1. Define the term software engineering.

Sample Answer:

- An early definition given at the first NATO conference on software engineering (1968): Software Engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

- IEEE standard Glossary of Software Engineering Terminology (1990): Software engineering is the application of systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.


- Object-Oriented and Classical Software Engineering, by S. Schach (1999): Software Engineering is a discipline whose aim is the production of fault-free software, delivered on time and within budget, that satisfies the user's needs. Furthermore, the software must be easy to modify when the user's needs change.

- Software Engineering, by Ian Somerville: Software Engineering is an engineering discipline which is concerned with all aspects of software production.

2. What are the major phases in a software development project?

Sample Answer:

- Requirements Phase (what does the customer need): During the requirements based we would try to understand what the customer needs and document this in the form of requirements document.

- Specification Phase (what to do): Software specification phase is intended to establish what services are required from the system based on the requirements. This is also referred to as the analysis phase.

- Design (how to do): Starting with the specification document, the aim of the design is to determine how to build the system to meet the specifications.
d. Implementation: Various components and modules of the design are coded and unit tested.

e. Integration and Testing: Combining the modules and ensuring that the modules work correctly as specified.

f. Maintenance: This is the most difficult and expensive phase during the lifetime of a software system (taking 50-75% of the efforts).

g. Retirement: This is the phase where the software product does not

3. What are the different kinds of maintenance?

Sample Answer:

(a) Corrective - The repair of actual errors and faults.
(b) Adaptive - Adapting the software to changes in the environment such as new hardware and or a new version of operating system or a database.
(c) Perfective - The changes that the client thinks will improve the effectiveness of the product, such as additional functionality or decreased response time.

4. What is Object-Oriented Programming (OOP)?

Sample Answer:

- Both data and actions are of equal importance.
- Systems is a collection of interacting Objects.
- Object: A software component that incorporates data and the actions that are performed on that data.

5. Discuss various features of OOP.

Sample Answer:

- Data Abstraction:
  - The technique of creating new data types that are well suited to an application.
  - It allows the creation of user defined data types, having the properties of built in data types and more.

- Abstract Data Type (ADT):
  - A structure that contains both data and the actions to be performed on that data. A Class is an implementation of an Abstract Data Type.

- Encapsulation:
  - All information (attributes and methods) in an object oriented system are stored within the object/class.
  - Information can be manipulated through operations performed on the object/class – interface to the class. Implementation is hidden from the user.
  - Object support Information Hiding – Some attributes and methods can be hidden from the user.
• Inheritance:
  o New data types (classes) can be defined as extensions to previously defined types.
  o Parent Class (Super Class) – Child Class (Sub Class)
  o Subclass inherits properties from the parent class.

• Polymorphism:
  o Polymorphic which means “many forms” has Greek roots: Poly – many
    Morphos - forms.
  o It allows a single object, method, operator associated with different meaning depending on the type of data passed to it.

• Persistence:
  o The phenomenon where the object outlives the program execution.
  o Databases support this feature.
  o In Java, this can be supported if users explicitly build object persistency using IO streams.