

Final Keyword

CS 272 Software Development

Final Keyword

- A **final** variable...
 - Cannot be reinitialized
- A **final** method...
 - Cannot be overridden
- A **final** class...
 - Cannot be inherited

Cannot be
REDEFINED

Final Classes

- Protects class from modification
 - Used to preserve immutability
 - Used to for singleton pattern (discussed later)
- Includes many core classes in Java
 - `String`, `Math`, `System`, `Integer`, ...

<https://docs.oracle.com/javase/tutorial/java/landl/final.html>



Final Methods

- Protects an individual method from being overridden
 - Protects `public` and `protected` methods
 - What about `private` methods?
- Used when method behavior **must** remain consistent
 - *e.g.* methods called within a constructor
- Examples include `Object.getClass()`

[https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Object.html#getClass\(\)](https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Object.html#getClass())



Final Variables

- Must be immediately initialized, or initialized in constructor
- For primitive types, results in a constant value
- For objects, only prevents reference from changing
 - Results in a constant if object is immutable
 - May still modify values stored by mutable objects

<https://www.ibm.com/developerworks/java/library/j-jtp1029/index.html> (archived)



Final Guidelines

- Use `final` **liberally** for members
 - Makes assumptions explicit and captures when that assumption is broken
- Use `final` **sparingly** for methods
- Use `final` **rarely** for classes
 - Protects integrity but reduces reusability

<https://www.ibm.com/developerworks/java/library/j-jtp1029/index.html> (archived)



Questions?

