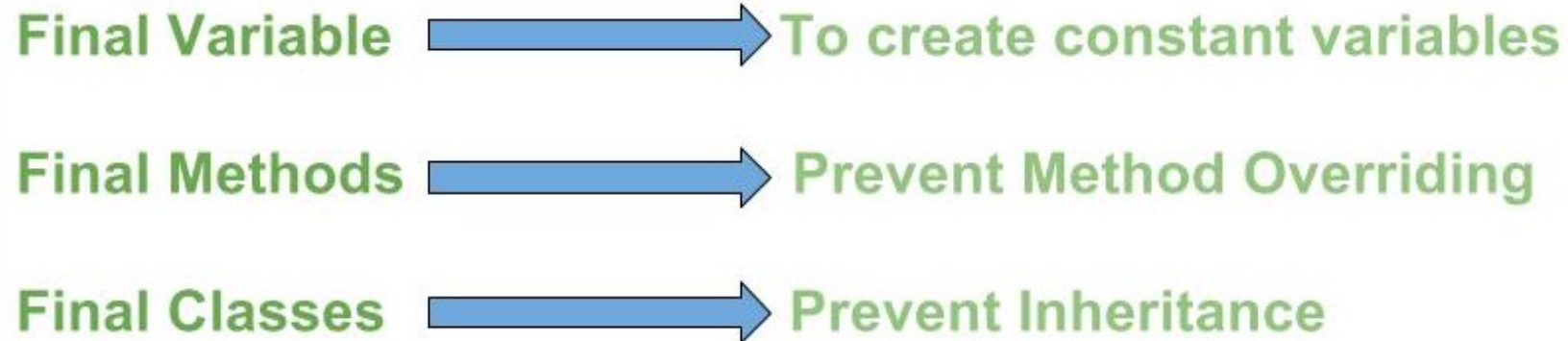


final keyword

- final is a non-access modifier which is applicable to a **variable**, **method**, and **class**.



final variables

- When a variable is declared with final keyword, its value can not be modified.
- When local variable is final, we can declare and assign value later in code
- When instance variable is final, we need to initialize it in:
 - same statement
 - constructor
 - init block

final arrays

- If array is declared as final, elements of array can be changed without any problem
- Arrays are objects and object variables are always references in Java. When we declare an object variable as final, it means that the variable can not be changed to refer to anything else.

final arrayList

- Follows same rules of arrays
- We can remove or update any values
- We can not re-assign it to another object using new keyword

final methods

- When a method is declared with final keyword, it is called final method.
- A final method can not be overridden.

final class

- When a class is declared with *final* keyword, it is called a final class. A final class cannot be extended(inherited).
- You can not make a class immutable without making it final.(String class)

Inheriting Variables

- We can not override instance variables; we can hide them.

Hiding Variables

- Variable hiding happens when we define a variable with the same name as a variable in a parent class.
- This creates 2 copies of the variable within an instance of the child class: one instance defined for the parent reference and another defined for the child reference.
- If you are referencing the variable from within the parent class, the variable defined in the parent class is used.
- If you are referencing the the variable from within the child class, the variable defined in the child class is used.

Hiding Static Methods

- We can not override static methods; we can hide them.
- A hidden method occurs when a child class defines a static method with the same name and signature as a static method defined in a parent class.
- Method hiding is similar but not exactly the same as method overriding.
- The four rules for overriding a method must be followed when a method is hidden. In addition, a new rule is added for hiding a method, namely that the usage of the static keyword must be the same between parent and child classes.

Hiding Static Methods

1. The method in the child class must have the same signature as the method in the parent class
2. The method in the child class must be at least as accessible or more accessible than the method in the parent class
3. The method in the child class may not throw a checked exception that is new or broader than the class of any exception thrown in the parent class method.
4. If the method returns a value, it must be the same or a subclass of the method in the parent class, known as covariant return types.
5. The method defined in the child class must be marked as static if it is marked as static in the parent class (method hiding). Likewise, the method must not be marked as static in the child class if it is not marked as static in the parent class (method overriding)