

Total No. of Pages: 2
6th Semester
End Semester Examination

Roll No.....
B. Tech.
(May-2018)

MC 310: Software Engineering

Time: 3 Hours

Max. Marks: 50

Note: Attempt any five questions. Each question carries equal marks.

1. (a) What do you understand by a software life cycle model? What problems might occur if a software development organization does not use any specific life cycle model?

(b) Discuss quality function deployment technique of requirements elicitation. Why an importance or value factor is associated with every requirement?
2. (a) Discuss the prototype model. What is the effect of designing a prototype on the overall cost of the software project?

(b) What are size metrics? How is function point metric is advantageous over LOC metric?
3. Given the following, create a context-level DFD for a university library system.

The system must record the books owned by the library. The library manager will update this list of books on a regular basis. The system must record who has borrowed what books, and which books have been returned. Before someone can borrow a book, they must show a valid ID card that is checked to ensure that it is still valid against the student database maintained by the Registrar's Office (for student borrowers), the faculty/staff database maintained by the Personnel Office (for faculty/staff borrowers), or against the library's own guest database (for individuals issued a "guest" card by the library). The system must also check to ensure the borrower does not have any overdue books or unpaid fines before he or she can borrow another book. Every Monday, the library prints and mails postcards to those

people with overdue books. If a book is overdue by more than two weeks, a fine will be imposed. Sometimes books are lost or are returned in damaged condition. The manager must then remove them from the database and will sometimes impose a fine on the borrower.

4. (a) Explain, with an example, how an intermediate COCOMO provides more accurate estimates as compare to basic COCOMO.

(b) Explain with examples, top down, bottom up and hybrid approaches of software design.

5. (a) Define module cohesion and explain defferent types of cohesion. If a module has logical cohesion, what kind of coupling is this module likely to have with others?

(b) For a program with number of unique operators $\eta_1 = 30$ and number of unique operands $\eta_2 = 50$, compute the program volume, effort and time, program length and program level.

6. (a) Define the terms 'software reliability' and 'software quality'. How can these be measured ?

(b) Differentiate between alpha testing and beta testing and explain black box testing.

7. (a) Explain Boehm software quality model.

(b) Explain equivalence class testing and integration testing.