Brooke Slampak

CS-320

**Module Two Journal**

Testing is an important part of making sure software works as expected, and there are two main ways to do it: static testing and dynamic testing. Static testing is checking the code and documents without actually running the program. It includes code reviews and using tools that scan the code for mistakes or missing parts. Since the code isn’t executed, static testing helps catch problems early, like typos, design flaws, or things that don’t match the requirements.

Dynamic testing, on the other hand, involves running the software to see how it behaves for real. This includes unit tests, integration tests, and system tests where the program is put through different scenarios to check if it works as expected. Dynamic testing helps find bugs that only show up when the software is running, like crashes or slowdowns, which static testing can’t catch.

The main difference between the two is that static testing happens without running the code, usually early on, and focuses on code quality and structure. Dynamic testing happens later, after some coding is done, and focuses on making sure the software works correctly in practice. Both have their own tools and techniques, but they complement each other.

Using both static and dynamic testing is important because they catch different kinds of problems. Static testing helps find issues early before they become costly to fix, while dynamic testing makes sure the software works well in real-world situations. Together, they give a fuller picture of how well the software has been developed and helps make sure the final product is reliable and high quality.