What is Software?

Software is a collection of integrated programs that consist of carefully organized instructions and code written by developers using any computer programming language.

What is Engineering?

Engineering is the process of designing and building applications—either in real life or on computers—based on scientific and practical knowledge. It involves:

- Invention
- Design
- Construction
- Maintenance
- Improvement of processes, frameworks, etc.

All with the goal of minimizing cost and achieving the best possible outcomes.

What happens when we combine both?

We get Software Engineering.

Software Engineering:

A branch of engineering concerned with the evolution, development, and production of software products. It applies well-defined scientific principles, techniques, and procedures to ensure quality and efficiency.

What is the result of applying Software Engineering?

An effective and reliable software product.

Why is it important to learn Software Engineering before working in any software-related job?

1. To handle large projects

- Enables better management of complex systems.
- Facilitates collaboration within teams or independent work with minimal issues.

2. To manage cost

- Involves careful planning and clear product requirements.
- Helps avoid unnecessary features and reduces overall development expenses.

3. To decrease development time

- Streamlines development, testing, and deployment.
- Achieved by applying proven software engineering techniques.

4. To deliver reliable software

- Developers are responsible for delivering products on schedule.
- Ensures the ability to detect and resolve post-delivery defects.

5. To ensure effectiveness

- Effectiveness refers to how well the software meets its requirements and aligns with industry standards.

Key Characteristics of Quality Software:

Operational

- Works within budget constraints
- Provides necessary functionality
- Delivers good usability

Transitional

- Adaptable to changes
- Portable across platforms
- Reusable and interoperable

Maintainable

- Easily modified to meet evolving user needs
- Customizable for different contexts
- Maintainability is a core requirement

Efficient

- No unnecessary use of system resources (e.g., memory, processor cycles)
- Fast response and processing times
- Optimized memory utilization

Dependable and Secure

- Ensures system safety and reliability
- Minimizes risk of physical or economic harm in case of failure

Types of Software

1-System Software (Operating Systems)

- Designed to provide a platform for other software to operate.
- Acts as a connecting path between hardware and application software.
- Manages hardware resources and facilitates system-level operations.

Examples:

- macOS
- Linux
- Android
- Windows

2-Interactive / Transition-Based Application Software

- Applications designed to carry out specific tasks for end-users through interaction or transitional data states.
- Commonly used in services that require real-time responses, cloud-based interaction, or user sessions.

Examples:

- Email clients (Outlook, Gmail)
- Remote desktop programs (RDP)
- Online banking applications
- Games and interactive services
- Online video platforms
- Cloud-based platforms (AWS, Azure)
- Shopping apps

Additional Notes:

- These applications often execute on remote servers or cloud infrastructure.
- Accessed via web browsers or client programs.
- Capable of handling large-scale data operations and dynamic user interaction.

3-Scientific and Engineering Software

- Used to facilitate precise computations and real-time scientific/engineering tasks.
- Designed for use in simulations, mathematical modeling, or technical analysis.

Examples:

- Weather prediction software
- Stock market forecasting tools
- Structural stress analysis systems
- Body measurement apps

4-Stand-alone Applications

- Applications that run independently on local machines.
- Do not require an internet or network connection.
- Include all required functionality within the software itself.

Examples:

- Microsoft Office
- CAD software
- Adobe Photoshop, Lightroom
- Offline video games

5-Embedded Control Systems Software

- Software that operates embedded systems used to manage and control hardware devices.
- Common in electronic appliances, vehicles, phones, and hardware interfaces.
- Numerically, embedded systems are the most common type in use.

Examples:

- ABS braking software in vehicles
- Phone button/touch controls
- Mouse/keyboard input handling
- BIOS firmware on motherboards

6-Batch Processing System Software

- Business systems designed to process large amounts of data in predefined batches.
- Suitable for repeated, scheduled, or high-volume processing.

Examples:

- Telecom billing systems
- Payroll software
- Periodic invoice generators

7-Entertainment Systems Software

- Designed for personal use with the primary goal of user entertainment.
- Most of these systems are games, video streaming services, or interactive content platforms.
- The quality of user interaction is the most important distinguishing characteristic.

Examples:

- Netflix
- Crunchyroll
- FIFA
- Battlefield series
- Casino websites

8-Systems for Modeling and Simulation

- Developed by scientists and engineers to model physical processes or complex situations.
- Often involve many interacting components or variables.
- These systems are computationally intensive and typically require high-performance hardware.
- May involve parallel processing for execution.

Examples:

- Weather simulation tools
- Engineering stress analysis
- Fluid dynamics modeling
- Robotic system simulation

9-Data Collection Systems Software

- Collect data from the environment using sensors and send it to other systems for processing.
- Frequently installed in hostile or industrial environments.
- Interacts directly with sensors and external hardware.

Examples:

- Engine monitoring systems
- Car brake sensors
- Fuel tank sensors
- CCTV systems
- Bank security sensors

10-System of Systems Software

- Composed of multiple independent software systems working together.
- Each subsystem performs a specific function but contributes to a larger, integrated system.

Types:

- Generic Software Products Example: Excel, Word, Google Sheets → Built for general use
- Special Software Products → Built specifically for a targeted environment or purpose

11-Artificial Intelligence System Software

- Mostly makes use of non-numerical (symbolic) algorithms to solve complex problems.
- Common application areas:
- Robotics
- Pattern recognition
- Game playing
- Artificial Neural Networks (ANN)
- etc.

Examples:

- TensorFlow
- NumPy
- Pandas
- Scikit-learn