1. **what is SDLC ?**

**Ans :**

SDLC or Software Development Life Cycle employs a defined methodology to create high-quality software at a low cost and within a short time. It has six stages or phases that provide an organisation with the guidelines to develop well-tested software ready for production and usage. Goals like low costing, high quality and short production timelines are typically considered divergent. SDLC achieves these goals by following a plan that eliminates potential risks incurred in software development processes. The first step in the plan is to evaluate needs and requirements with respect to the deficiencies of existing systems.

Following this, there are stages that allow developers to analyse, plan, design, build, test and deploy the software. Potential errors and pitfalls are accounted for early on, with the help of client or end-user feedback at different stages. This also eliminates redundancy of effort and shortens the timeline of the overall software development process. SDLC puts focus on testing. It is conventionally used as an iterative mechanism, and code quality is put to test with every cycle of development. This means that each cycle delivers some takeaway for the developer or improvement for the product.

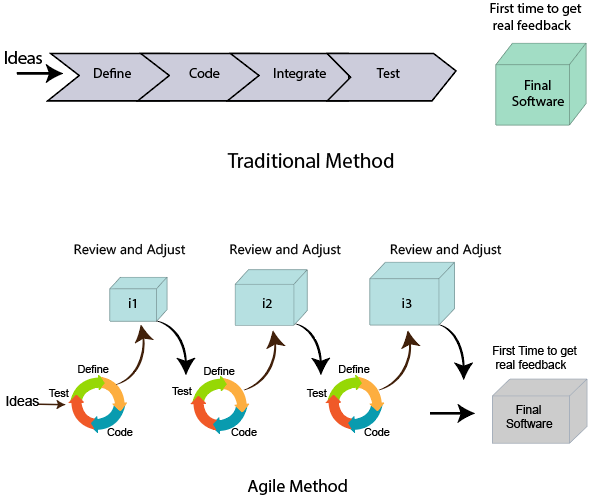
1. **What is agile methodology ?**

**Ans :**

An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risks management factors are predicted initially.

The agile software development process frequently takes the feedback of workable product. The workable product is delivered within 1 to 4 weeks of iteration.



1. **What is SRS ?**

**Ans :**

* SRS stands for “Software Requirement Specification”.

It is a document prepared by business analyst or system analyst.

* Its describes what will be the features of software and what will be its behaviour **i.e** how it will perfrom.
* It is the detail description of software system to be developed with its functional & non – functional requirement.
* The SRS consists of all necessary requirements required for the project development’
* In order to get all details of software from customer and to write the SRS document system analyst is required.
* SRS document is actually an agreements between client & developer.
* **Parts of SRS document**

1. Functional requirementof the system.
2. Non-functional requirement of the system.
3. Goal of implementation.

Charwhatacteristics of SRS document :-

1. Complete : The SRS document must be complete by taking all the requirement to related to software development.
2. Consistent : It should be consistent from beginning to end , so that uses can easily understand the requirement and consistency can be achieved only when there is no conflict between the two requirements.
3. Feasible : All the requirements included in SRS document must be feasible to implement.
4. Modifiable : The SRS document must be created in such a manner that it should be modifiable.
5. Testable : The entire software or the individual module of the software must be testable.
6. Correct : The all the requirements given in the SRS document should be correct so, we can implement in software part.
7. Verifiable : All the document of SRS must be verified.
8. Unambiguous : The module of the software must be unambiguous so one module is compatible with another module.

Note : without proper SRS document it is very difficult to provide the maintenance for the engineers.

1. **What is oops ?**

**Ans :**

Oop stands for object oriented programming , the main purpose of oop is to deal with real world entity using programming language.

* Oops features :-
* Class
* object
* inheritance
* polymorphism
* encapsulation
* abstraction

1. **write basic concepts of oops ?**

**Ans :**

### Class:

A class is basically a combination of a set of rules on which we will work in a specific program. It contains definitions of new data types like fields or variables, operations that we will perform on data (methods) and access rules for that data

### ****Object:****

Objects are defined as an instance of a class. Objects are built on the model defined by the class. An object can be created in memory using the "new" keyword. In C# objects are reference types and other data type variables are value types. In C# objects are stored in the heap while other value types are stored in the stack.

### ****Inheritance:****

Inheritance is relevant due to the concept of “Code Reusability”. Inheritance is a feature by which a class acquires attributes of another class. The class that provides its attributes is known as the base class and the class that accepts those attributes is known as a derived class. It allows programmers to enhance their class without reconstructing it.

### ****Encapsulation:****

Encapsulation is defined as hiding irrelevant data from the user. A class may contain much information that is not useful for an outside class or interface. The idea behind this concept is “Don’t tell me how you do it. Just do it.”.  
So classes use encapsulation to hide its members that are not relevant for an outside class or interface. Encapsulation can be done using access specifiers.

### ****Abstraction:****

Abstraction is defined as showing only the relevant data to the user. Generally abstraction and encapsulation are explained in confusing manners.

**Example:**So just take an example of Mobile Phone Design.

**Relevant Features :**of a phone in which the user is interested are Camera, Music player, Calling function, Voice recording and so on.

**Irrelevant Features :** of a phone in which the user is not interested are circuit board design, hardware used and so on.  
So in designing the Mobile Phone Model, both relevant and irrelevant features are required. But we need to show only relevant features. So we design a Mobile Phone in such a way that only relevant features are shown and irrelevant features are hidden. And remember one thing, that deciding relevant and irrelevant features is totally a user choice.

**Polymorphism:**

Polymorphism is defined as the implementation of the same method with different attributes in a different context.  
**Example :**For example, we have a class named shape with a method name buildshape(). We need to use it in the class Circle, class triangle and the class quadrilateral. So for every class we use the buildshape() method but with different attributes.

1. **What is object ?**

**Ans :**

Object - oriented programming is a programming paradigm based on the concept of “object”

An object can be defined as a data field that lies unique attributes and behaviour.

Real time entity :

Things – product , books

Place – restaurant city

Person – customer

Process

1. **What is class ?**

**Ans :**

Class is a collection of objects and it doesn’t take any space on memory, class is also called as blueprint / logical entity.

User defined : A class which is created by java programmes is called user – defined class ‘

1. **What is encapsulation**

**Ans :**

**What does encapsulation mean**: In object-oriented computer programming (OOP) languages, the notion of encapsulation (or OOP Encapsulation) refers to the bundling of data, along with the methods that operate on that data, into a single unit. Many programming languages use **encapsulation**frequently in the form of **classes**. A **class**is a program-code-template that allows developers to create an object that has both variables (data) and behaviors (functions or methods). A class is an example of encapsulation in computer science in that it consists of data and methods that have been bundled into a single unit.

Encapsulation may also refer to a mechanism of restricting the direct access to some components of an object, such that users cannot access state values for all of the variables of a particular object. Encapsulation can be used to hide both data members and data functions or methods associated with an instantiated class or object.

Benefits of Encapsulation Programming

Encapsulation in programming has a few key benefits. These include:

**Hiding Data:** Users will have no idea how classes are being implemented or stored. All that users will know is that values are being passed and initialized.

**More Flexibility:** Enables you to set variables as red or write-only. Examples include: setName(), setAge() or to set variables as write-only then you only need to omit the get methods like getName(), getAge() etc.

**Easy to Reuse:**With encapsulation it's easy to change and adapt to new requirements.

Encapsulation in Java

Object oriented programming is one of the key pillars of Java. The concepts here refer to object oriented programming as a whole and Java development. To be come a Java developer it is essential to understand the concepts of encapsulation, inheritance, abstraction, and [polymorphism](https://www.sumologic.com/glossary/polymorphism/).

1. **What is inheritance ?**

**Ans :**

Inheritance is an Object Oriented Programming (OOP) feature that allows the properties of an object to be used by different objects.

* A written class can be inherited by another class.
* When this process is done, all properties of the base class are transferred to the new class.
* Reusability of the written code is ensured.
* A base class is the parent class of a derived class.
* The parent class of derived class is called Base Class, and the inherited class is called the Derived Class.

**In Java, three inheritance types supports:**

* Single Inheritance
* Multilevel Inheritance
* Hierarchical Inheritance

**In C++, five inheritance types supports:**

* Single Inheritance
* Multiple Inheritance
* Hierarchical Inheritance
* Multilevel Inheritance
* Hybrid Inheritance

**10 ) What is polymorphism ?**

**Ans :**

Polymorphism is the combination of two greek word one is poly means many and another is morphism is from.

“whose meaning is same object having different behaviour”.

* **Types of polymorphism :**

Compile time polymorphism

Runtime polymorphism

1. Compile time = compile time polymorphism achieved by compile time called compile time polymorphism.

**Ex :** method overloading

1. Run time polymorphism = A polymorphism which is achieved by runtime is called runtime polymorphism.

**Ex :** method overloading

1. **What is RDBMS ?**

**Ans :**

* A database is a set of data stored in a computer. This data is usually structured in a way that makes the data easily accessible.
* A relational database is a type of database. It uses a structure that allows us to identify and access data in relation to another piece of data in the database. Often, data in a relational database is organized into tables.

**Tables: Rows and Columns**

Tables can have hundreds, thousands, sometimes even millions of rows of data. These rows are often called records.

Tables can also have many columns of data. Columns are labeled with a descriptive name (say, age for example) and have a specific data type.

For example, a column called age may have a type of INTEGER (denoting the type of data it is meant to hold).

|  |  |  |
| --- | --- | --- |
| Name | Age | Country |
| Saurabh | 20 | India |
| Arun | 22 | India |

In the table above, there are three columns (name, age, and country).

The name and country columns store string data types, whereas age stores integer data types. The set of columns and data types make up the schema of this table.

The table also has four rows, or records, in it (one each for Saurabh, arun ).

A relational database management system (RDBMS) is a program that allows you to create, update, and administer a relational database. Most relational database management systems use the SQL language to access the database.

SQL (**S**tructured **Q**uery **L**anguage) is a programming language used to communicate with data stored in a relational database management system. SQL syntax is similar to the English language, which makes it relatively easy to write, read, and interpret.

Many RDBMSs use SQL (and variations of SQL) to access the data in tables. For example, SQLite is a relational database management system. SQLite contains a minimal set of SQL commands (which are the same across all RDBMSs). Other RDBMSs may use other variants.

(SQL is often pronounced in one of two ways. You can pronounce it by speaking each letter individually like “S-Q-L”, or pronounce it using the word “sequel”.)

Conclusion

Relational databases store data in tables. Tables can grow large and have a multitude of columns and records. Relational database management systems (RDBMSs) use SQL (and variants of SQL) to manage the data in these large tables. The RDBMS you use is your choice and depends on the complexity of your application.

**12 ) What is SQL ?**

**Ans :**

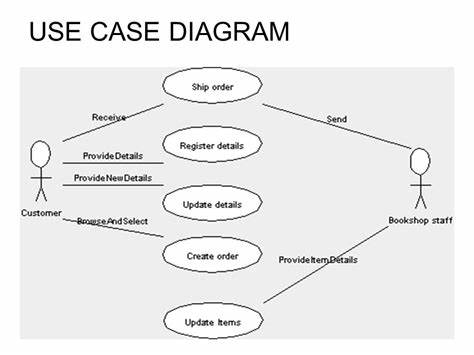
**SQL** stands form “structured query language’

Every rational database software intract with a language known as SQL, because it’s a simple English like language which guidelines are provided by a standard organization ‘ANST’ adopted by all database vendors like oracle, my SQL , Microsoft etc…

An SQL is very complex language to reduce its complexity , it can sub - categorized into 5 languages :-

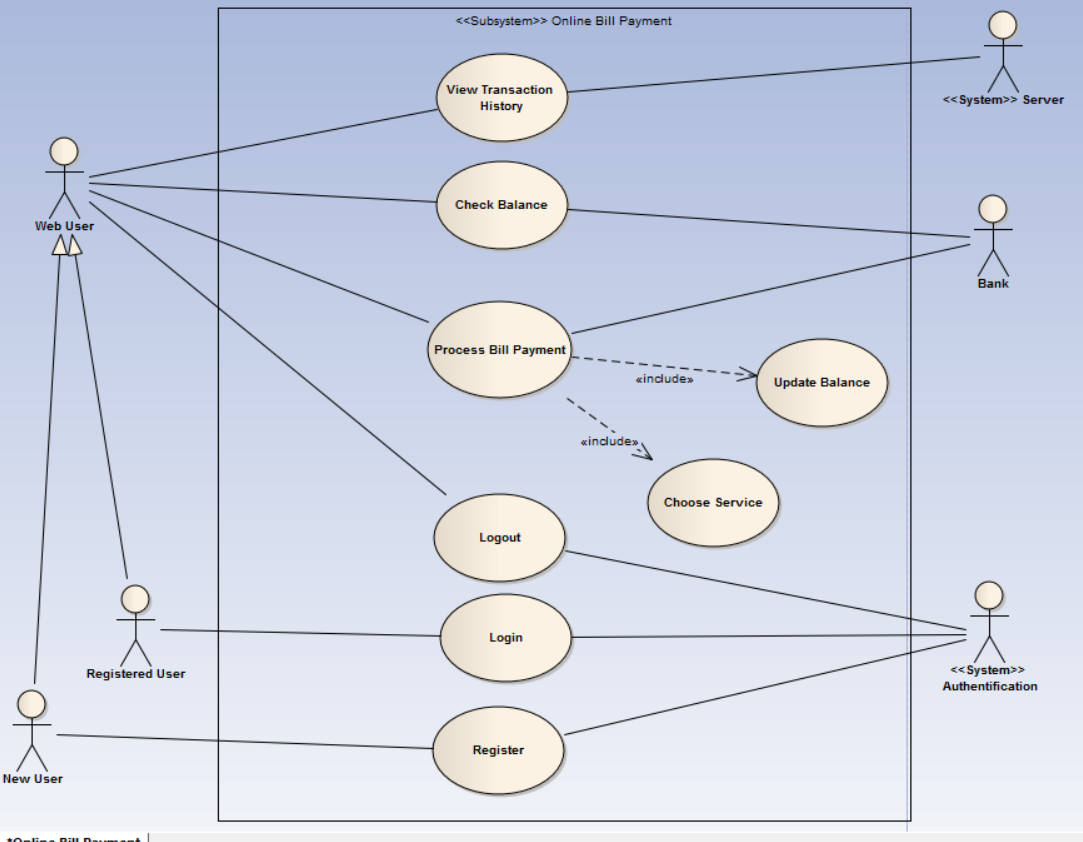
1. **DDL :** Data definition language. (create, drop, truncate, alter, rename.
2. **DML :** Data manipulation language. (insert, update, delete).
3. **DQL :** Data query language (select).
4. **DCL :** Data control language (grant, revoke).
5. **TCL :** Transaction control language (commit, rollback, save point)
6. **) Draw Usecase on Online book shopping ?**

**Ans :**



**14) Draw Usecase on online bill payment system (paytm) ?**

**Ans :**



**15) Write SDLC phases with basic introduction ?**

**Ans :**

**SDLC** stand for ‘software development life cycle’ model point it describes the sequence of phases or steps to develop any software

In single word “entire life time of software from beginning to ending”.

It contains three main stages :

1. Conception stages
2. Implementation stages
3. Maintenance

The SDLC model is classified into three categories based on there advantages ?

* **Disadvantages :**

1. Waterfall model
2. Prototype model
3. Spiral model

**16 ) Explain Phases of the waterfall model ?**

**Ans :**

With the above infographics, we can understand that the waterfall model has a total of 7 phases of the design and development software cycle which are as follows :

1. Requirements
2. Analysis
3. Design
4. Coding / implementation
5. Testing
6. Maintenance

So we can see that the waterfall model works hierarchy from top to bottom with one phase completed with full verifications then switching to another phase including phase processes like Conception, Initiation, Analysis, Design, Coding / implementation, Testing, and Maintenance. In order to get a more brief knowledge about the waterfall model, we need to understand all of its processes in deep with its work model. There is a basic prerequisite phase that needs to be understood before starting the deep phases of knowledge. It is about the feasibility study for the software product. It deals with the financial and technical aspects of the project requirements. This phase deals with the correction of the measures based on the analyzed benefits and drawbacks. Thus the best solution is chosen.

1. **Requirement gathering :** Establish customer needs.
2. **Analysis:** Model and specify the requirement “what .
3. **Design:** Model and specify a solution “why.
4. **Coding / implementation**: Construct a solution in software.
5. **Testing:** Validate the solution against the requirements
6. **Maintenance:** Repair defects and adopt the solution to the new requirements.

**17) Write phases of spiral model ?**

**Ans :**

The spiral model is an SDLC model that combines elements of an iterative software development model with a waterfall model. It is advisable to use this model for expensive, large and complex projects.

In its diagrammatic representation, we have a coil having many cycles or loops. The number of cycles varies for each project and is usually specified by the project manager. Each spiral cycle is a stage in the software development process.

The Spiral Model allows the product to be rolled out and refined in each phase of the spiral, with the ability to build prototypes in each stage. A prototype is created at the beginning of each phase as a risk management technique.

The most important feature of the model is that once the project starts, it has the ability to manage unknown risks. Let’s go through the different phases of the Spiral model first and after that, we would be able to see how risk is handled in this model.

## **Spiral Model Phases**

It has four stages or phases: The planning of objectives, risk analysis, engineering or development, and finally review. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

**Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.

**Risk Analysis and resolving –**In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.

**Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.

**Review and planning of the next phase –**In this phase,the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

* **Spiral Model Advantages**

1. The spiral model is perfect for projects that are **large and complex** in nature as continuous prototyping and evaluation help in mitigating any risk.
2. Because of its **risk handling ability**, the model is best suited for projects which are very critical like software related to the health domain, space exploration, etc.
3. This model supports the client feedback and **implementation of change requests** (CRs) which is not possible in conventional models like a waterfall.
4. Since customer gets to see a prototype in each phase, so there are higher chances of customer satisfaction.

* **Spiral Model Disadvantages**

1. Because of the prototype development and risk analysis in each phase, it is very **expensive and time taking**.
2. It is **not suitable for a simpler and smaller** project because of multiple phases.
3. It requires **more documentation** as compared to other models.
4. Project **deadlines can be missed** since the number of phases is unknown in the beginning and frequent prototyping and risk analysis can make things worse.

**18) Write agile manifesto principles ?**

**Ans:**

The Agile Manifesto is a document that sets out the key values and principles behind the Agile philosophy and serves to help development teams work more efficiently and sustainably. Known officially as ‘The Manifesto for Agile Software Development’, the manifesto detailing 4 Values and 12 Principles.

1. Satisfy Customers Through Early & Continuous Delivery.
2. Welcome Changing Requirements Even Late in the Project.
3. Deliver Value Frequently.
4. Break the Silos of Your Project.
5. Build Projects Around Motivated Individuals.
6. The Most Effective Way of Communication is Face-to-face.

* **What are 4 values and 12 principles of agile?**

The Agile Manifesto is comprised of four foundational values and 12 supporting principles which lead the Agile approach to software development. Each Agile methodology applies the four values in different ways, but all of them rely on them to guide the development and delivery of high-quality, working software.

* **6 scrum principles :**

The six principles are:

* Control over the empirical process. In Scrum, the empirical process is based on observation of hard evidence and experimentation rather than theory.
* Self-organization.
* Collaboration.
* Value-based prioritization.
* Time-boxing.
* Iterative development.

**Which of the Agile Manifesto values is the most important to you why?**

Continuous attention to technical excellence and good design enhances agility. An Agile focus should be on improving the product and advancing consistently. Simplicity — the art of maximizing the amount of work not done — is essential. The goal is to get just enough done to complete the requested project.

Whether you are a newbie or a seasoned veteran, the 3 C’s of User Stories help keep the purpose of the user story in

**What are 3 C’s in user stories?**

Whether you are a newbie or a seasoned veteran, the 3 C’s of User Stories help keep the purpose of the user story in perspective.

* The first C is the user story in its raw form, the Card.
* The second C is the Conversation.
* The third C is the Confirmation.
* **What is the purpose of the Agile Manifesto?**

Satisfy customers through early and continuous delivery – Welcome changing requirements even late in the project Deliver value frequently – Build projects around motivated individuals – Self-organizing teams generate most values

* **What are the 12 principles of the Agile Manifesto?**

The Agile Manifesto is comprised of four foundational values and 12 supporting principles which lead the Agile approach to software development. Each Agile methodology applies the four values in different ways, but all of them rely on them to guide the development and delivery of high-quality, working software.

* **What is the value of the Agile Manifesto?**

Individuals and interactions over processes and tools;

 working software over comprehensive documentation;

 customer collaboration over contract negotiation; and

 responding to change over following a plan.

While there are many very good artifacts, articles and contract models available, we strongly believe that in a world where change is the only constant thing and rapid adaptation to change is the key to success, an Agile contract manifesto, can be one of

**19) What is join ?**

**Ans :**

The purpose of join is to combine the data across tables. A join is actually perform by the where clause which combines the specified rows of the table.

**Note :** If a join involves more than 700 table then join first two table based on join condition and then compares the result with next table and so on.

**20) Write type of joins.**

**Ans :**

**Oracle provided in seven types of join :**

1. equi join
2. non-equi join
3. self join
4. natural join
5. cross join
6. outer join (right outer join) (left outer join) (full outer)
7. inner join

**21) Explain working methodology of agile model and also write pros and cons ?**

**Ans :**

* **Pros:**
* Is a very realistic approach to software development

• Promotes teamwork and cross training.

• Functionality can be developed rapidly and demonstrated.

• Resource requirements are minimum.

• Suitable for fixed or changing requirements

• Delivers early partial working solutions.

* Good model for environments that change steadily.

• Minimal rules, documentation easily employed.

• Enables concurrent development and delivery within an overall planned context.

• Little or no planning required

• Easy to manage

• Gives flexibility to developers

* **Cons:**
* Not suitable for handling complex dependencies.

• More risk of sustainability, maintainability and extensibility.

• An overall plan, an agile leader and agile PM practice is a must without which it will not work.

• Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

• Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

• There is very high individual dependency, since there is minimum documentation generated.

• Transfer of technology to new team members may be quite challenging due to lack of

**22) Draw usecase on Online shopping product using COD.**

Ans : 

**23) Draw usecase on Online shopping product using payment gateway ?**

**Ans :**

