WEEK 1: INTRODUCTION TO SOFTWARE DEVELOPEMENT

-**OVERVIEW OF SDLC**

SDLC(Software Development Life Cycle) . It provides a structured approach to software development, ensuring that all phases are completed in a logical and efficient order.

The SDLC consists of the following phases:

1. **Planning** :
   - Define project scope, goals, and deliverables.
   - Identify stakeholders, resources, and timelines.
   - Develop a project plan and budget

2.**Requirements**:
   - Collect and document user requirements.
   - Define functional and non-functional requirements.
   - Create a product backlog or requirements specification.

3. **Design**:
   - Create a detailed design of the software architecture.
   - Develop user interface (UI) and user experience (UX) designs.
   - Plan the database and data models.

4. **Implementation** (Coding):
   - Write the code according to design specifications
   - Develop the software application.
   - Integrate third-party libraries and services.

5. **Testing** :
   - Plan and execute various types of testing (unit, integration, system, acceptance).
   - Identify and fix defects.
   - Conduct performance, security, and usability testing.

6. **Development** :
   - Plan and execute the deployment of the software.
   - Configure the production environment.
   - Conduct post-deployment testing.

7.**Maintenance** :

- Provide ongoing support and maintenance.
- Fix defects and make updates.
- Refactor code to improve performance and quality.

-SOFTWARE DEVELOPMENT METHODOLOGIES(agile, waterfall, DevOps)

Software development methodologies are frameworks that guide the planning, design, development, testing, and delivery of software applications.

AGILE:
- Iterative and incremental development
- Emphasizes flexibility, collaboration, and rapid delivery
- Sub-methodologies: Scrum, Kanban, Lean, Extreme Programming (XP)

WATERFALL :
- Linear and sequential approach
- Emphasizes predictability, stability, and phase-by-phase completion.

DevOps:
- Focuses on collaboration between development and operations teams
- Emphasizes continuous integration, delivery, and deployment.

-**INTRODUCTION TO PROGRAMMING LANGUAGE C++ AND JAVA**

-C++

C++ is an object-oriented programming language that is viewed by many as the best language for creating large-scale applications. C++ is a superset of the C language.

ADVANTAGES

1. **Performance** : C++ is a compiled language, which means that the code is converted to machine code before execution, making it faster than interpreted languages.

2. **Control**: C++ provides low-level memory management, allowing for fine-grained control over system resources.

3. **Flexibility** : C++ supports multiple programming paradigms, including object-oriented, imperative, and functional programming.

DISADVANTAGES

1. **Complexity**: C++ has a steep learning curve due to its complex syntax, many features, and nuances.

2. **Memory management**: Manual memory management can lead to memory leaks, dangling pointers, and other issues if not handled properly.

3. **Compatibility**: C++ code may not be compatible with newer versions of the language or different compilers.

APPLICATIONS

1. **Operating system** : C++ is used in the development of operating systems, such as Windows and Linux.

2. **Games**: Many games are built using C++, including AAA titles and indie games.

3. **Web browser**: C++ is used in the development of web browsers, such as Google Chrome and Mozilla Firefox.

-**JAVA**

Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

ADVANTAGES

1. **Platform independent** : Java code can run on any platform that has a Java Virtual Machine (JVM) installed.

2. **Object-oriented**: Java supports encapsulation, inheritance, and polymorphism, making it easy to write reusable and maintainable code.

3. **Simple and familiar syntax**: Java's syntax is based on C++ but is simpler and more intuitive.

DISADVANTAGES

1. **Performance** : Java's interpreted nature can make it slower than compiled languages like C++.

2. **Verbose**: Java requires more code than some other languages to accomplish the same tasks.

3. **Limited low-level acess**: Java's abstraction and security features limit direct access to system resources.

APPLICATION

1. **Android Apps**: Java is used to develop the majority of Android apps.

2. **Web Applications** : Java is used in web development, especially with the Spring and Hibernate frameworks.

3. **Enterprise software** : Java is widely used in enterprise software development for its scalability and maintainability.