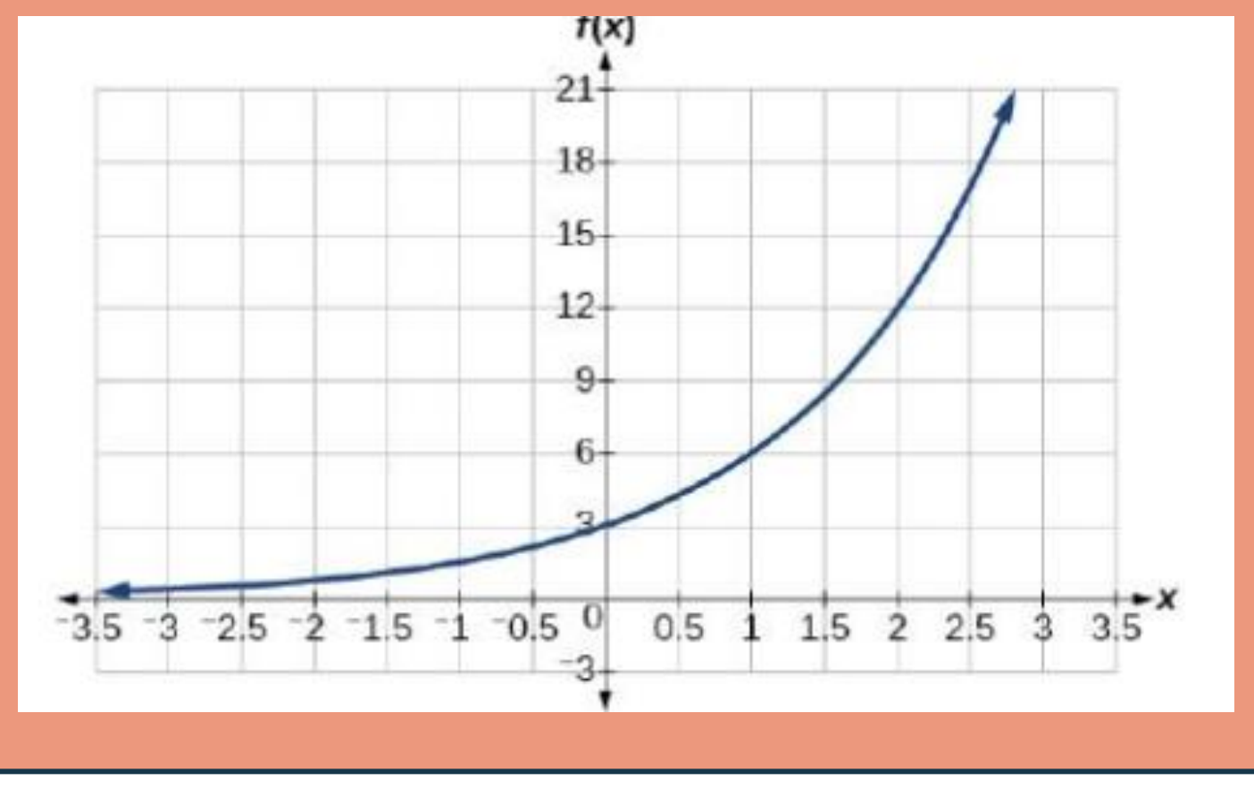
SOEN 6011 : Software Engineering Processes

F5: Harshavardhana Mudduluru, Sanjay Bhargav Pabbu, Dev Bhupendra Pandya

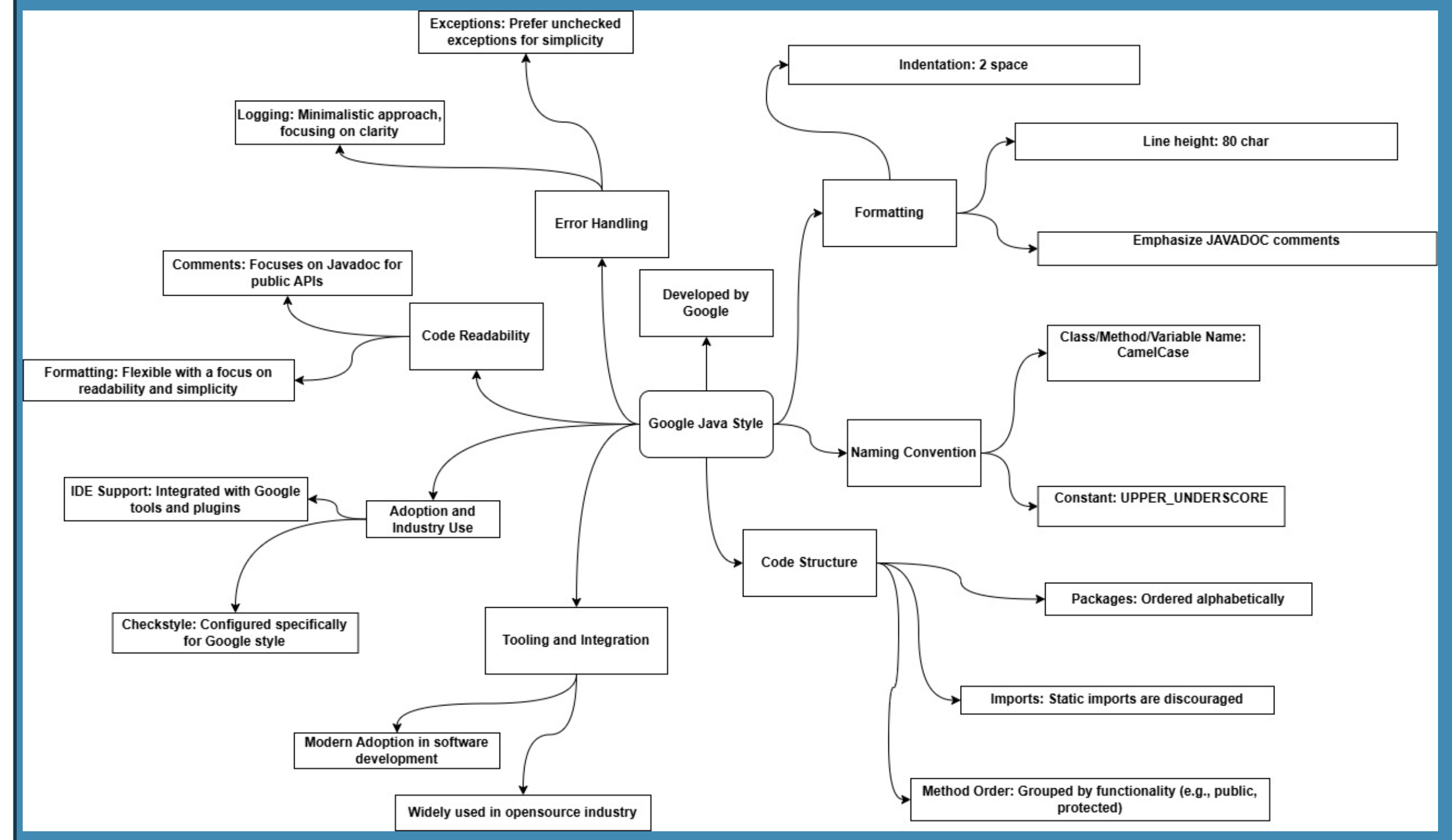
Concordia University, Montreal, Quebec

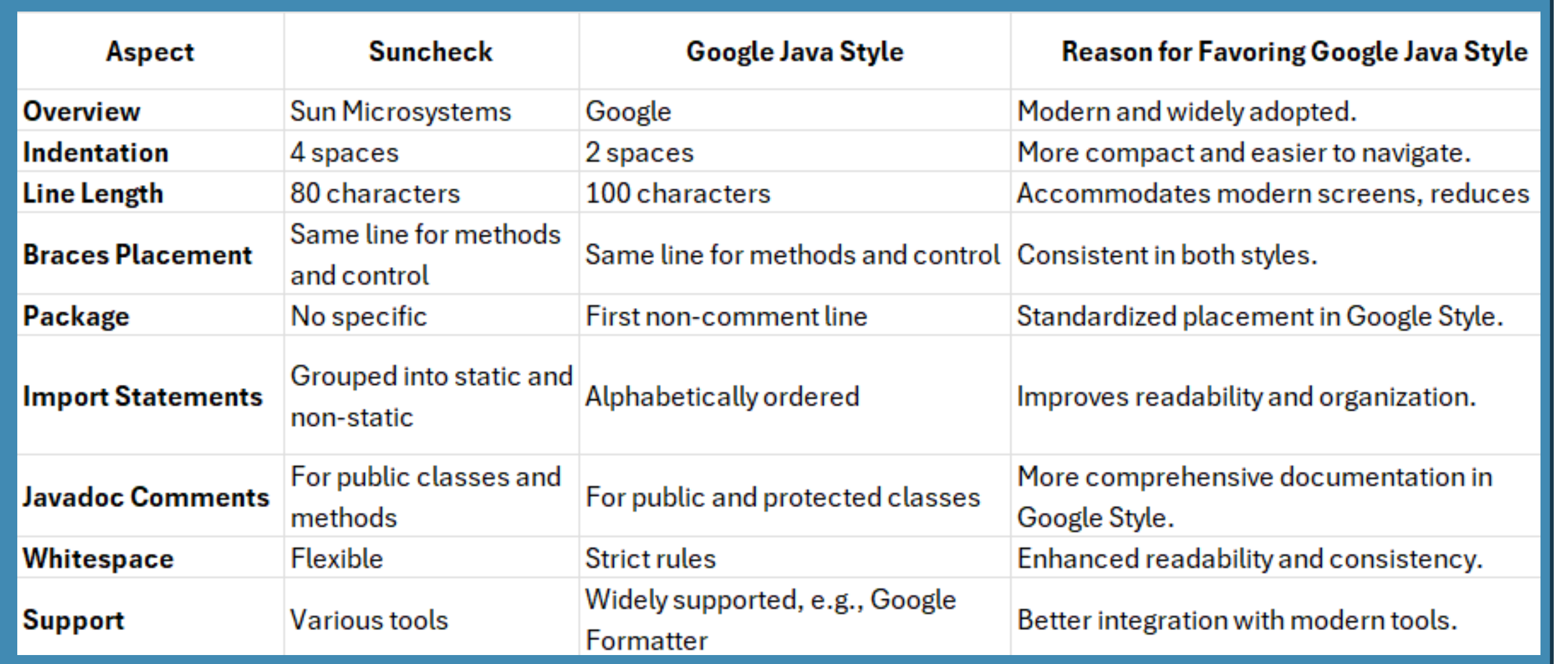
F5 : ab^x (Exponential Function)

- This function is widely used in exponential growth and decay models, such as population growth, radioactive decay, and compound interest calculations.
- The objective of Deliverable 3 is to refine the implementation of the application by ensuring that it conforms to established Java programming styles, utilizes effective debugging and static analysis tools, and adheres to semantic versioning.



Programming Style: Java Google Style

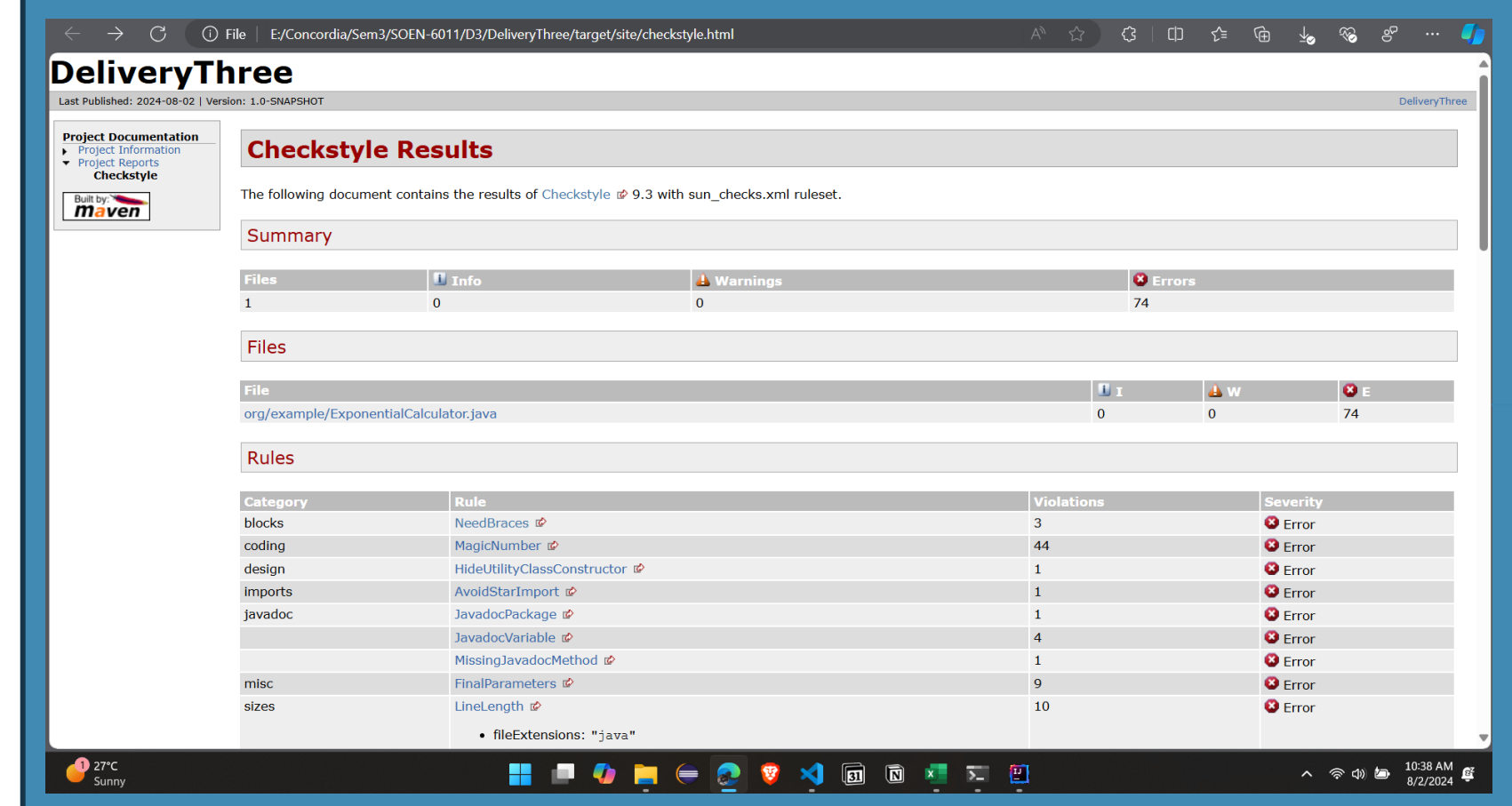

Google Java Style
Developed by Google
Error Handling
Exceptions: Prefer unchecked exceptions for simplicity
Logging: Minimalistic approach, focusing on clarity
Code Readability
Comments: Focuses on Javadoc for public APIs
Formatting: Flexible with a focus on readability and simplicity
Adoption and Industry Use
IDE Support: Integrated with Google tools and plugins
Checkstyle: Configured specifically for Google style
Tooling and Integration
Modern Adoption in software development
Widely used in opensource industry
Formatting
Indentation: 2 space
Line height: 80 char
Emphasize JAVADOC comments
Naming Convention
Class/Method/Variable Name: CamelCase
Constant: UPPER_UNDERSCORE
Code Structure
Packages: Ordered alphabetically
Imports: Static imports are discouraged
Method Order: Grouped by functionality (e.g., public, protected)

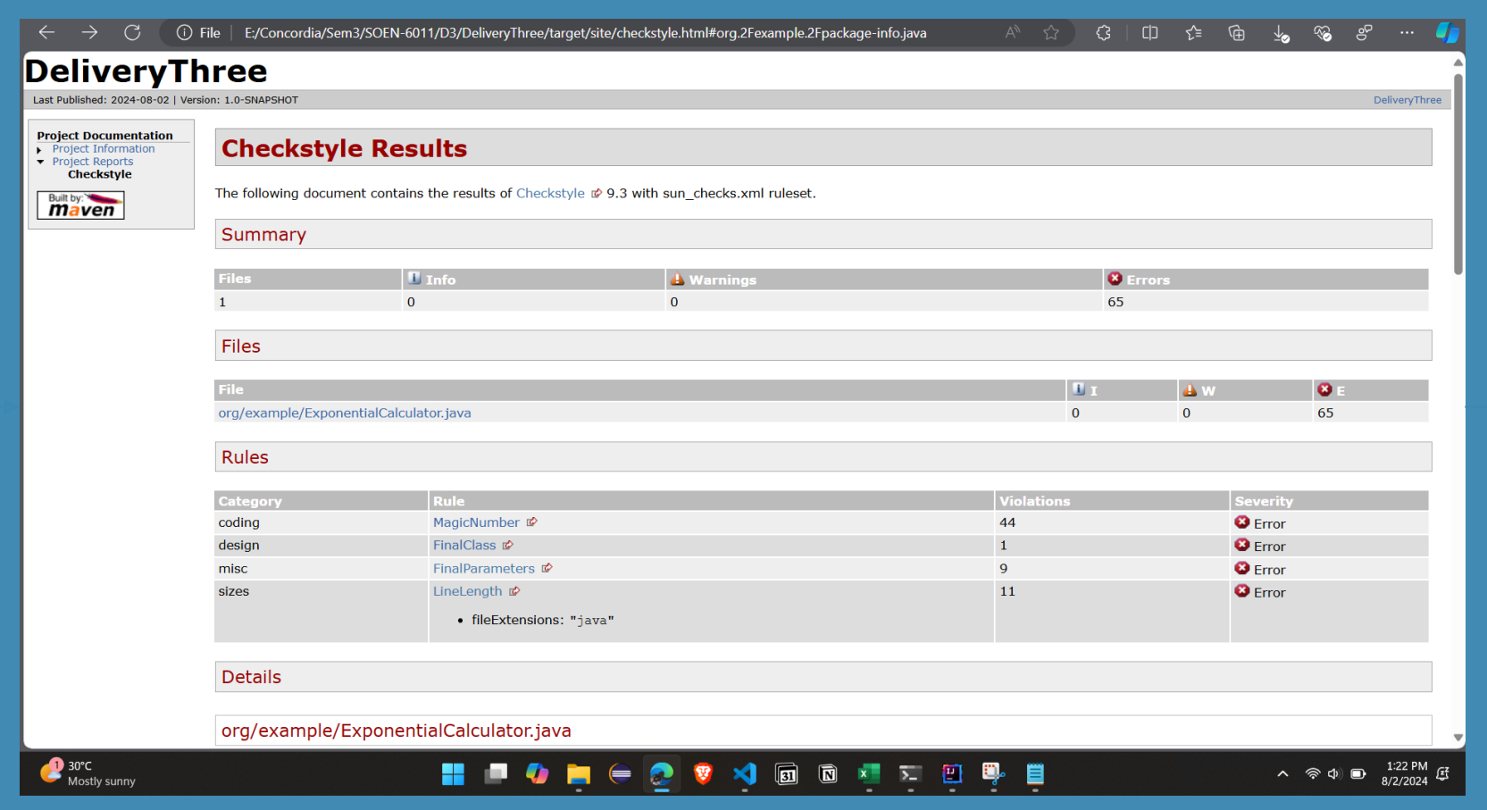


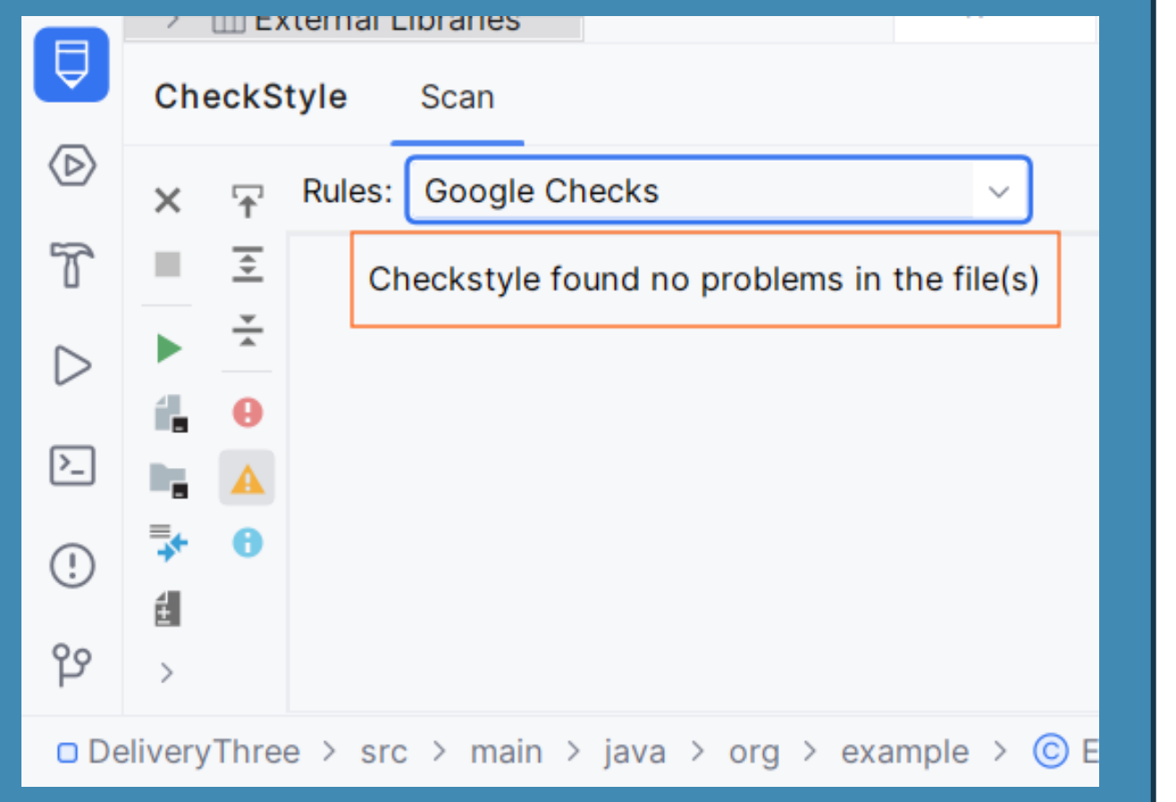
Aspect	Suncheck	Google Java Style	Reason for Favoring Google Java Style
Overview	Sun Microsystems	Google	Modern and widely adopted.
Indentation	4 spaces	2 spaces	More compact and easier to navigate.
Line Length	80 characters	100 characters	Accommodates modern screens, reduces
Braces Placement	Same line for methods and control	Same line for methods and control	Consistent in both styles.
Package	No specific	First non-comment line	Standardized placement in Google Style.
Import Statements	Grouped into static and non-static	Alphabetically ordered	Improves readability and organization.
Javadoc Comments	For public classes and methods	For public and protected classes	More comprehensive documentation in Google Style.
Whitespace	Flexible	Strict rules	Enhanced readability and consistency.
Support	Various tools	Widely supported, e.g., Google Formatter	Better integration with modern tools.

These factors make Google Java Style a preferable choice for maintaining high code quality in modern Java projects.

Programming Style: Checkstyle






CheckStyle Scan

Rules: Google Checks

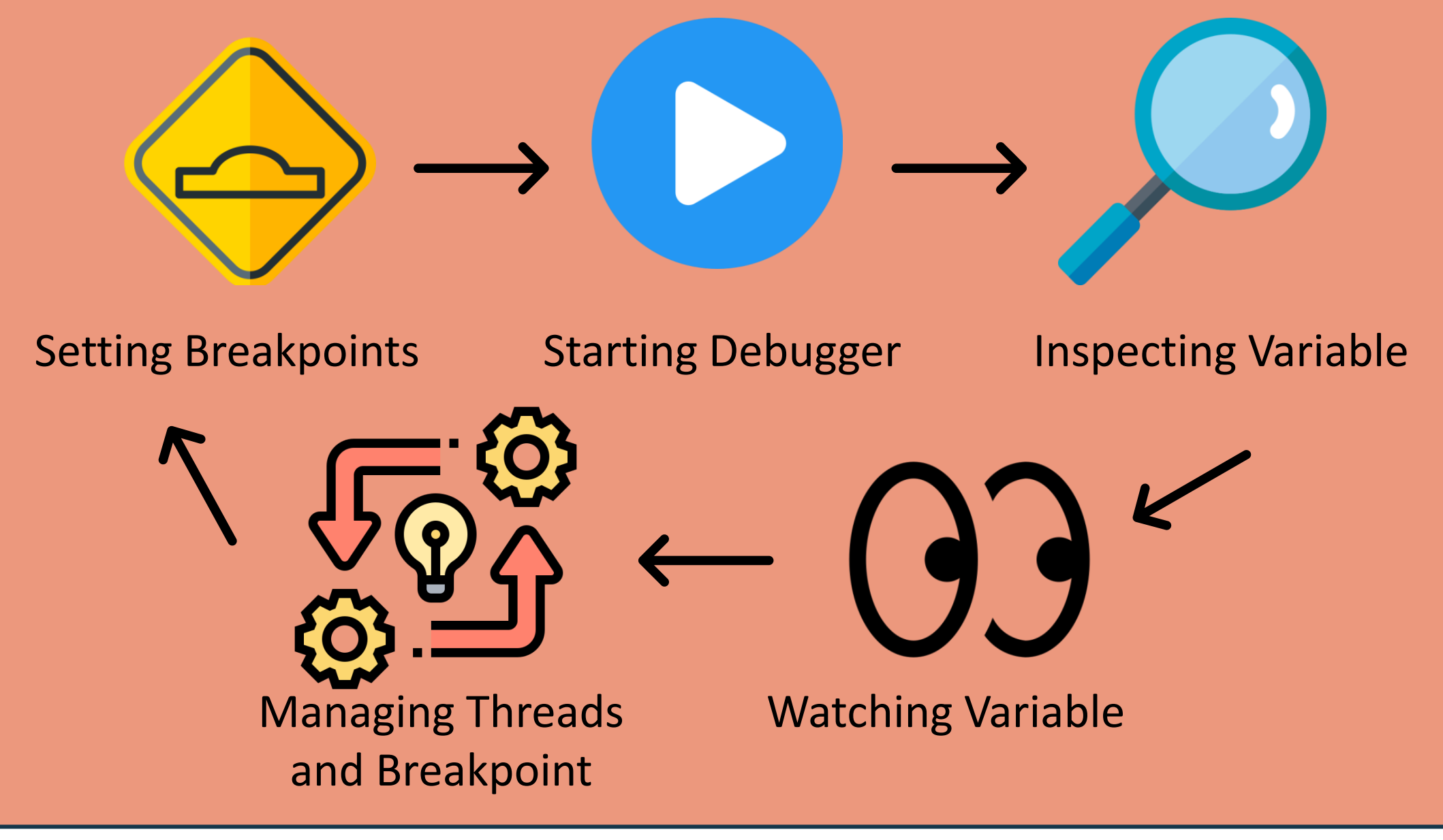
Checkstyle found no problems in the file(s)

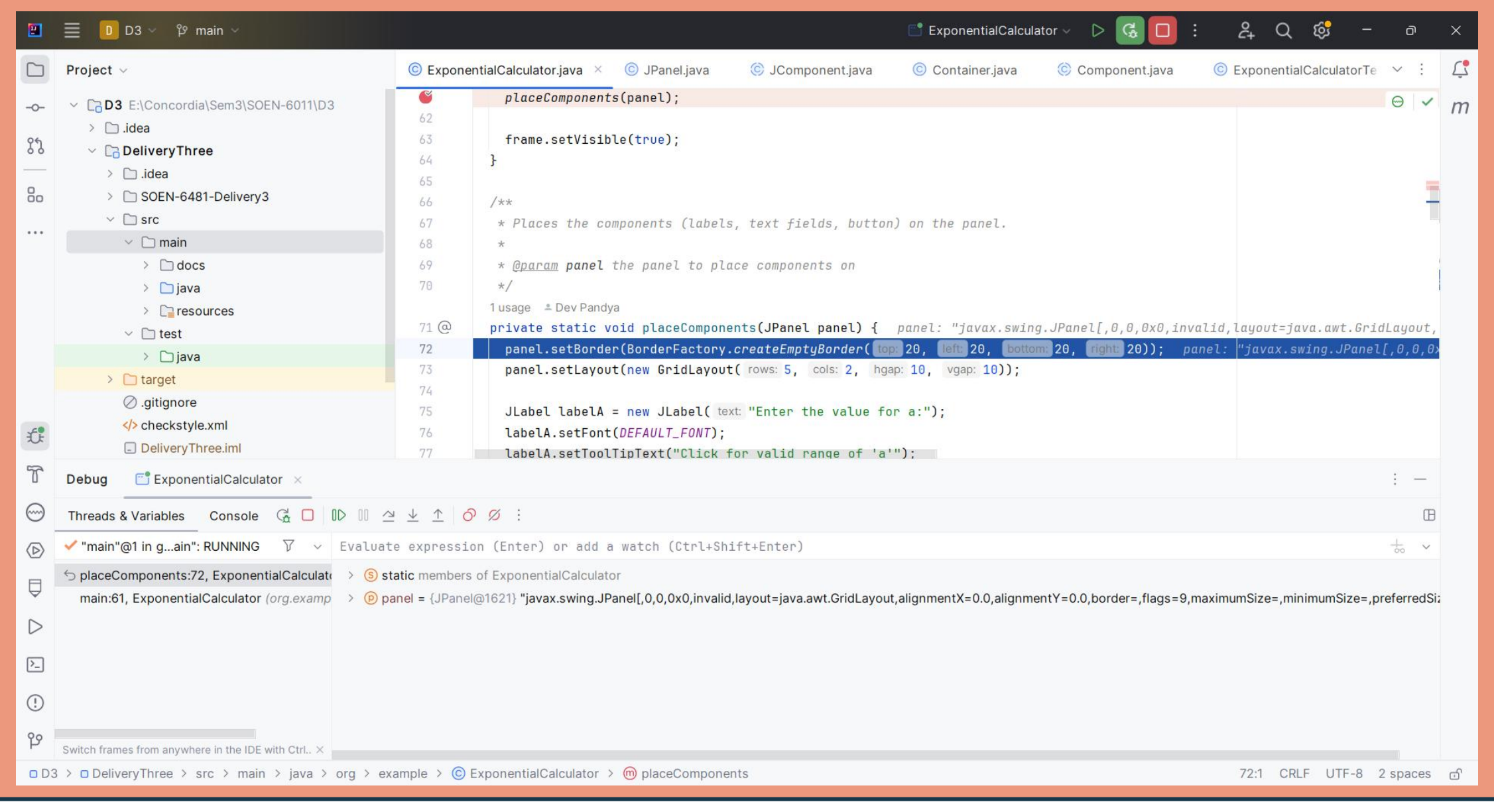
DeliveryThree › src › main › java › org › example ›

The images demonstrate the Checkstyle issues found before and after applying the fixes using the IDE Checkstyle and maven Checkstyle plugin. The adjustments ensure that the code adheres to the specified style guidelines.

Debugging

Debugging is a crucial part of software development that involves identifying and fixing errors or issues in the code. IntelliJ IDEA's integrated debugger provides powerful tools for this purpose. Here's a summary of the debugging process using this tool:

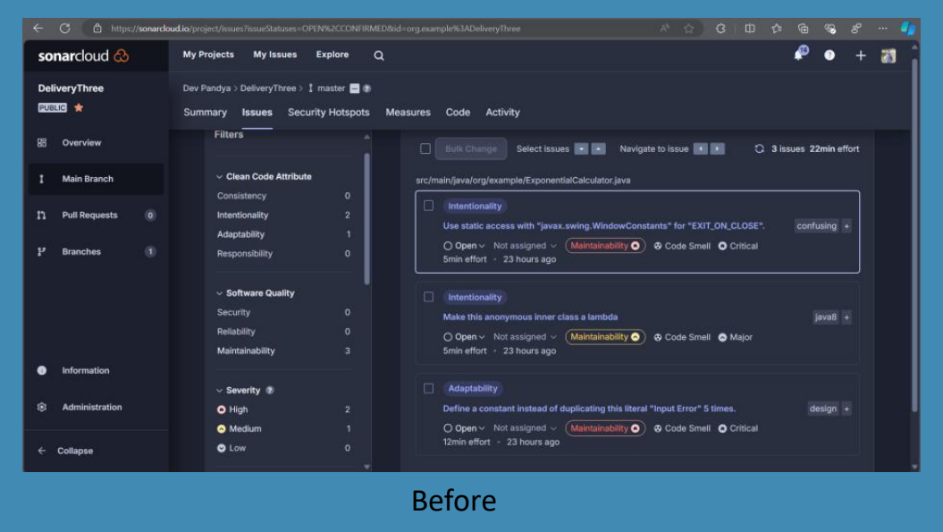


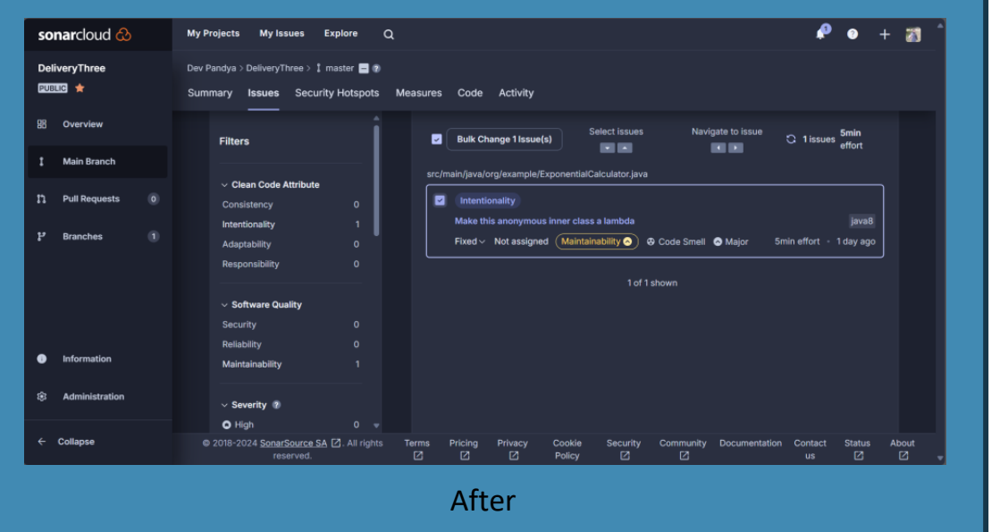


Static Code Analysis

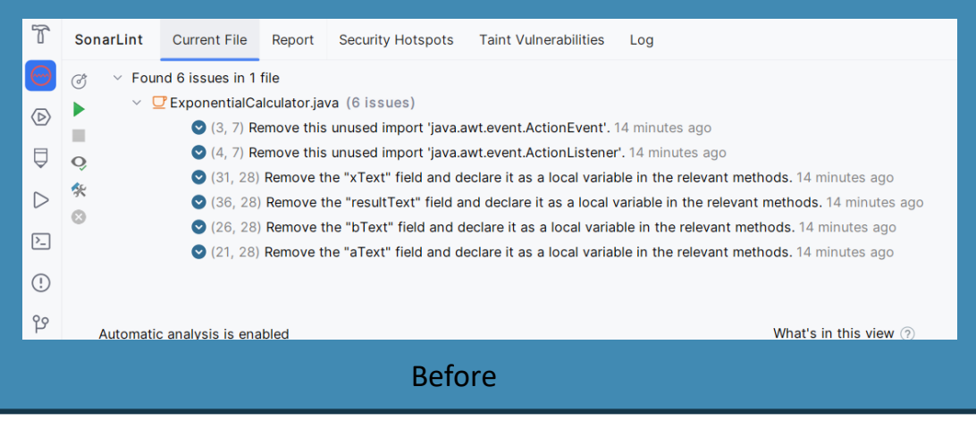
- SonarCloud and SonarLint were employed to ensure high code quality and adherence to best practices. SonarCloud provided comprehensive, server-side analysis of the entire codebase, identifying complex issues, bugs, vulnerabilities, and code smells through its detailed reports
- SonarLint was used for real-time, local analysis directly within the IDE. It provided immediate feedback on code issues as they were introduced, ensuring instant corrections and maintaining high coding standards.
- Snapshots of the before and after results illustrate how the code quality improved over time, showcasing resolved issues and enhanced adherence to best practices.

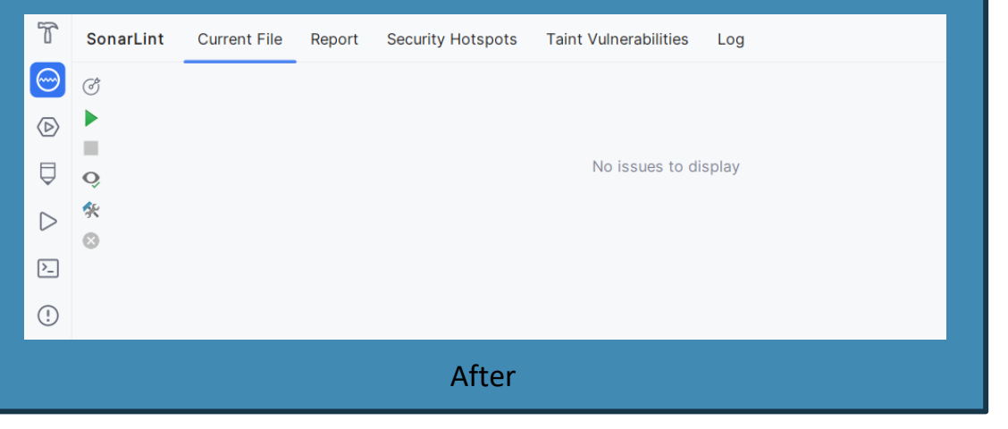
Sonarcloud





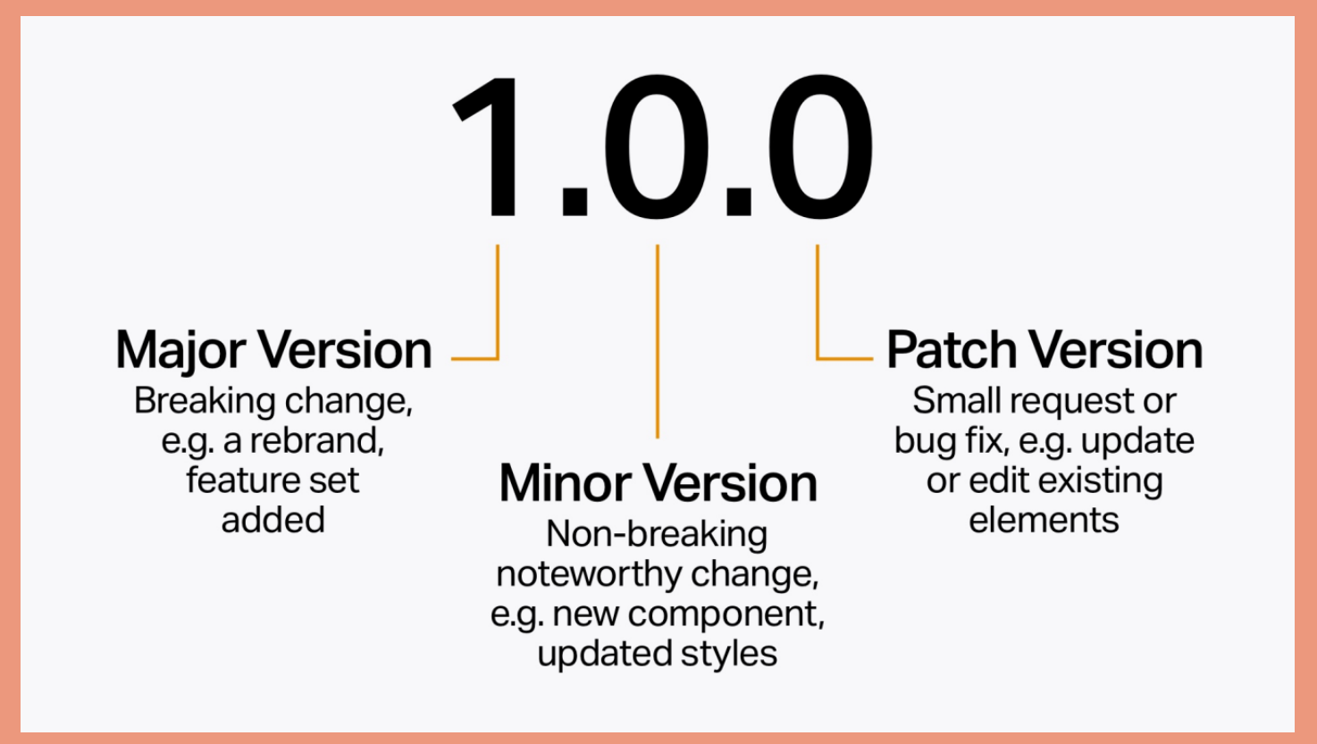
Sonarlint



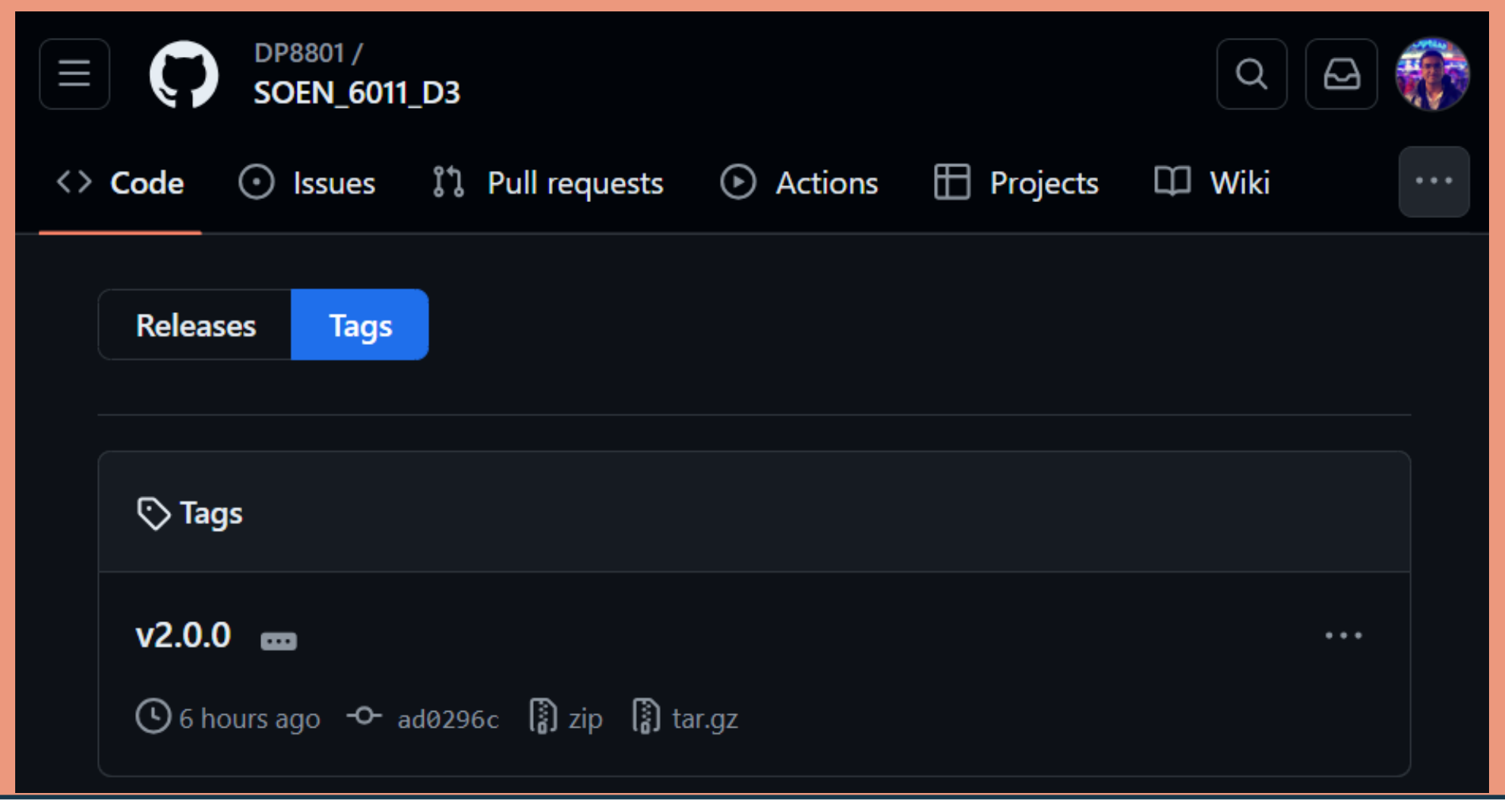


Semantic Versioning

- Semantic Versioning (SemVer) is a versioning scheme for software that aims to convey meaning about the changes in a release through the version number.
- It follows a format of MAJOR.MINOR.PATCH, where:


1.0.0
Major Version
Breaking change, e.g. a rebrand, feature set added
Minor Version
Non-breaking noteworthy change, e.g. new component, updated styles
Patch Version
Small request or bug fix, e.g. update or edit existing elements

- In project, Semantic Versioning was adopted to ensure clear and consistent communication of changes and updates. Initially, the project is tagged as version 2.0.0 on GitHub.



v2.0.0

6 hours ago · ad0296c · zip · tar.gz

Design Principles

User-Centric Design: Intuitive UI with clear labels and error messages.

Separation of Concerns: Code is modular for better readability and maintenance.

Consistency: Uniform fonts, colors, and button styles for a cohesive look.

Error Handling: Robust mechanisms with clear feedback and visual indicators.

Responsive Design: Adapts to various window sizes and resolutions.

Accessibility Features

Color Contrast: High-contrast colors for better readability.

Tooltips and Error Messages: Tooltips for input info and real-time error alerts.

Keyboard Navigation: Supports efficient keyboard interactions.

Real-Time Validation: Immediate feedback via DocumentListener.

Descriptive Error Messages: Clear guidance for necessary corrections.

Testing

- JUnit is a widely-used framework for writing and running unit tests in Java. It provides a structured way to test individual units of code, ensuring that they work as expected.
- Overall, JUnit is instrumental in ensuring that your ExponentialCalculator application functions correctly and reliably by providing a framework for automated and structured testing.
- JUnit tests the core functionalities of the ExponentialCalculator application, including methods for exponential calculations, power functions, logarithms, and exponentials. Each method is verified against expected outcomes to ensure correctness.

Github

- Link to the repository SOEN_6011_D3: https://github.com/DP8801/SOEN_6011_D3

References

[1]: Generative AI. Available at: https://chat.openai.com/

[2]: ISO/IEC/IEEE 29148. Available at: ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering (concordia.ca)

[3]: Transcendental function – Wikipedia

[4]: Exponential Functions

[5]: Series expansion of exponential function and logarithmic function at: https://mathworld.wolfram.com/SeriesExpansion.html

[6]: Semantic Versioning: https://support.invisionapp.com/docs/semantic-versioning