Understanding Software Construction and Collaboration

GROUP-TEC (BSCS 3:2)

Name	Reg No.
Okidi Norbert	S23B23/051
Najjuma Teopista	S23B23/041
Tendo Calvin	S23B23/013
Ezamamti Ronald Austine	S23B23/018
Kiisa Angela	S23B23/027
Buwembo David Denzel	S23B23/074

1. **Explain the difference between programming and software construction using one real-world example.**

Programming refers to the act of writing code to handle a specific task or problem using a particular programming language.
While:
Software construction is a stage beyond programming that involves designing, organizing, testing, documenting, and maintaining code in a structured way to ensure the software remains reliable, understandable, and adaptable over time. It considers teamwork, and future modifications.
An-example: A programmer writes code to calculate account balances after deposits and withdrawals. While in Software Construction, the team designs the system with secure authentication, modular transaction services, logging, testing, and documentation so it can support millions of users and future features like mobile banking.

2. Describe a situation where poor maintainability could cause serious problems?

Poor software maintainability can cause serious problems in a financial banking transaction system.

Banking systems handle high volumes of sensitive transactions such as fund transfers, loan processing, and account balance updates. If the software is poorly maintainable, characterized by tightly coupled components, outdated frameworks, and inadequate documentation implementing regulatory updates or security patches becomes slow and error-prone.

This can result in transaction failures, incorrect account balances, or unpatched security vulnerabilities that expose customer data to fraud. Additionally, poor maintainability hinders collaboration between development, security, and compliance teams, making it difficult to respond quickly to financial regulations or cyber threats.

In this context, poor maintainability can lead to financial losses, legal penalties, loss of customer trust, and long-term damage to the institution's reputation.

3. Why is version control critical in team-based development ?

Without version control like git:

(i) People overwrite each other's code.
(ii) Bugs cannot be traced.
(iii) No backup of previous working versions
(vi) Collaboration becomes cha

In a real team using Git:

(i) Each developer works on their own branch
(ii) Changes are reviewed before merging
(iii) You can go back to any previous version
(iv) You can see who wrote which line of code
(v) Multiple people can work on the same project safely
(vi) Deployment pipelines connect directly to the repository

For instance:

- Two developers editing the same file:
- Without Git → one deletes the other's work. With Git → merge conflict is resolved safely.
- Version control is the bedrock of collaboration in software construction.

4. Describe how code reviews improve both software quality and developer skills.

Code reviews improve software quality by allowing developers to examine each other's code before it is merged into the main project (commonly in git branches) . This process helps identify bugs, conflicts, security issues, and poor design choices early.

They also improve developer skills because:

- Developers learn better coding practices from peers.
- Feedback encourages consistency and readability.
- Junior developers gain insights from experienced team members.
- Teams build shared standards and trust.

As a result, the software becomes more reliable, and developers grow technically and professionally.

5. Reflect briefly on how AI can help in understanding code without replacing learning.

AI can assist developers by explaining code logic, suggesting improvements, and helping identify errors. It acts as a learning support tool rather than a replacement for understanding.
For example, one can use AI to explain how a function works, clarify complex logic, or understand why an error occurs. However, there's still a need to think critically, practice coding, and apply concepts independently.

When used responsibly and ethically, AI enhances learning by speeding up understanding and reducing confusion, while still requiring one to actively engage with the material.