

a2

A2 Guides

These guides support Assignment 2: Road Rage. Read them as you work through the assignment.

Foundational Guides

This page shows only NEW guides introduced in A2. If you need foundational guides covered in previous assignments (Git, Packages, Interface Contracts, Inheritance, JUnit 5, etc.), see the [All Guides](#) page.

Core Concepts

Guide	When to Read
Java Enums	Understanding Direction, Light, and Terrain enums
Polymorphism	Key guide – understanding how the simulation uses polymorphism
Comparable and Comparator	Understanding collision resolution and ordering strategies

Testing

Guide	When to Read
Testing Complex Logic	Testing random behavior, directional preferences, and terrain-seeking logic

Advanced Topics

Guide	When to Read
Creating Custom Maps	(Reference) – Map file format for testing custom configurations
Exploring the Road Rage Codebase	(Optional) – Deep-dive into design patterns in the starter code (Strategy, Factory, Observer, Sealed Types)

References

This section consolidates references from all guides linked above.

Primary Texts:

- Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley.
 - Item 11: Always override hashCode when you override equals
 - Item 14: Consider implementing Comparable
 - Item 18: Favor composition over inheritance
 - Item 19: Design and document for inheritance or else prohibit it
 - Item 20: Prefer interfaces to abstract classes
 - Item 34: Use enums instead of int constants
 - Item 35: Use instance fields instead of ordinals
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
 - Strategy Pattern (pp. 315-323)
 - Factory Method Pattern (pp. 107-116)
 - Observer Pattern (pp. 293-303)
- Freeman, E., & Freeman, E. (2020). *Head First Design Patterns* (2nd ed.). O'Reilly Media.
- Horstmann, C. S. (2022). *Core Java, Volume I: Fundamentals* (12th ed.). Oracle Press.
 - Chapter 5: Inheritance
 - Chapter 14: Concurrency (Comparator examples)
- Reges, S., & Stepp, M. (2020). *Building Java Programs* (5th ed.). Pearson.
- Beck, K. (2002). *Test Driven Development: By Example*. Addison-Wesley.
- Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. [Available free online](#)

Foundational Papers:

- Meyer, B. (1992). Applying "Design by Contract". *IEEE Computer*, 25(10), 40-51.
- Liskov, B., & Wing, J. M. (1994). A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 16(6), 1811-1841.

Language Documentation:

- [Oracle JDK 25 Documentation](#) – Official Java SE reference
- [Oracle Java Tutorials: Inheritance](#) – Official inheritance tutorial
- [Oracle Java Tutorials: Abstract Classes](#) – Abstract class concepts
- [Oracle Java Tutorials: Enum Types](#) – Enum tutorial
- [Oracle JDK 25: java.lang.Enum](#) – Enum class documentation
- [Oracle JDK 25: java.util.Comparator](#) – Comparator interface
- [Oracle JDK 25: java.lang.Comparable](#) – Comparable interface
- [Oracle JDK 25: java.util.Random](#) – Random number generation
- [Oracle JDK 25: java.lang.reflect](#) – Reflection API
- [Oracle JDK 25: Sealed Classes](#) – Sealed classes and interfaces
- [Oracle JDK 25: Pattern Matching](#) – Pattern matching for switch
- [Java Language Specification: Enum Types](#) – Formal enum specification

Testing Frameworks:

- [JUnit 5 User Guide](#) – Official JUnit 5 documentation
- [JUnit 5 Assertions](#) – Assertion methods reference
- [Mockito Documentation](#) – Mocking framework (reference only)

Git and GitHub:

- [GitHub Pull Requests Documentation](#) – Official GitHub PR guide
- [Git Documentation](#) – Official Git reference

Tooling:

- [JetBrains IntelliJ IDEA Documentation](#) – Official IDE reference
- [GitHub Classroom](#) – Educational features for assignments