*class B extends A implements I*

In the above case what happens if class A is abstract , and I is interface , and both have same method to be overridden ?

In Java, when you have a class B that extends an abstract class A and implements an interface I, and both the abstract class A and the interface I have a method with the same signature, the following scenarios can occur:

***Class A Provides an Implementation***: If the abstract class A provides an implementation (concrete method) for the method that's also declared in interface I, then class B is not required to provide an implementation for that method. Class B can inherit the implementation from class A, and it satisfies the contract of the interface I.

abstract class A {

public void myMethod() {

// Implementation in class A

}

}

interface I {

void myMethod();

}

class B extends A implements I {

// No need to provide an implementation for myMethod()

}

***Class A Declares the Method as Abstract***: If the abstract class A declares the method as abstract (without providing an implementation), then class B must provide an implementation for that method, as it's required to implement all the methods declared in the interface I.

abstract class A {

public abstract void myMethod(); // Declared as abstract

}

interface I {

void myMethod();

}

class B extends A implements I {

@Override

public void myMethod() {

// Implementation in class B

}

}