**Placement**

**OOP:**

* Main aim of oop: OOP enables developers to write code that is more modular, reusable, and easier to maintain and modify over time.
* In summary, a class is a template or blueprint that defines the properties and methods that objects of that class can have, while an object is a specific instance of a class that has its own unique set of values for the properties and can perform the operations defined in the class.
* Disadvatnages of oop: Complexity: OOP can be more complex than other programming paradigms, such as procedural programming, due to the use of classes, objects, and inheritance. This can make it harder for beginners to learn and for developers to maintain large codebases. Overhead: OOP can introduce some overhead due to the need to create and manage objects and their associated data structures. This can lead to slower performance compared to other programming paradigms, especially for memory-intensive applications.
* Diamond problem: object-oriented programming (OOP) languages that support multiple inheritance. It occurs when a class inherits from two classes that each inherit from a common base class. This results in a diamond-shaped inheritance hierarchy, where the derived class ends up with two copies of the base class. 2 copies of base class. Ways to solve : virtual inheritance, interface, composition.
* Interface vs abstract: In summary, an abstract class is a class that cannot be instantiated on its own and may have both concrete and abstract methods, while an interface is a collection of abstract methods that must be implemented by any class that implements the interface. An abstract class is used to provide a base class for its subclasses, while an interface is used to define a contract between classes.
* Relationship b/w objects: association, aggregation, composition
* Association: A relationship between two classes where one class uses an object of another class, but there is no ownership involved. It is a "has-a" relationship. For example, a school has teachers. The teachers can exist independently of the school, and they may teach at other schools too.
* Aggregation: A special type of association where a whole object is made up of parts, but the parts can exist independently. It is a "part-of" relationship. For example, a car has wheels, but the wheels can be removed and used on other cars.
* Composition: A stronger form of aggregation where the parts are essential to the whole and cannot exist independently. It is a "composed-of" relationship. For example, a car has an engine, but the engine is an essential part of the car and cannot exist on its own. If the car is destroyed, the engine is also destroyed.
* In summary, association is a "has-a" relationship where there is no ownership involved, while aggregation is a "part-of" relationship where one class owns another class's object, but the object can exist independently.
* Multiple inheritance issue: overriding
* Abstraction: showing only essential info and hiding the details

Link:

https://docs.google.com/spreadsheets/d/1j05cmJ5JI8O9S4bIdKbw\_jd0Y4KMZ\_5liwyL8vjEbCw/edit#gid=0

https://www.w3schools.com/cpp/cpp\_oop.asp

<https://www.guru99.com/difference-between-object-and-class.html>

<https://www.educative.io/answers/what-are-the-solid-principles-in-java>

**DB:**

* 1-1 relation: 1 user 1 cart example
* 1-many 1 cart many products product in cart

**OS:**

**WEB:**

Cookie vs Session: Cookie is just to identify the user. Session id and all the things you do on web stored in cookie which is stored in our browser

To maintain the state b/w web pages we use sessions

Session is generated by the server will be stored in the database . we as a client only received id referred as session id. Cookies are used as the transport medium for the session id as browser will automatically send any cookie with associated with a website

when a malicious site can cause a visitor’s browser to make a request to your server that causes a change on the server. The server thinks that request comes from a user’s cookie the user wanted to submit that form

**CN:**

Transmission Control Protocol – It is used to create a point-to-point communication channel and ensures that the information is delivered error free and in correct order that it was originally transmitted.

**OTHERS:**

* GET and POST are two HTTP request methods used to retrieve or send data from a web server. The main differences between GET and POST requests are:

Data Submission: GET requests are used to retrieve data from a server, while POST requests are used to submit data to a server.

Data Visibility: GET requests pass data as part of the URL, which means that the data is visible in the browser's address bar and can be cached by the browser. POST requests pass data in the request body, which means that the data is not visible in the address bar and is not cached by the browser.

* req is used to receive incoming client requests, while res is used to send server responses back to the client. req contains information about the client request, such as the HTTP method, URL, headers, and any data submitted in the request body, while res contains methods for setting response headers, setting the response status code, and sending data back to the client.
* 1. The Internet is a global network of interconnected computer networks that allows for the exchange of information and communication between devices and users around the world.
* 2. The WWW (World Wide Web) and the Internet are not the same. The Internet is the infrastructure that connects millions of computers and devices together, while the WWW is a system of interconnected web pages and resources that are accessed using a web browser over the Internet.
* 3. The Web works by using a client-server model, where a web browser (client) requests resources (web pages, images, videos) from web servers using HTTP (Hypertext Transfer Protocol) requests over the Internet. The web server responds with the requested resources, which are displayed by the web browser.
* 4. To program for the web, you need to learn web development languages such as HTML, CSS, and JavaScript, as well as server-side languages like PHP, Python, or Ruby. You also need to be familiar with web development frameworks and libraries, as well as web design principles and best practices.
* 5. Web developers need to pay attention to issues such as security, performance, accessibility, user experience, and search engine optimization.
* 6. Web apps are applications that are accessed using a web browser and run on remote servers, while desktop apps are applications that are installed and run on a local computer or device. Web apps are typically easier to develop and maintain, but may have limitations in terms of performance and offline functionality, while desktop apps can provide a richer and more powerful user experience but may require more resources to develop and maintain.

**OOAD:**

1. Creational: How objects are created

* Singleton: type pf object that can only be initiated only once
* Factory

1. Structural: how objects relate to each other

* Façade
* Proxy

1. Behavioral: how objects communicate with each other

* Iterator
* Mediator
* State
* Intro (edu background + family)
* Currently
* Job
* Improve English
* Skills(fields)
* Move abroad

**React:**

* React components help us to write reusable, modular, and better organized code.
* Library vs framework: a tool that provides specific functionality. Framework: set of tools and guidelines for building apps
* Key: to keep track of the list when we add or remove items dynamically react knows what part of the code should be updated
* Hook: function that allows us to tap into built in features in react so usestate is called statehook’’’’’’’’