Software Testing Unit-1 Quick Revision (JNTUA R20)

Purpose of Testing

- To verify the software works correctly.
- Bug-free code is a myth. Even good code has 1-3 bugs/100 lines.
- Testing improves productivity and quality.
- Prevention is better than detection.

Dichotomies

- Testing vs Debugging: Testing finds bugs; debugging fixes them.
- Function vs Structure: Black-box vs White-box testing.
- Designer vs Tester: Designer may be biased. Testers should be different.
- Modularity vs Efficiency: Small modules = easier testing.
- Builder vs Buyer: Builder, Buyer, Tester, User have different views.

Model for Testing

- Program: Simplified model of the software.
- Environment: OS, hardware, compilers, etc.
- Bug Model: Assumptions about bugs.

Consequences of Bugs

- Depends on Frequency, Cost, Impact.
- Bug impact: 1 (Mild) to 10 (Infectious).
- Severe bugs = high cost + risk.

Path Testing Basics

- Structural testing technique using control flow paths.
- CFG includes nodes (code), links (paths), decisions, junctions.
- Coverage:
  * Statement (C1): Every line.
  * Branch (C2): Every condition.
  * Path: Every possible path (not practical).

Loop Testing

- Test with 0, 1, 2, max, max+1 iterations.
- Nested Loops: Start inner, then outer.
- Avoid horrible loops (complex structure).

Likely Questions (JNTUA R20)

Short (2M):
1. Define testing and debugging.
2. What is a bug?
3. What is a control flow graph?

Software Testing Unit-1 Quick Revision (JNTUA R20)

Long (10M):

1. Explain dichotomies.
2. Describe model for testing.
3. Explain consequences of bugs.
4. What is path testing and coverage?
5. Explain loop testing.