**NAME:** ABANG, JESSE ONEONYI

**REG. NO:** 202210108513JF

**UNIT:** SOFTWARE

**WEEK 1 SUMMARY REVIEW**

**INTRODUCTION TO SOFTWARE DEVELOPMENT**

Software development is the process of creating, designing, coding, testing, and maintaining applications, systems, or programs. It involves a systematic approach to transform ideas into functional software solutions that meet specific needs.

**OVERVIEW OF SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)**

The Software Development Life Cycle (SDLC) is a framework that outlines the various stages involved in the creation of software. It provides a structured approach to ensure quality and efficiency. While there are different SDLC models, the core phases typically include:

1. **Planning**: This phase involves defining the project scope, goals, and requirements. It includes market analysis, feasibility studies, and creating a project plan.

2. **Requirements Gathering and Analysis:** This phase involves understanding the specific needs and expectations of the users or stakeholders. It includes conducting interviews, surveys, and creating detailed requirements documents.

3. **Design**: This phase involves creating the blueprint for the software, including its architecture, data structures, and user interface. It involves making decisions about the technologies and programming languages to be used.

4. **Development**: This phase involves writing the actual code for the software. It involves implementing the design specifications and creating the functional components.

5. **Testing**: This phase involves identifying and fixing defects in the software. It includes various testing activities such as unit testing, integration testing, system testing, and acceptance testing.

6. **Deployment**: This phase involves deploying the software to the production environment. It includes installation, configuration, and data migration.

7. **Maintenance**: This phase involves ongoing activities to ensure the software continues to function properly. It includes bug fixes, updates, and enhancements.

**SOFTWARE DEVELOPMENT METHODOLOGIES**

Different methodologies are used in software development to manage the process effectively. Some common methodologies include:

1. **Waterfall**: This is a linear model where each phase is completed before moving to the next. It is suitable for projects with well-defined requirements and minimal changes.

2. **Agile**: This is an iterative and incremental model that emphasizes flexibility and customer satisfaction. It involves frequent releases and continuous feedback. Popular Agile frameworks include Scrum and Kanban.

3. **DevOps**: This is a cultural shift that promotes collaboration between development and operations teams. It focuses on automating the software delivery process and improving reliability.

**INTRODUCTION TO PROGRAMMING LANGUAGES: C++ AND JAVA**

C++ and Java are two widely used programming languages in software development.

C++ is a general-purpose, high-level programming language known for its efficiency and performance. It is often used for system programming, game development, and scientific computing. C++ is object-oriented and supports both procedural and generic programming paradigms.

Java is a general-purpose, high-level programming language known for its platform independence and portability. It is widely used for web development, enterprise applications, and Android app development. Java is object-oriented and has a simpler syntax compared to C++.

**TOOLS AND IDE SETUP**

Visual Studio is a popular Integrated Development Environment (IDE) for C++ development. It provides a comprehensive set of tools for coding, debugging, testing, and deployment.

Eclipse and Intelligence IDEA are popular IDEs for Java development. They offer similar features to Visual Studio, including code completion, debugging, and version control integration.

**VERSION CONTROL SYSTEMS (GIT BASICS)**

Version control systems are essential for managing changes to software code. Git is a distributed version control system that is widely used in the software development industry. It allows developers to track changes, collaborate effectively, and revert to previous versions if needed.

**Basic Git Commands:**

git init: Creates a new Git repository.

git add <file>: Adds a file to the staging area.

git commit -m "Commit message": Commits changes to the repository.

git push <remote> <branch>: Pushes changes to a remote repository.

git pull <remote> <branch>: Pulls changes from a remote repository.

By understanding these concepts and tools, I can lay a solid foundation for my software development journey.