Interfaces

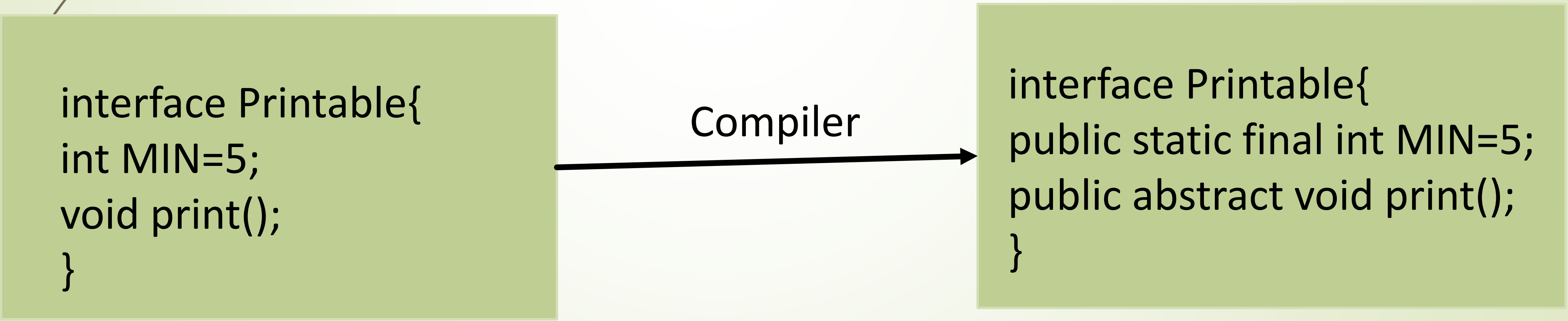
- Interfaces are blue-prints of classes.
- It is a way to achieve abstraction.
- It can contain only abstract methods.

Why use Interface?

- It is used to achieve abstraction.
- To achieve multiple inheritance.

Internal addition by the Compiler

- Interface methods are public and abstract by default.
- Data members are public, static and final.

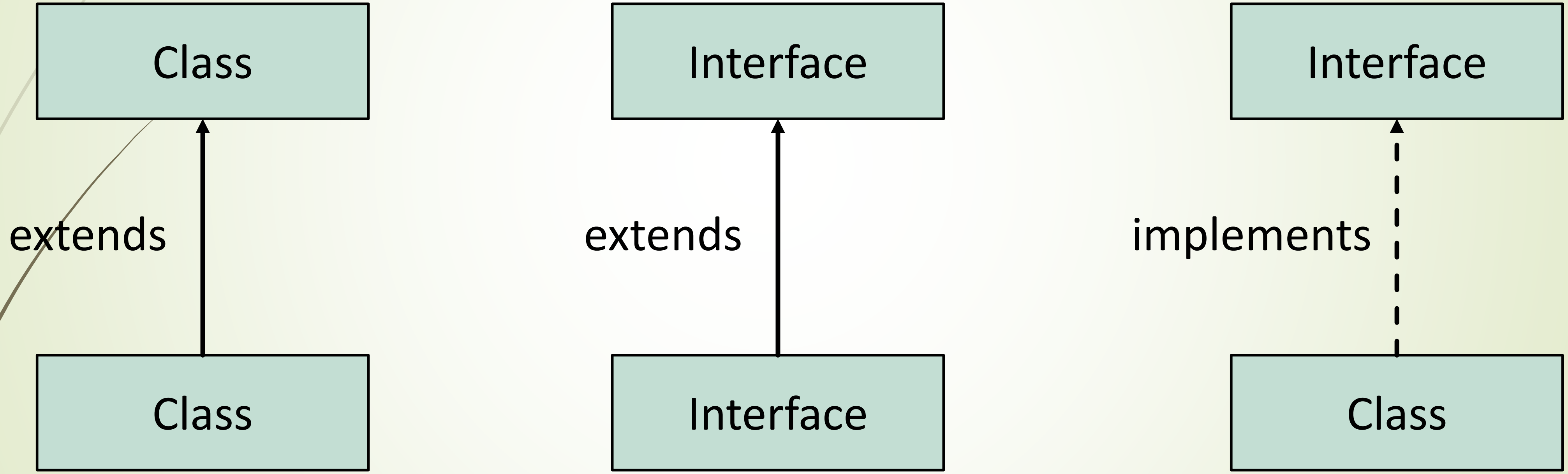


```
interface Printable{
int MIN=5;
void print();
}
```

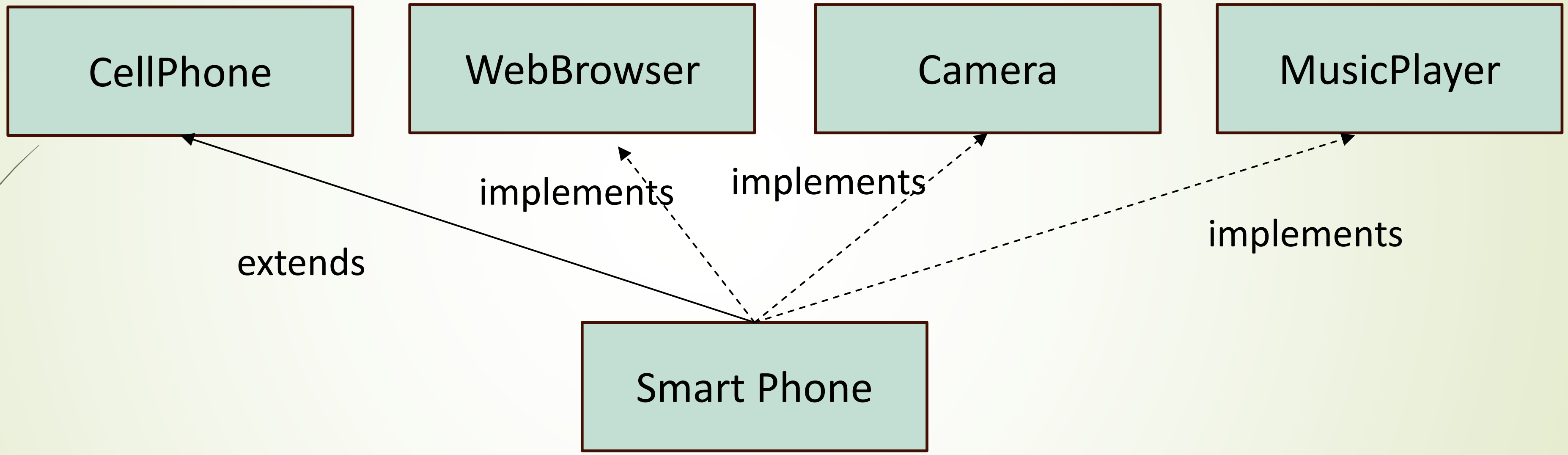
Compiler

```
interface Printable{
public static final int MIN=5;
public abstract void print();
}
```

Relationship Between Classes and Interfaces



Java supports multiple inheritance through Interfaces



CellPhone

WebBrowser

Camera

MusicPlayer

implements

implements

implements

extends

Smart Phone

Class SmartPhone extends CellPhone implements WebBrowser, Camera, MusicPlayer

Default Methods (Java 8 addition)

- Before java 8, interfaces could have only abstract methods.
- If a new method needs to be added in an interfaces, all class implementing it would be affected.
- Default methods remove this problem by providing implementation in the interface.

Private methods

- Interfaces can also contain private methods.
- But, private methods can not be called from outside the class.
- What is the use?
- If length of abstract methos is very long, we can call private methods from within it.