

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 80445

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2024.

Fourth/Fifth Semester

Computer Science and Engineering

CS 8494 – SOFTWARE ENGINEERING

(Common to : Computer and Communication Engineering / Computer Science and Business Systems / Information Technology)

(Regulations 2017)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Which process models will you prefer for the following? Justify.
“New requirements are introduced during analysis / design phase”
2. Give one example where waterfall model cannot be applied but incremental model can be applied.
3. Define Petrinets.
4. Differentiate between functional and non-functional requirements.
5. “A module [component] should be open for extension but closed for modification” — Why?
6. Define coupling and cohesion.
7. Differentiate between alpha and beta testing.
8. List the types of testing that address verification process.
9. What is Gantt chart?
10. Identify the type of maintenance for each of the following:
 - (a) correcting the software faults.
 - (b) adapting to the change in environment.

PART B — (5 × 13 = 65 marks)

11. (a) (i) Explain in detail extreme programming process. (6)
- (ii) Which software process model is good for risk management? Explain the model. (7)

Or

- (b) (i) List the principles of agile software development. (6)
- (ii) Consider the process of development of a Word Processing System (like MS Word). Which process model would you prefer to develop the system? Why? Discuss the process model. (7)
12. (a) What is requirements elicitation? Discuss in detail different phases in requirements elicitation process by considering an example of a Watch System with facilities to set and get time / alarm.

Or

- (b) What is SRS? Explain in detail the various components of an SRS.
13. (a) What is software architecture? Describe in detail different types of software architectures.

Or

- (b) What is a component? Explain the steps component level design with classes with illustrations.
14. (a) (i) Compare forward and reverse engineering and give suitable applications. (7)
- (ii) What is black box testing? Design the black-box test suite for the following program. The program computes the intersection point of two straight lines and displays the result. It reads two integer pairs (m1, c1) and (m2, c2) defining the two straight lines of the form $y = mx + c$. (6)

Or

- (b) (i) What is integration testing? Discuss any one method in detail. (7)
- (ii) Perform boundary value analysis and equivalence partition testing for "password field accepts minimum 6 characters and maximum 10 characters". (6)

15. (a) Explain the process of function point analysis. Why is it better than LOC approach? Given the following values, compute function point when all complexity adjustment factor (CAF) and weighting factors are average.

User Input = 50

User Output = 40

User Inquiries = 35

User Files = 6

External Interface = 4

Or

- (b) Describe in detail COCOMO model for software cost estimation. Use it to evaluate a simple project with 12 screens, 10 reports and 80 software components. Assume average complexity and average developer maturity. Use application composition model and object points.

PART C — (1 × 15 = 15 marks)

16. (a) Consider the case of a food ordering system with Supplier, Kitchen, Manager, and Customer entities that interact with the system. Customer can place an Order. The Order Food process receives the Order, forwards it to the Kitchen, store it in the Order data store, and store the updated Inventory details in the Inventory data store. The process also delivers a Bill to the Customer. The Manager can receive Reports through the Generate Reports process, which takes Inventory details and Orders as input from the Inventory and Order data store respectively. The Manager can also initiate the Order Inventory process by providing Inventory order. The process forwards the Inventory order to the Supplier and stores the updated Inventory details in the Inventory data store.

Construct a Context diagram, DFD upto level 1 and data dictionary.

Or

- (b) Consider the following program that finds whether the given number is prime or not. Perform Basis path testing and identify the test cases. Discuss the steps.

```
int main ()
{
int n, index;
cout << "Enter a number : " << endl;
cin >> n ;
index = 2;
while (index <= n-1){
if (n% index == 0) {
cout << "It is not a prime number" << endl;
break;
}
index ++;
}
if (index ==n)
cout << "It is a prime number" <<endl;
} // end main
```