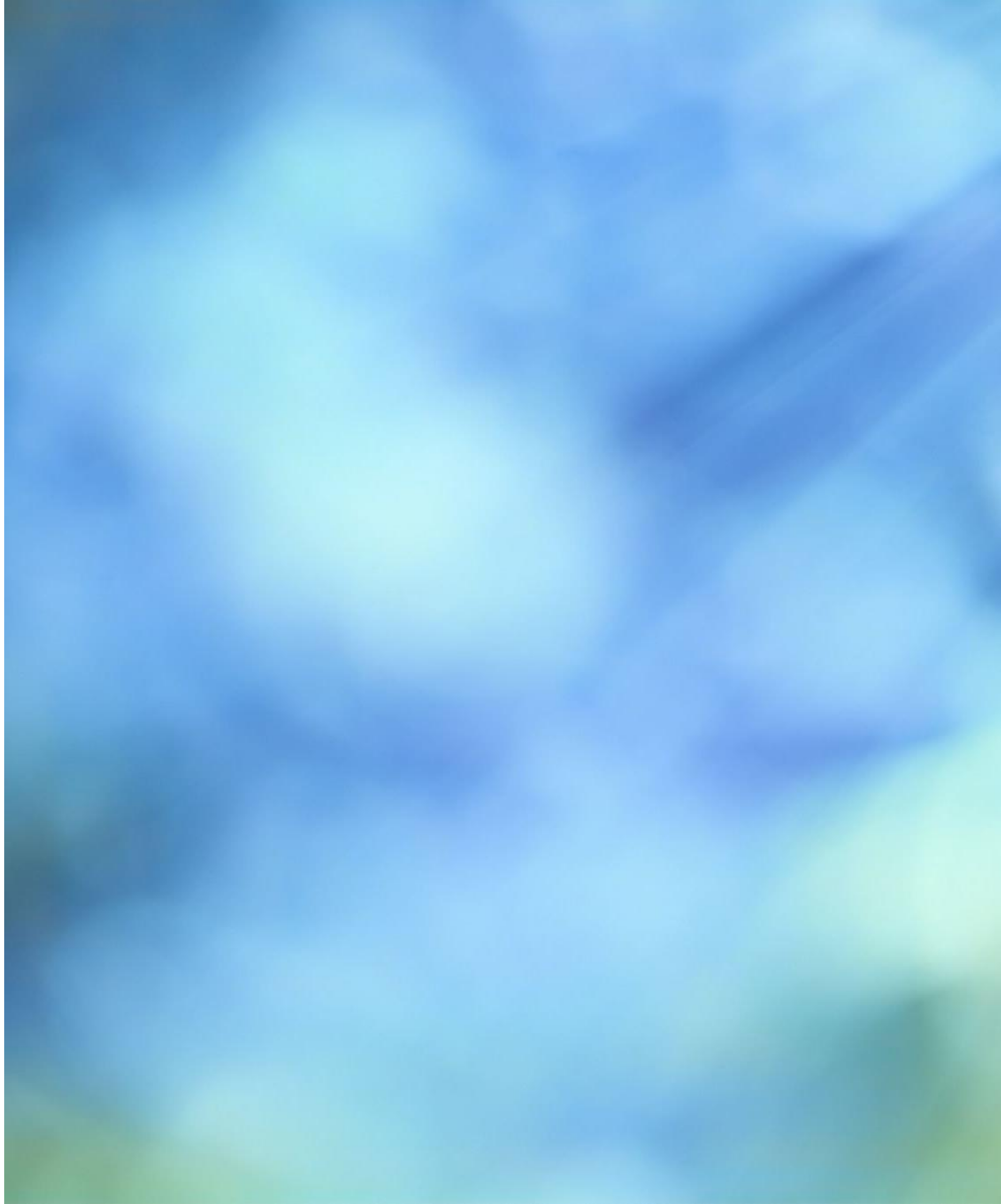
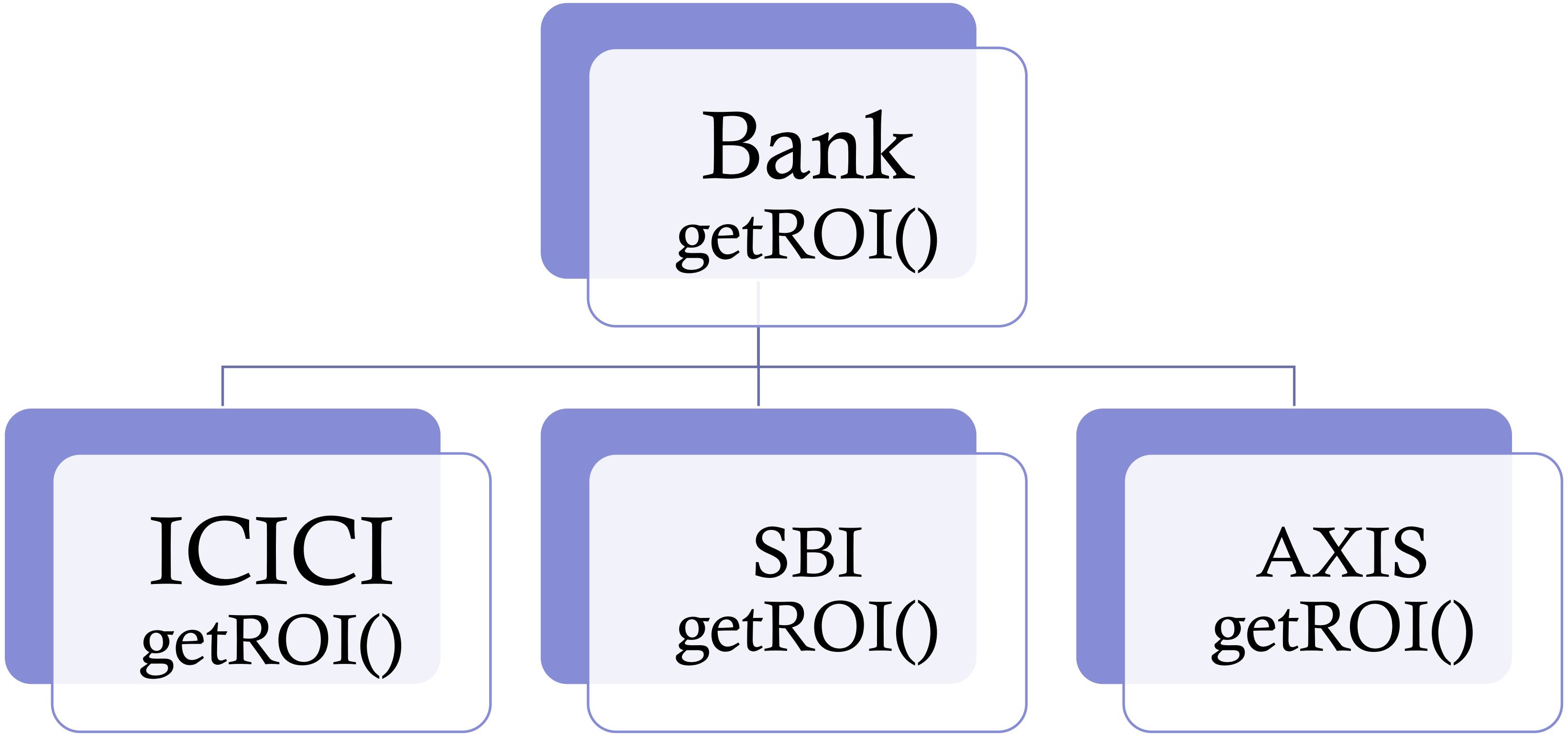
CORE JAVA

UNIT III – Abstract Methods, Final Class, Interfaces



ABSTRACT METHOD



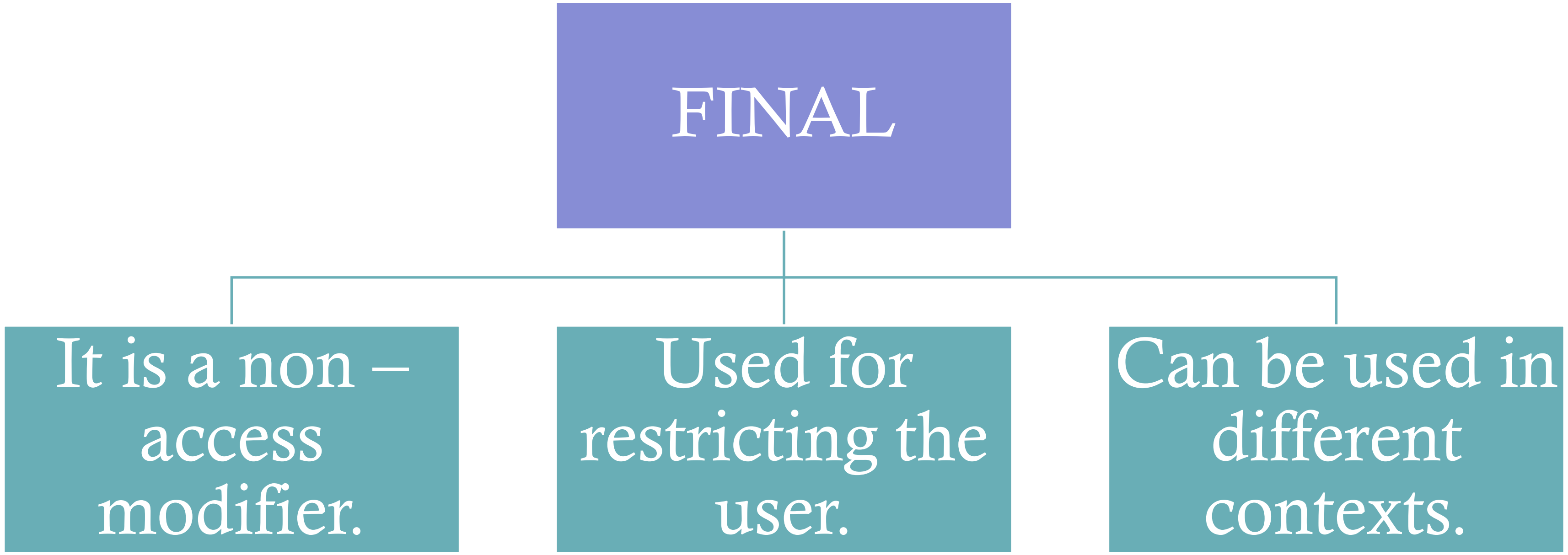
Bank
getROI()

ICICI
getROI()

SBI
getROI()

AXIS
getROI()

FINAL CLASS

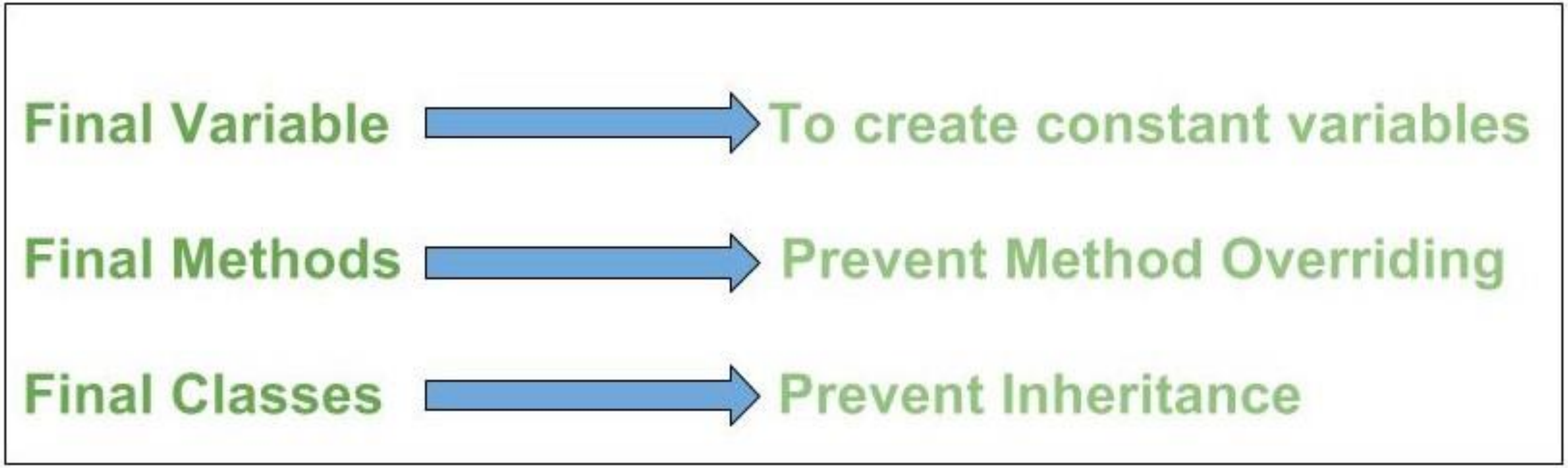


FINAL

It is a non – access modifier.

Used for restricting the user.

Can be used in different contexts.



Final Variable → To create constant variables

Final Methods → Prevent Method Overriding

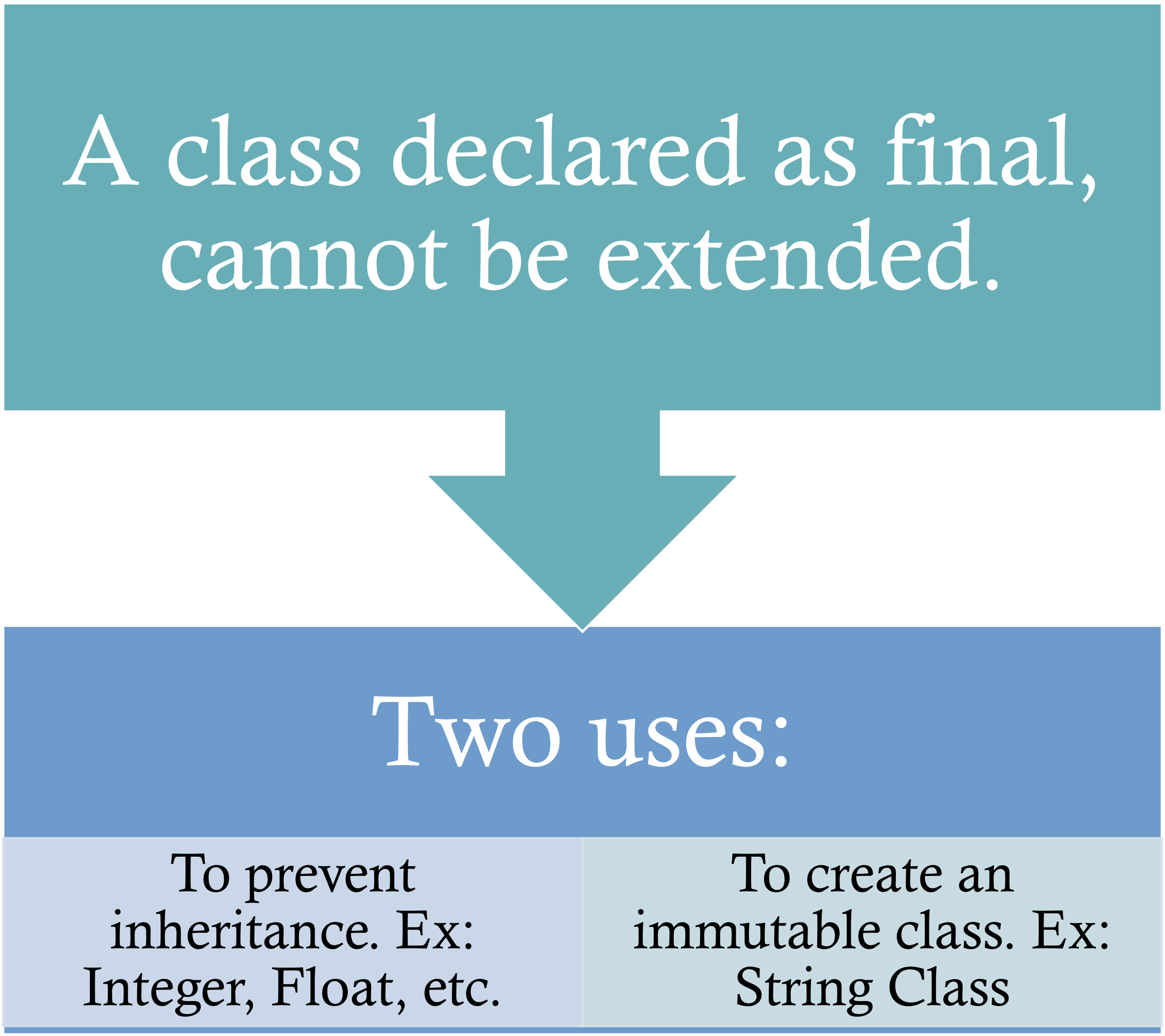
Final Classes → Prevent Inheritance

FINAL METHOD

Methods declared final cannot be overridden.

Main intention for making a method final is to prevent the modification of the method’s content by any user.

FINAL CLASSES



A class declared as final, cannot be extended.

Two uses:

To prevent inheritance. Ex: Integer, Float, etc.

To create an immutable class. Ex: String Class

INTERFACE

Blueprint of the class.

Can have static constants and abstract methods.

To declare an interface, use the "*interface*" keyword.

To use an interface, use the "*implements*" keyword.

WHY USE INTERFACE?

1. Achieve total abstraction.
2. Support multiple inheritance.
3. Achieve loose coupling.

DIFFERENCE BETWEEN ABSTRACT CLASS AND INTERFACE

ABSTRACT CLASS	**INTERFACE**
Can have abstract and non – abstract methods.	Can have only abstract methods.
May contain non – final variables.	Variables are by default final.
Can provide the implementation of interface.	Cannot provide the implementation of abstract class.
Can extend another java class and implement multiple Java interfaces.	Can extend another java interface only.
Can have private and protected class members only.	Class members are public by default.

Thank You!