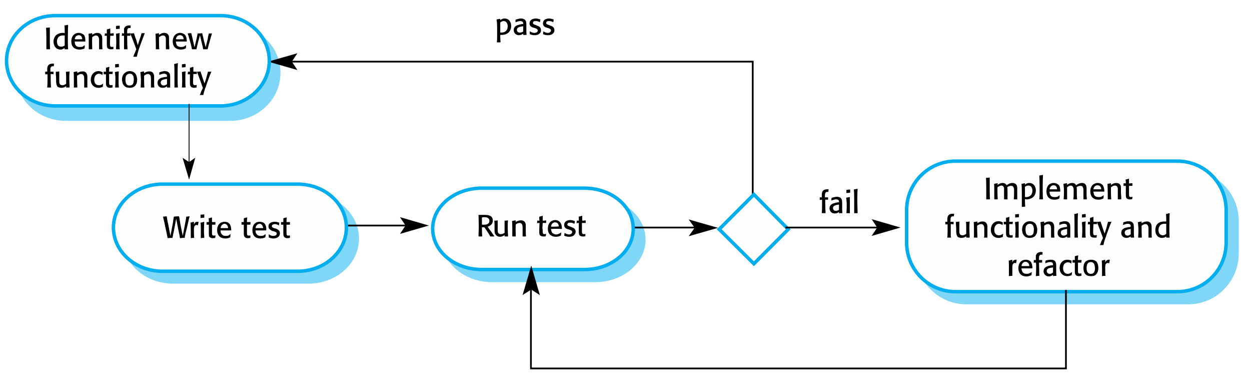
Question 02: What is the difference between data and procedural abstraction, explain with examples?

- ✧ **Procedural abstraction:**
  - ✧ the separation of the logical properties of an action from the details of how the action is implemented.
- ✧ **Data abstraction:**
  - ✧ the separation of the logical properties of data from the details of how the data are represented.

Question 03: Explain the test-driven development approach and also show the process of test driven development with the help of the diagram?

Answer:

- ✧ Test-driven development (TDD) is an approach to program development in which you inter-leave testing and code development.
- ✧ Tests are written before code
- ✧ and 'passing' the tests is the critical driver of development.
- ✧ You develop code incrementally, along with a test for that increment. You don't move on to the next increment until the code that you have developed passes its test.



pass
Identify new functionality
Write test
Run test
fail
Implement functionality and refactor

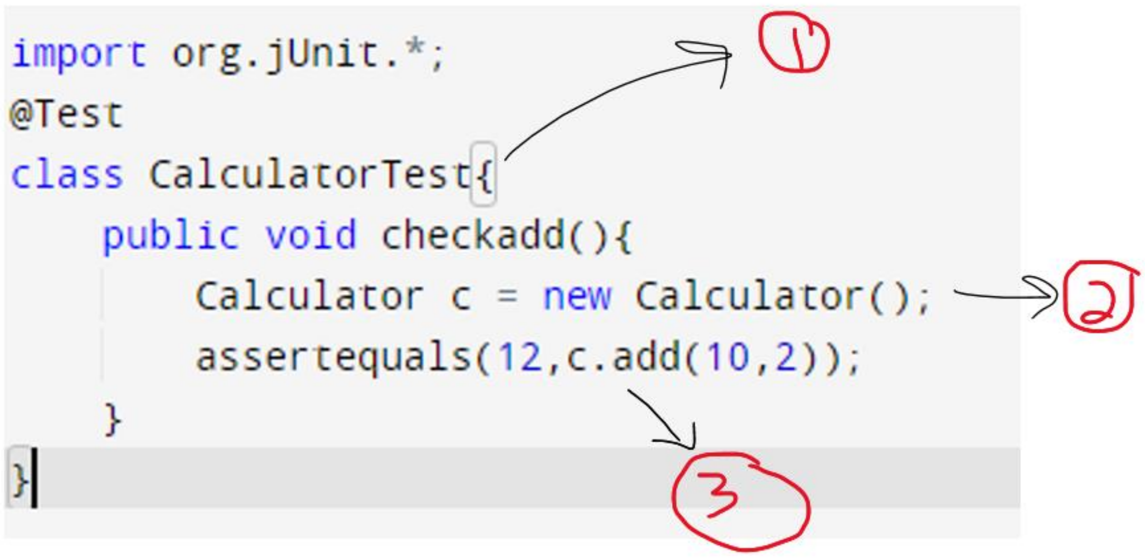
Question 04: What does mean by automated testing, describe the three components of automated testing with java code example.

Answer:

- ✧ Unit testing should be automated so that tests are run and checked without manual intervention.

  Three components of automated testing

  1. A setup part, where you initialize the system with the test case, namely the inputs and expected outputs.
  2. A call part, where you call the object or method to be tested.
  3. An assertion part where you compare the result of the call with the expected result. If the assertion evaluates to true, the test has been successful, if false, then it has failed.



```
import org.jUnit.*;
@Test
class CalculatorTest{
    public void checkadd(){
        Calculator c = new Calculator();
        assertequals(12,c.add(10,2));
    }
}
```

Question 05: List and briefly describe various types of interfaces that exist between program components.

Answer:

- Interface types between program components:
    - Parameter interfaces
        - Data passed from one method or procedure to another.
        - E.g., methods in object have a parameter interface, (passing parameters to multiple methods)
    - Shared memory interfaces
        - Same Block of memory is shared between different components.
        - Like in embedded systems, sensor component store value in db and analyser component may work on these values.
    - Procedural interfaces
        - Sub-system encapsulates a set of procedures to be called by other sub-systems.
    - Message passing interfaces
        - Sub-systems request services from other sub-systems

- E.g., MPI in client server arch.