Brian Preston

7/27/2025

CSD-430

**The Power of Coding Standards in Software Development**

As I have progressed through my degree in Business Software Development, I have come to really understand how foundational coding standards are in the real-world programming environment. I used to think writing code was mostly about just getting something to work. But now, I realize that how we write code matters just as much as what the code does. I believe coding standards are important because they hold us accountable for our work and ensure that what we write can be understood, reused, and maintained by others down the line.

What exactly are coding standards? According to Codacy (2025), “Coding standards, also known as coding guidelines or programming style guides, are rules and conventions that developers follow when writing code.” These standards go beyond personal preference—they create a uniform codebase and allow teams to collaborate more effectively. I think of them like a team’s shared language. Everyone might have their own voice, but we can build something much greater together if we all speak the same language.

In my experience so far, the more I have learned about version control, team collaboration, and debugging, the more I have realized how coding standards make all these easier. They do not just make the code pretty—they make it secure, scalable, and reliable. Codacy reports that “the estimated cost of poor software quality in the U.S. in 2022 was at least $2.41 trillion.” That’s a staggering number, and I feel that adopting coding standards is a direct way to help reduce that cost.

I believe one of the best examples of how I have personally applied coding standards is in my Module 8 project for this class. This assignment required me to use a JavaBean to interact with a database and display dynamic form data using JSP. At first, it felt very overwhelming, but once I broke it down, I realized how much the standards I had learned made the code easier to write, debug, and maintain.

Here’s a simplified version of a method I used that follows clean naming conventions, concise logic, and error handling:

java

public Book getBookById(int bookId) {

Book book = null;

try {

Connection conn = DatabaseConnection.getConnection();

String sql = "SELECT \* FROM books WHERE id = ?";

PreparedStatement stmt = conn.prepareStatement(sql);

stmt.setInt(1, bookId);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

book = new Book();

book.setId(rs.getInt("id"));

book.setTitle(rs.getString("title"));

book.setAuthor(rs.getString("author"));

}

rs.close();

stmt.close();

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

return book;

}

This method checks so many of the boxes we talked about in our lessons and in the readings: meaningful method and variable names, indentation, reusability, proper error handling, and a clear structure that makes the logic easy to follow. I didn’t realize it at the time, but I was following standards that would make this code understandable to any other developer joining the project.

I feel proud looking back on this because it shows how far I have come. At the start of this course, the idea of using a JavaBean felt intimidating. But now I understand how these standards are not about adding extra work—they are there to make the code run smoother. If I had ignored them, this code would have been messy, hard to troubleshoot, and nearly impossible for someone else to understand.

Another quote that really stood out to me from Codacy was, “Coding standards enhance developer productivity and creativity rather than restrict them.” I believe that’s true. Standards give developers the mental freedom to focus on logic and functionality instead of constantly second-guessing how things should be written. From a compliance perspective, coding standards also play a critical role in security and regulations. Codacy (2025) points out that standards help with laws like GDPR, PCI DSS, and HIPAA. I think this is one of the most overlooked yet important reasons to follow them. It’s not just about style—it’s about safety.

The Manifestly Checklists article also makes a great point about how using checklists and tools like linters or pre-commit hooks can help enforce consistency across teams. I think having something like that in place is essential, especially when working with larger groups or distributed teams. It ensures that standards aren’t just ideas—they’re habits. GeeksforGeeks (2024) also highlights how good coding standards promote readability, maintainability, and efficiency. Their guidelines on naming conventions, indentation, and limiting global variables were echoed in a lot of the feedback I received during code reviews. I believe those small habits added up to a massive improvement in my work quality.

To wrap things up, I believe coding standards are a must-have in any modern development environment. They build a culture of quality, help teams onboard faster, reduce bugs, and ensure we’re building software that lasts. As someone who’s gone from beginner to someone who now understands why clean code matters, I feel grateful for how much I learned. Going forward, I will continue to follow the lessons I have picked up—especially the value of writing code that someone else can understand and trust.

References

Codacy. (2025, March 14). Coding standards: What are they and why are they important?https://blog.codacy.com/coding-standards

GeeksforGeeks. (2024b, May 23). Coding standards and guidelines. <https://www.geeksforgeeks.org/software-engineering/coding-standards-and-guidelines/>

Manifestly Checklists. (2025). Coding standards checklist. <https://www.manifest.ly/use-cases/software-development/coding-standards-checklist>