

Tutorial questions for CSE206

1. What is software construction?

Software construction involves the practices and tools about design, development, and maintenance of software.

2. Differentiate between software construction, software design, software development, software testing and software engineering.

Software construction involves the practices and tools about design, development, and maintenance of software.

While

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

While

Software development is the process of dividing development work into distinct phases to improve design, product management, and project management.

While

Software testing involves providing a program with a set of inputs (or test cases) and observing if the program behaves as expected.

And while

Software engineering is the application of a systematic, disciplined and quantifiable approach to the development, operation, and maintenance of software.

3. What is activity diagram? And its usefulness.

It is a diagram that is used to describe a use case and can represent parallel relationship between multiple actions. It helps to understand the complex processing activities involving many components.

4. What are the Software Construction activities and related it with software construction languages?

- i. Problem definition.
- ii. Requirement engineering.
- iii. Construction planning.
- iv. Software architecture, or high level design.
- v. Detailed design.
- vi. Unit testing.
- vii. Integration testing.
- viii. Integration.
- ix. System testing.
- x. Corrective maintenance.

5. What are the roles of software testing in software construction?

Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected.

The aim of the testing process is to identify all defects existing in a software product.

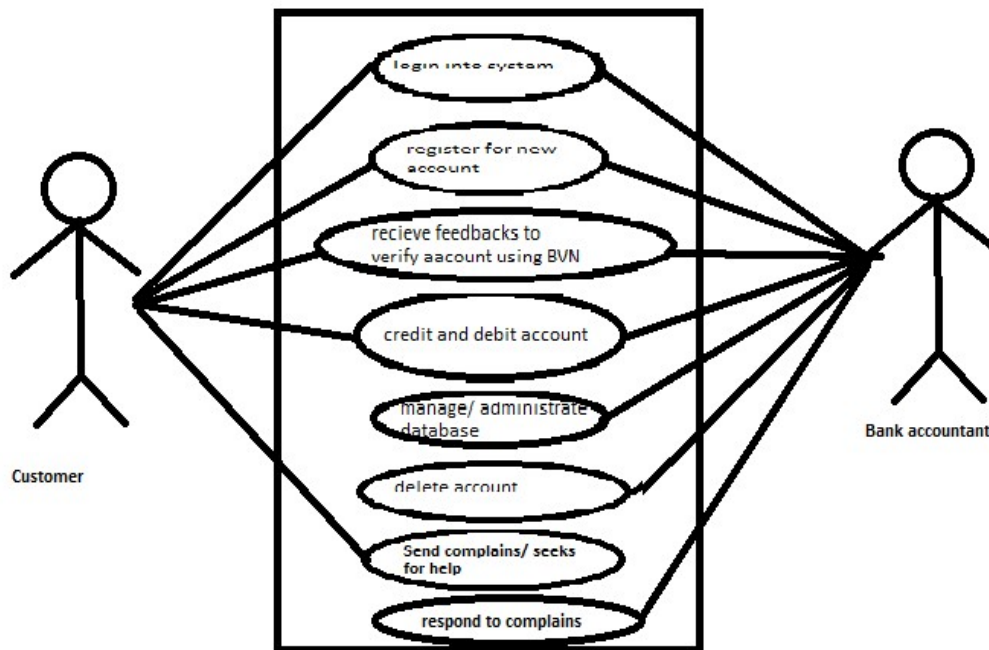
6. List 5 different stages of testing.

- i. Unit (component) testing.
- ii. Module testing,
- iii. Sub-system testing,
- iv. System testing,
- v. Acceptance testing.

7. What is use case diagram?

Use case diagram represent the different ways in which a system can be used by the users.

8. Draw a use case diagrams for the open of new account in the banking sector.



9. What is a software process?

The set of activities which guide software engineers to produce a software product.

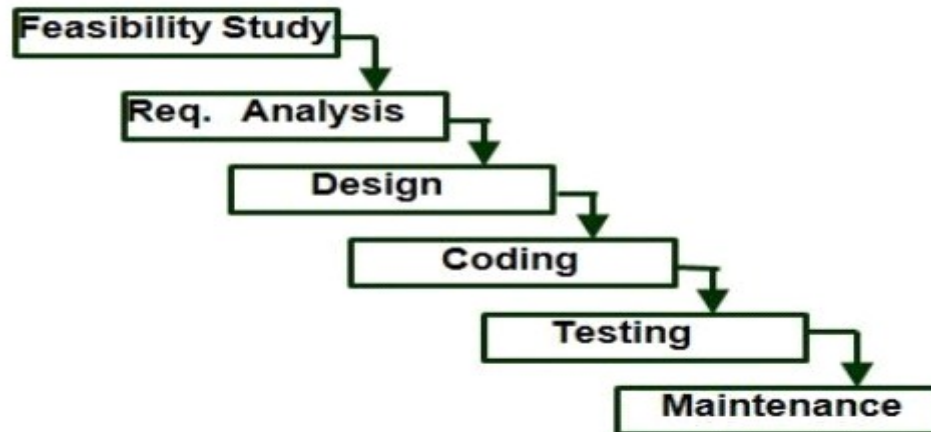
10. List key process activities in software engineering.

- a. Problem definition.
- b. Requirement gathering and analysis.
- c. Design.
- d. Coding and implementation.
- e. Testing and Integration.

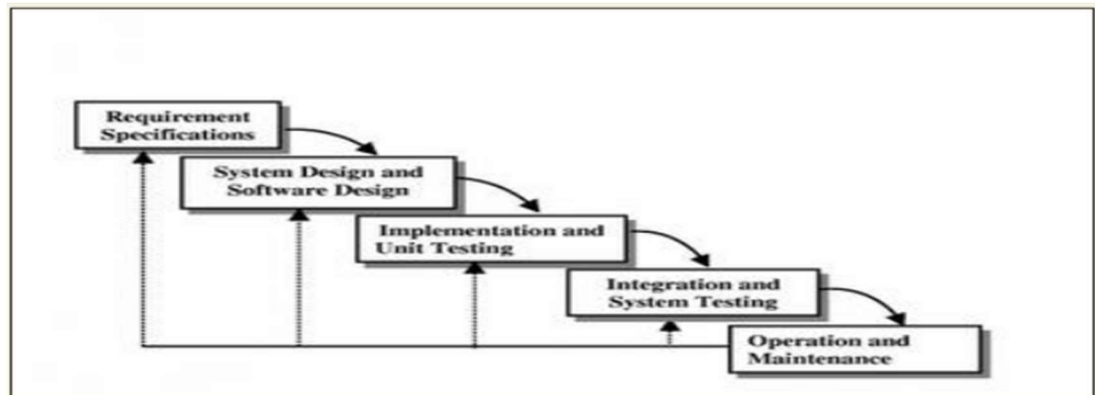
f. Maintenance.

11. Explain 5 software development model?

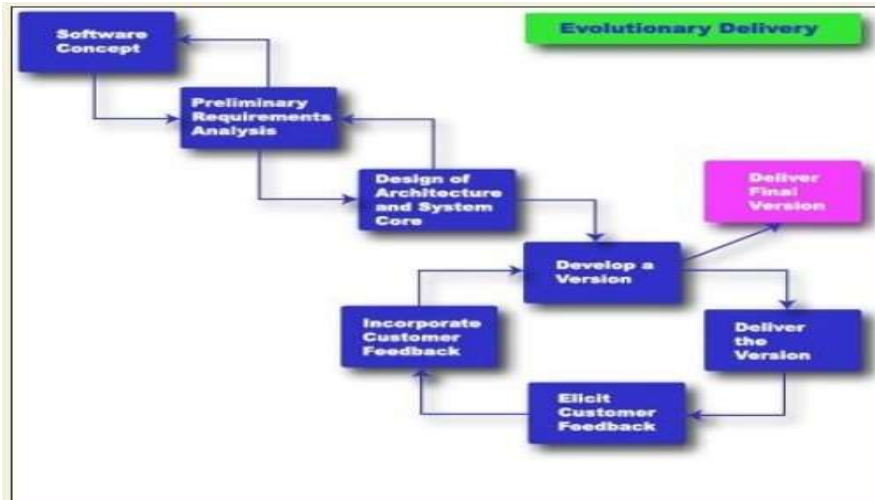
- a. Classical Water fall model: it is the theoretical way of software development and a source of derivation for all other models but it is not a practical model in the sense that it cannot be used in actual software development projects.



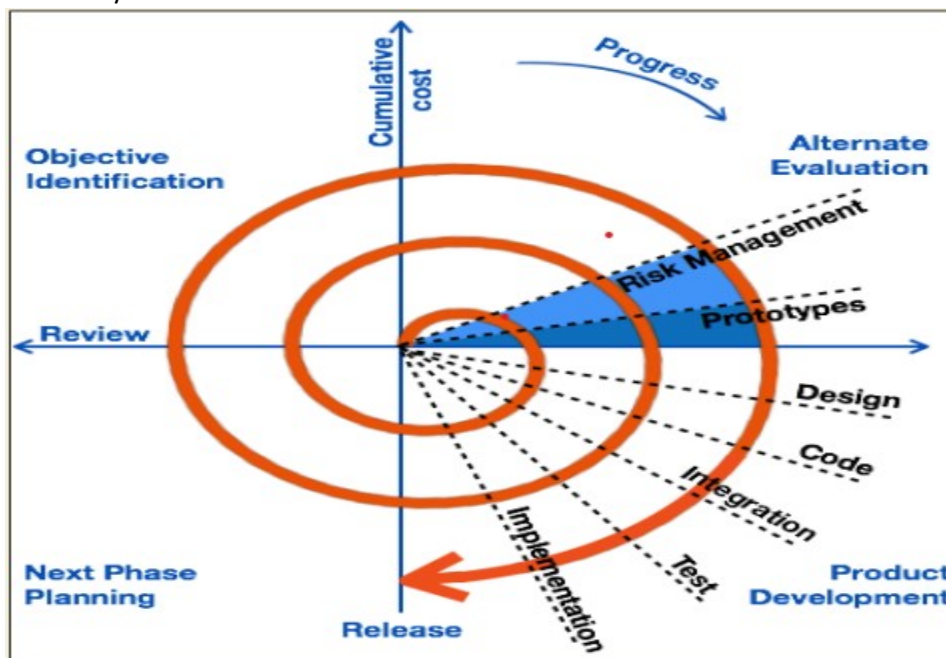
- b. Iterative waterfall model: To overcome the major shortcomings of the classical waterfall model, we come up with the iterative waterfall model. It provides feedback paths for error correction as & when detected later in a phase. The advantage of this model is that there is a working model of the system at a very early stage of development which makes it easier to find functional or design flaws but the advantage is that it can only be applied to large and bulky software development project.



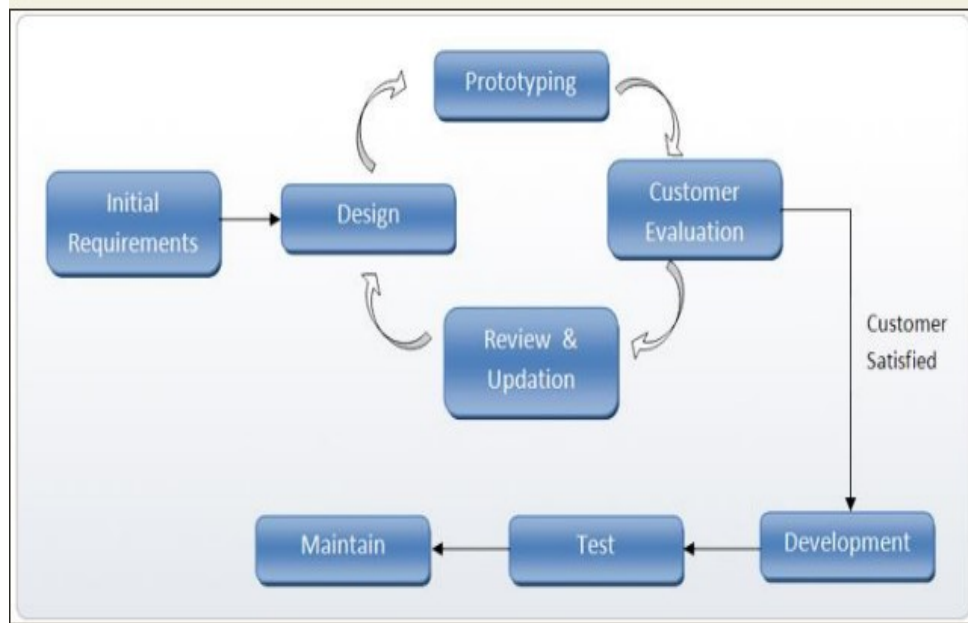
- c. Evolutionary model: It is also called successive version or incremental model. At first, a simple working model is built. Which subsequently undergoes functional improvements and addition of new functionalities till the desired system is built.



- d. Spiral model: it is a Meta model that encompasses all other life cycle models and it has the ability to handle risk.



- e. Prototyping model: a prototype is usually a crude version of the actual system. It is used to illustrate the input data formats, messages, reports, and the interactive dialogues to the customer.



12. What are the difference and relations between Encapsulation, Polymorphism and Inheritance?

Encapsulation meaning Information hiding is the packing of data and function into a single component.	Polymorphism is the provision of a single interface to entities of different types.	Inheritance is a mechanism for code reuse and allow independent extensions of the original software via public classes and interfaces.
--	---	--

13. Define code refactoring in software engineering.

It is the process of restructuring existing computer code without changing its external behavior.

14. What is the usefulness of code refactoring in software construction.

Code refactoring is useful in software construction because it modify or restructure code to a simpler form causing it less time of execution with fewer bugs.

15. What is the relationship between refactoring and testing?

Refactoring is done during testing to make the code more simplified and free of defect.

16. What is software requirement?

It contains the functionalities of the system and its operational constraints.

17. What is the importance of software development requirement?

- System implementation is impossible without software development requirement.
- Developers will not be able to develop the exact system.
- It is difficult for maintenance engineer to understand the system functionalities.
- It will be difficult to write user's manuals properly.

18. Differentiate between functional and non-functional requirements in software construction.

Functional requirement describes how a system should behave at a particular situation, the output it should produce at the particular input.	Nonfunctional requirements describes the operational constraints of the proposed system, such as maintainability, portability and system usability.
--	---

19. Define the following terms: classes, objects, inheritance, design by contract, exception handling, event-based systems, and concurrency

- Classes: a class is a generalized description of an object.

- ii. Object: entities with state and behavior are called objects.
  - iii. Inheritance: involves reusing the state or behavior of a parent class by the child classes
  - iv. Design by contract:
  - v. Exception handling:
  - vi. Event-based system:
  - vii. Concurrency: this involves implementing by splitting the software into multiple independent units of execution, like modules and executing them in parallel, providing capability to the software to execute more than one part of code in parallel to each other.
20. Discuss the concept of abstraction techniques in software construction.
- The principle of abstraction implies that a problem can be simplified by omitting irrelevant details.