**Object-Oriented Programming (OOP)**

1. Design Patterns in OOP:

Investigate and analyze various design patterns such as Singleton, Factory, Observer, etc., and evaluate their applications and effectiveness in real-world scenarios.

1. OOP and Functional Programming Integration:

Explore the integration of functional programming concepts (like immutability and higher-order functions) into object-oriented languages, and assess the benefits and challenges.

1. OOP Languages Comparison:

Conduct a comparative analysis of different object-oriented programming languages (e.g., Java, C++, Python) regarding features, performance, and best practices.

1. OOP in Embedded Systems:

Study the application of OOP principles in the development of embedded systems and analyze the impact on code maintainability and system performance.

1. OOP and Artificial Intelligence:

Investigate the role of OOP in the development of AI applications, considering factors like modularity, encapsulation, and adaptability.

**Web Development:**

1. WebAssembly (Wasm):

Explore the applications and performance implications of WebAssembly in web development, focusing on its role in enhancing web application performance.

1. Progressive Web Apps (PWAs):

Research the adoption and impact of Progressive Web Apps, examining their benefits, challenges, and potential future developments in the web development landscape.

1. Serverless Architecture in Web Development:

Analyze the adoption and effectiveness of serverless architecture in building scalable and cost-efficient web applications.

1. Web Accessibility and Inclusive Design:

Investigate the current state of web accessibility, explore emerging technologies for improving accessibility, and propose best practices for inclusive web design.

1. Web Security:

Explore evolving threats in web security, assess the effectiveness of current security practices, and propose innovative approaches or technologies to enhance web application security.