

SOFTWARE ENGINEERING
(INFO 3104)

Time Allotted : 3 hrs

Full Marks : 70

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group - A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**
- (i) When is code review performed during software life cycle?
 - (a) After unit testing
 - (b) After coding and compiling
 - (c) During integration testing
 - (d) During system testing
 - (ii) The purpose of error seeding is which one of the followings?
 - (a) Determine the origin of the bugs
 - (b) Plant trojans
 - (c) Determine the number of latent bugs
 - (d) Plant insidious bugs before delivery to the customer
 - (iii) Which one of the following assertions is true?
 - (a) Code inspection is carried out on tested and debugged code.
 - (b) Code inspection and code walkthrough are essentially synonymous.
 - (c) Adherence to coding standards is checked during code inspection.
 - (d) Code inspection is a code validation techniques
 - (iv) Usability issues are tested during which one of the following levels of testing?
 - (a) Unit testing
 - (b) Integration testing
 - (c) Performance testing
 - (d) Regression testing
 - (v) Which of the following is a black-box testing approach?
 - (a) Path testing
 - (b) Boundary value testing
 - (c) Mutation testing
 - (d) Branch testing
 - (vi) A DFD does not contain
 - (a) External Entity
 - (b) Process
 - (c) Data Store
 - (d) Predicate
 - (vii) Coupling is
 - (a) an intra-module activity
 - (b) an inter- module activity
 - (c) a design-oriented activity
 - (d) a measurement-oriented activity

- (viii) Which of the following is not activity in software process?
(a) Feasibility analysis (b) Software marketing
(c) Installation of software (d) Software debugging
- (ix) The best type of cohesion is
(a) Temporal (b) Sequential (c) Functional (d) Coincidental
- (x) The CMM model defined by
(a) IEEE (b) ISO (c) SEI (d) None of the above

Group- B

2. (a) Write the components of SRS in brief. [(CO1) (Remember/LOCQ)]
(b) Draw an ERD for Hospital Management System showing cardinalities, strong and weak entities, derived attributes, primary key etc. Your diagram should have at least five entities. [(CO3) (Evaluate/HOCQ)]
3 + 9 = 12
3. (a) Differentiate between rapid throwaway prototyping and evolutionary prototyping using suitable diagram. [(CO1) (Remember/LOCQ)]
(b) What do you mean by aggregation and generalization? Explain with a proper example. [(CO2) (Understand/LOCQ)]
(c) What do you mean by context level diagram? Give example. [(CO2)(Remember/LOCQ)]
6 + 3 + 3 = 12

Group - C

4. (a) Draw a Class diagram for online job portal in which job seeker log in the system, then they can view the current job (by location, by job nature, by qualification). After that they can apply for a suitable job. System must send an acknowledgement after being successfully applied. Different employer also can log in the system and they can request to admin to post different new jobs. Admin has the right to add/delete job, change job details etc. You should include different class (with attribute and functionality), relationship between classes, multiplicity, generalization etc. [(CO3) (Analyze/IOCQ)]
(b) Explain that “Loosely coupled and strongly cohesive system is our target”. [(CO2) (Analyze/IOCQ)]
(c) When should we use an activity diagram? [(CO3) (Remember/LOCQ)]
6 + 4 + 2 = 12
5. (a) With an example differentiate between include and extend relationship used in UML diagram. How do we represent private and public data member in class diagrams? [(CO3) (Understand/LOCQ)]

- (b) Draw an activity diagram for Electricity Bill payment system in which user/customer log in the system and they can view the current bill, previous bill and late fine if any. By using the consumer id user can make payment for current/previous bill. Admin can make changes in the bill, Add/Delete connection. User also can apply for a new connection in the system, by viewing that request Admin has to take necessity step on the basis of reconnection as a defaulter. You must include the concept of Special states, Normal States, Swimlanes, Fork and Join. [(CO3) (Evaluate/HOCQ)]
- (4 + 2) + 6 = 12**

Group - D

6. (a) Give an example of a program error that may not cause any failure.
[(CO4)(Understand/LOCQ)]
- (b) Design black-box test suites for a function called *find-intersection()*. The function *find-intersection()* takes four real numbers m_1, c_1, m_2, c_2 as its arguments representing two straight lines $y = m_1x + c_1$ and $y = m_2x + c_2$. It determines the points of intersection of the two lines. Depending on the input values to the function, it displays any one of the following messages:
- (i) single point of intersection
 - (ii) overlapping lines—infinite points of intersection
 - (iii) parallel lines—no points of intersection
 - (iv) Invalid input values. [(CO4) (Create/HOCQ)]
- (c) Asses whether the test suite $\{(x=3,y=2);(x=4,y=3); (x=5,y=1)\}$ is sufficient to test the following code segment:
if (x>y) max = x;
else max = y; [(CO4) (Evaluate/HOCQ)]
- (d) Compare various methods of debugging. [(CO4)(Analyse/IOCQ)]
- 2 + 4 + 2 + 4 = 12**

7. (a) Consider the following C function named *bin-search*:
- ```
/* num is the number the function searches in a presorted integer array arr */
int bin_search(int num) {
 int min,max;
 min =0;
 max =100;
 while(min!=max){
 if(arr[(min+max)/2]>num)
 max=(min+max)/2;
 else if(arr[(min+max)/2]<num)
 min=(min+max)/2;
 else return((min+max)/2);
 }
 return(-1);
}
```

}

Design a test suite for the function bin\_search() that satisfies the following white-box testing strategies (Show the intermediate steps in deriving the test cases):

(i) Condition coverage

(ii) Path coverage. [(CO4) (Evaluate/HOCQ)]

(b) Write down four important coding standards and coding guidelines that you would recommend for any software organization. [(CO4) (Understand /LOCQ)]

(c) What is the McCabe’s Cyclomatic complexity for the following code segment? Derive all the test cases based on the cyclomatic complexity.

```
int f1(int x, int y){
while (x != y){
if (x > y) then
x = x - y;
else y = y - x;
}
```

return x; [(CO4) (Apply/IOCQ)]

$$(3 + 3) + 2 + (1 + 3) = 12$$

### Group - E

8. (a) Draw the PERT diagram for the following set of tasks and dependencies. Also draw the Gantt Chart for the given task with showing critical path. Assume start time = 0.

| Subtask | Time to complete | Dependencies |
|---------|------------------|--------------|
| 1       | 8                | -----        |
| 2       | 10               | -----        |
| 3       | 8                | 1,2          |
| 4       | 9                | 1            |
| 5       | 5                | 2            |
| 6       | 3                | 3,4          |
| 7       | 2                | 4,5          |
| 8       | 4                | 6,7          |
| 9       | 3                | 5,6          |

[(CO6) (Analyse/IOCQ)]

(b) Explain the different types of Metrics used in software project management.

[(CO5) (Analyse/IOCQ)]

$$(4 + 3 + 2) + 3 = 12$$

9. (a) List out the major shortcomings of function point metric in order to use it as a software project size metric. [(CO5) (Analyse/IOCQ)]

(b) Consider a project with four major modules of size 10 KLOC, 10 KLOC, 24 KLOC and 16 KLOC. Use COCOMO to determine development effort, development time. Assume that this project will fall in an organic category. The different cost driver attributes along with their multiplying factors are:

- i) Complexity High : 1.15
- ii) Reliability High : 1.15
- iii) Experience Low : 1.13
- iv) Programmer's Capability Low : 1.17

Find effort and duration required to complete the project.

[(C06) (Analyse/IOCQ)]

- (c) What is Function Point? Give significance of it in software project management.

[(C05)(Analyse/IOCQ)]

**4 + 5 + 3 = 12**

| Cognition Level         | LOCQ  | IOCQ  | HOCQ  |
|-------------------------|-------|-------|-------|
| Percentage distribution | 28.1% | 43.8% | 28.1% |

**Course Outcome (CO):**

After the completion of the course students will be able to

- 1) Prepare software requirement specifications as per IEEE guidelines
- 2) Model function-oriented software systems using DFD
- 3) Develop object-oriented software systems using ERD, UML
- 4) Analyze different approaches of testing methodology in software system.
- 5) Estimate software size using Function Point Analysis.
- 6) Work out software project schedule and staffing plan.

\*LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question; HOCQ: Higher Order Cognitive Question

| Department & Section | Submission Link                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IT                   | <a href="https://classroom.google.com/c/NDA1MjMzOTk4MjE0/a/NDYzNjgwMzYwMDA1/details">https://classroom.google.com/c/NDA1MjMzOTk4MjE0/a/NDYzNjgwMzYwMDA1/details</a> |