Software testing is a crucial part of the software development process, ensuring that applications function as intended and are free of errors. Two main types of testing are static testing and dynamic testing, both of which contribute to overall software quality.

**What is Static Testing?**  
 Static testing involves reviewing the code, documentation, or design without running the program. It focuses on finding errors, ensuring coding standards are followed, and identifying vulnerabilities before execution. Techniques such as code reviews and walkthroughs help identify issues early, preventing costly fixes later in development.

**What is Dynamic Testing?**  
 Dynamic testing, on the other hand, involves running the software to observe how it performs in real-world conditions. This testing reveals issues like memory leaks, crashes, or incorrect outputs. It includes unit tests, integration tests, and user acceptance tests, each ensuring the software behaves as expected during execution.

**Differences Between Static and Dynamic Testing**  
 The key difference is simple: static testing reviews code without execution, while dynamic testing requires running the program. Static testing is usually done early in the development process, focusing on errors in the code itself. Dynamic testing happens later, ensuring the software works properly during execution.

**Why Use Both?**  
 Static and dynamic testing each offer unique advantages. Static testing catches issues early, saving time and resources, while dynamic testing ensures the software functions correctly in real-world scenarios. Using both methods provides a comprehensive approach, improving overall quality and reliability. Static testing prevents problems before they arise, and dynamic testing ensures the program works as intended when users interact with it. Together, they help create better, more dependable software.