

SOFTWARE ENGINEERING

(CSEN 3202)

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) Various notations that are commonly used are, data-flow diagrams, structure charts, decision table, and program design language.
(a) Circles (b) Flowcharts (c) Gantt charts (d) Rayleigh curves.
- (ii) As a design notation, structure charts show the and flow between modules.
(a) control and information (b) circular and data
(c) control and data (d) regular and irregular.
- (iii) Program design language is a language that is used to specify the system design using natural language (English) along with the constructs of a
(a) Hindi language (b) Binary language
(c) Programming language (d) Both (b) and (c).
- (iv) A module is said to have *logical cohesion*, if
(a) it performs a set of tasks that relate to each other very loosely.
(b) all the functions of the module are executed within the same time span.
(c) all elements of the module perform similar operations, e.g., error handling, data input, data output etc.
(d) None of the above.
- (v) High coupling among modules makes it
(a) difficult to understand and maintain the product
(b) difficult to implement and debug
(c) expensive to develop the product as the modules having high coupling cannot be developed independently
(d) all of the above.
- (vi) The deviation of the observed behavior to the specified is called:
(a) Error (b) Fault (c) Failure (d) Defect.

- (vii) When user requirements are not complete and/or technical issues are not clear which model should be followed for software development?
(a) Spiral Model (b) Waterfall Model
(c) Prototyping Model (d) RAD Model.
- (viii) Component testing is also known as:
(a) Alpha testing (b) Unit testing
(c) Ad hoc testing (d) Beta testing.
- (ix) Consider the following pseudo-code:
If (A > B) and (C > D) then
 A = A + 1
 B = B + 1
Endif
The cyclomatic complexity of the pseudo-code is
(a) 2 (b) 3 (c) 4 (d) 5.
- (x) Black-box testing attempts to find errors in which of the following categories?
(a) Incorrect or missing functions (b) Interface errors
(c) Performance errors (d) All the above.

Group – B

2. (a) Suppose you have been appointed as the analyst for a large software development project. Discuss the aspects of the software product you would document in the Software Requirement Specification (SRS) document. How would you validate your SRS document?
- (b) “It is easy for software engineers to develop software according to user requirements even if they are incomplete as software engineers can consider the user requirements of earlier developed software.” Do you agree with this statement? Why or why not? Give reasons in support of your answer.
- (c) What is a prototype? Under what circumstances is it beneficial to construct a prototype? Does construction of a prototype always increase the overall cost of software development?
- 4 + 2 + 6 = 12**
3. (a) What do you understand by the term *phase containment of errors*? Why phase containment of errors is important? How can phase containment of errors be achieved?
- (b) Suggest a suitable life cycle model for a software project which your organization has undertaken on behalf of certain customer who is unsure of his requirements and likely to change his requirements frequently. Give the reasoning behind your answer.
- (c) If a prototyping model is being used in a development effort, is it necessary to develop an SRS document? Justify your answer.
- 6 + 3 + 3 = 12**

Group – C

4. (a) What do you understand by the term functional independence in the context of software design? What are the advantages of functional independence? How can functional independence in a software design be achieved?
- (b) Is it true that whenever you increase the cohesion of your design, coupling in the design would automatically decrease? Justify your answer by using suitable examples.
- (c) Do you agree with the following assertion? “A design solution that is difficult to understand would lead to increased development and maintenance cost.” Give reasoning's for your answer.

(2 + 2 + 2) + 3 + 3 = 12

5. (a) What do you understand by information hiding in the context of software design? Explain why a design approach based on the information hiding principle is likely to lead to a reusable and maintainable design. Illustrate your answer with a suitable example.
- (b) Differentiate between structured analysis and structured design in the context of function oriented design.
- (c) What do you mean by balancing a DFD? Illustrate your answer with a suitable example.

(2 + 2 + 2) + 3 + 3 = 12

Group – D

6. (a) Identify the types of defects that you would be able to detect during the following: (i) Code inspection, (ii) Code walkthrough.

- (b) Consider the following C function named *bin-search*:

```
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 using the following white-box testing strategies (Show the intermediate steps in deriving the test cases):

- Statement coverage

- Branch coverage
- Condition coverage
- Path coverage

$$2 + (2.5 \times 4) = 12$$

7. (a) Consider the following C function named *sort*.
/* *sort* takes an integer array and sorts it in ascending order */

```
void sort () {
    int i, j;
    for (i=0; i < n-1; i++)
        for (j=i+1; j<n; j++)
            if (a[i] > a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
}
```

Determine the cyclomatic complexity of the *sort* function.

Design a test suite for the function *sort* using the following white-box testing strategies (Show the important steps in your test suite design method).

- Statement coverage
- Branch coverage
- Condition coverage
- Path coverage

- (b) Draw the control flow graph for the following function named *find-maximum*. From the control flow graph, determine its cyclomatic complexity.

```
int find-maximum (int i, int j, int k)
{
    int max;
    if (i>j) then
        if (i>k) then max=i;
        else max=k;
    else if (j>k) max=j;
    else max=k;
    return (max);
}
```

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

Group – E

8. (a) Suppose you are developing a software product in the organic mode. You have estimated the size of the product to be about 100,000 lines of code. *Compute the nominal effort and the development time*. The multiplicative factor for this model is given as 2.4 for the software development on organic systems, while the

exponentiation factor is given as 1.05. For estimated development time, the constants are 2.5 and 0.38.

- (b) Suppose you are the project manager of a software project requiring the following activities.

<i>Activity No.</i>	<i>Activity Name</i>	<i>Duration (weeks)</i>	<i>Immediate Predecessor</i>
1	Obtain requirements	4	-
2	Analyze operations	4	-
3	Define subsystems	2	1
4	Develop database	4	1
5	Make decision analysis	3	2
6	Identify constraints	2	5
7	Build module 1	8	3,4,6
8	Build module 2	12	3,4,6
9	Build module 3	18	3,4,6
10	Write report	10	6
11	Integration and Test	8	7,8,9
12	Implementation	2	10,11

- (i) Draw the Activity Network representation of the project.
(ii) Draw the Gantt chart representation of the project.
(iii) Which tasks are on the critical path?

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

9. (a) List three common types of risks that a typical software project might suffer from. Explain how you can identify the risks that your project is susceptible to. Suppose you are the project manager of a large software development project, point out the main steps you would follow to manage risks in your software project.
- (b) Schedule slippage is a very common form of risk that almost every project manager has to encounter. Explain in 3 to 4 sentences how you would manage the risk of schedule slippage as the project manager of a medium-sized project.
- (c) What problems would you face if you are developing several versions of the same product according to a client's request, and you are not using any configuration management tools.
- (d) Suppose you are the project manager of a small team developing a business application. Assume that you have experience in developing several similar products. If you are asked to make a choice between democratic and chief programmer team organizations, which one would you adopt for your team? Explain the reasoning behind your answer.

$$3 + 3 + 3 + 3 = 12$$

Department & Section	Submission Link
CSE	https://classroom.google.com/c/MzY3NTA1Mzk0MjU5/a/MzY5MDk2OTU1MTI5/details