Q.1 a. What is the role of software engineering? (4)

Answer:
Role of software engineering with reference to producing good quality software , maintainable software, and on time within budget.

b. Differentiate between reviews, walkthroughs and inspections. (4)

Answer:
- Inspection process is a formal in-process manual examination of an item to detect bugs. It may be applied to any product or partial product of the software development process. This process is carried out by a group of peers with the help of checklists.
- A walkthrough is less formal, has fewer steps, and does not use a checklist to guide or a written report to document the team's work.
- A technical review is intended to evaluate the software in light of development standards, guidelines, and specifications and to provide management, with evidence that the development process is being carried out according to the stated objectives. A review is similar to an inspection or walkthrough, except that the review team also includes management.

c. List any four software design concepts that span both traditional and object-oriented software development. (4)

Answer: Abstraction, modularity, structural partitioning, information hiding, etc.

d. What is structured programming? (4)

Answer: The structural constructs to limit the procedural design of software to smaller number of predictable operations

e. What is regression testing? Why is regression testing considered a problem for testers? (4)

Answer: The purpose of regression testing is to ensure that bug fixes and new functionality introduced in a new version of a software do not adversely affect the correct functionality inherited from the previous version.

Regression testing is considered as a problem as the existing test suite with probable additional test cases needs to be tested again and again whenever there is modification. The following difficulties occur in retesting:

- Large systems can take a long time to retest
- It can be difficult and time consuming to create the tests.
- It can be difficult and time consuming to evaluate the tests. Sometimes requires a person in the loop to create and evaluate the results.
- Cost of testing can reduce resources available for software improvements.

f. What is software configuration management? (4)

Answer:
It is an umbrella activity applied throughout the software process.
Define Baselines
Define Software configuration items

g. Explain the COCOMO model in brief. (4)

Answer: Constructive cost model
Software cost estimation model implemented in three models: basic, intermediate and detailed.

Q.2 a. Why is software engineering considered a "layered" technology? Explain, with the help of a suitable diagram, the various layers of software engineering. (4)

Answer:

b. What is requirement elicitation? Explain, in brief, the various methods of requirement elicitation. (6)

Answer: Activity that that helps to understand the problem to be solved, requirements are gathered by various methods
Most communication intensive activity of software development.
Name all methods of requirement elicitation

c. Explain the "waterfall" model in detail. In what kind of projects this model is applicable? (8)

Answer:
Waterfall model is a sequential model. Its phases like requirement analysis & specification, design, implementation & unit testing, Integration and system testing, maintenance with diagram

The model is applicable to only those projects where requirements have been settled as it expects complete and accurate requirements.

Q.3 a. What is a data dictionary? Explain its purpose. (4)

Answer:
Repositories to store information about all data items defined in DFD. Define customer data items information including name, aliases, description, related data items, range of values, data structure definition.

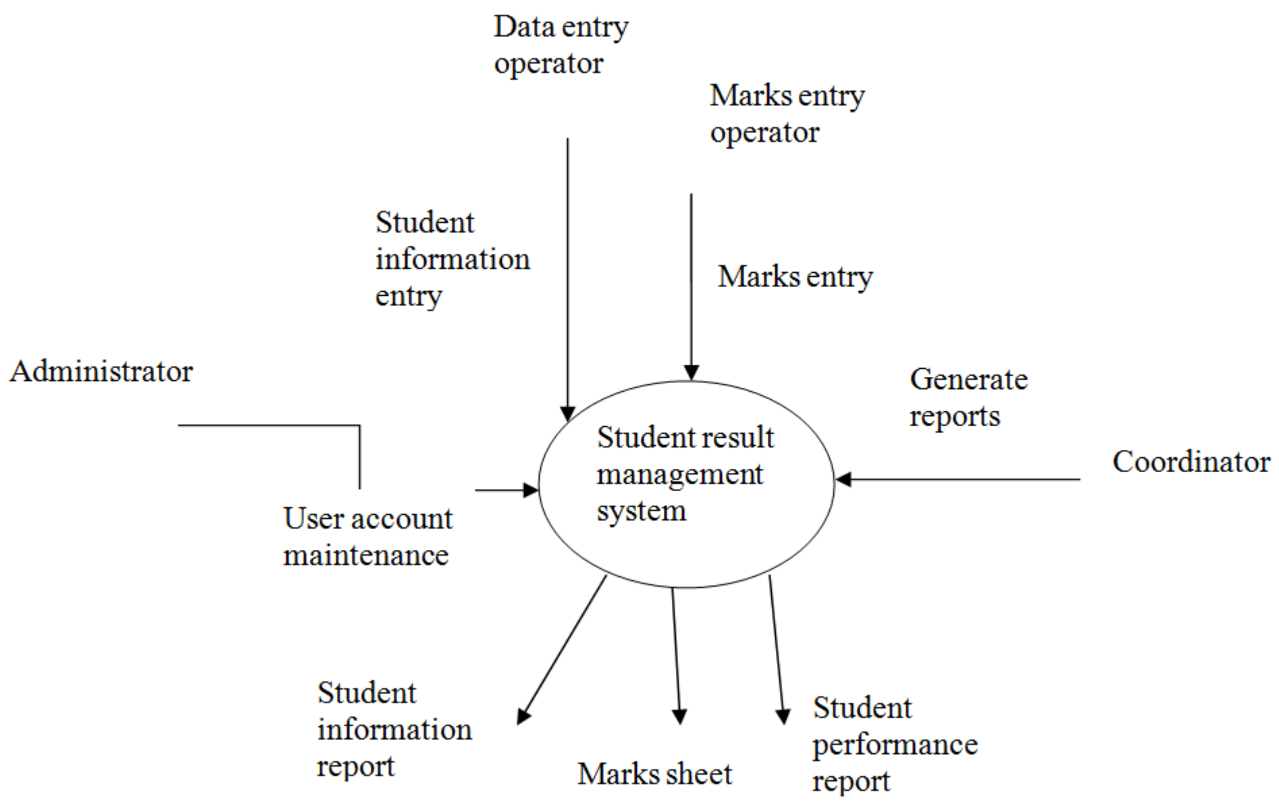
Used to
create an ordered listing of all data items,
create an ordered listing of a subset of data items,
find a data item name from a description,
design the software and test cases.

b. Explain different symbols used for DFD. **(4)**

Answer: symbols for data flow, process, source/sink, data store.

c. The student result management system of an institute needs to be automated. Draw a context diagram for this system. **(10)**

Answer:



Q.4 a. What is a structured chart? What are the various notations used in a structured chart? **(5)**

Answer:
Method used for system design. Partitions a system into black boxes.
Hierarchical format of a structured chart with diagram
Notations for data, module, library module, physical storage, conditional call of module, repetitive call of module

b. What is an interaction diagram in UML? Explain its all notations. What is its role in object oriented design? **(6)**

Answer:
shows an interaction consisting of a set of objects and their relationships including the messages .

Address the dynamic view of the system

Notations for entity object, interface object and control object.

c. What is module cohesion? Identify & briefly explain various levels of module cohesion. (7)

Answer:
Measure of degree to which the elements of a module are functionally related.

Types: Functional, sequential, communicational, procedural, temporal, logical and coincidental.

Q.5 a. Explain any two concepts of object oriented programming. (4)

Answer: Two concepts of object oriented programming like inheritance and polymorphism.

b. What is basis path testing when is it used? Define an independent path. (4)

Answer:
Basis path testing is used for white box testing when there are thousands of paths in a flow graph of a program. Basis path testing provides a reduced set of paths to test the software.

Basis path testing is the technique of selecting the paths that gives a basis set of execution paths through the program.

An independent path is any path through the graph that introduces at least one new set of processing statements or new conditions. An independent path must move along at least one edge that has not been traversed before the path is defined

c. A program takes an angle as input within range [0,360] and determines in which quadrant the angle lies. Design all test cases for this problem using equivalence class partitioning method. (10)

Answer:
1. First we partition the domain of input as the valid input values and invalid values, getting the following classes:

I_1 = {<Angle> : $0 \leq$ Angle ≤ 360}
I_2 = {<A,B,C> : Angle < 0}
I_3 = {<A,B,C> : Angle >0}

The test cases designed from these classes are shown below:

Test Case Id	Angle	Expected Results	Classes Covered by the test case
1	50	I Quadrant	I_1
2	-1	Invalid Input	I_2
3	361	Invalid Input	I_3

2. The classes can also be prepared based on the output criteria as shown below:

O_1 = {<Angle>: First Quadrant, if $0 \leq$ Angle ≤ 90 }
O_2 = {<Angle>: Second Quadrant, if $91 \leq$ Angle ≤ 180 }
O_3 = {<Angle>: Third Quadrant, if $181 \leq$ Angle ≤ 270 }
O_4 = {<Angle>: Fourth Quadrant, if $271 \leq$ Angle ≤ 360 }
O_5 = {<Angle>: Invalid Angle}; However, O_5 is not sufficient to cover all invalid conditions this way. Therefore it must be further divided into equivalence classes as shown below:
O_{51} = {<Angle>: Invalid Angle, if Angle < 0}
O_{52} = {<Angle>: Invalid Angle, if Angle > 360}

Now the test cases can be designed from above derived classes as shown below:

Test Case Id	**Angle**	**Expected Results**	**Classes Covered by the test case**
1	50	I Quadrant	O_1
2	135	II Quadrant	O_2
3	250	III Quadrant	O_3
4	320	IV Quadrant	O_4
5	370	Invalid Angle	O_{51}
6	-1	Invalid Angle	O_{52}

Q.6 a. What is the need of coding standards in an organization? List any four commonly used conventions for naming followed in Java Programming. (6)

Answer:

b. Explain Boehm's model for software maintenance. (6)

Answer: model as a closed loop cycle with diagram.

c. Explain the change control process as a part of software configuration management. (6)

Answer: Change control process flow starting from the need for change to distribute the new version.

Q.7 a. Consider a project with following parameters: EI = 50, EO = 40, EQ = 35, ILF = 06, ELF = 04. Assume all weighing factors are average. In addition system requires, critical performance, average end user efficiency, moderate distributed data processing, and critical data communication. Other GSCs are incidental. Compute the function points using FPA. (8)

Answer:
UFP = 50*4+40*5+35*4+6*10+4*7 = 628

TDI = 4+3+2+4+10*1 = 130

VAF = 130*0.01 +0.065 = 1.365

AFP = 628*1.365 = 857.22

b. How do PERT and CPM helps in project scheduling? (4)

Answer:
program evaluation & review technique
critical path method with appropriate diagrams.
Help in tracking the project with timelines.

c. There are 100 errors estimated to be present in a program. However, 80 errors have been experienced. Use Jelinski-Moranda model to calculate failure intensity with a given Ø = 0.03. What will be the failure intensity after the experience of 80 errors? (6)

Answer:
$N = 100$
$i = 60$
$Ø = 0.03$

Since $\lambda(t) = Ø(N - i + 1)$
$= 0.03\,(100 - 60 + 1)$
$= 1.23$ failures/CPU hr.
After 80 failures
$\lambda(t) = 0.03(100 - 80 + 1)$
$= 0.63$ failures/CPU hr
Hence there is continuous decrease in the failure intensity as the number of failures experienced increases.

TEXT BOOKS

I. Roger Pressman, Software Engineering: A Practitioners approach, Sixth Edition, McGraw-Hill International Edition
II. Ian Sommerville, Software Engineering, Seventh Edition, Pearson Education India
III. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli, Fundamental of Software Engineering, Second Edition, Pearson Education India