



# Interface & Abstract *Fundamental Programming*

# Definition

- abstract methods = Methods that are declared, with no implementation
- abstract class = A class with abstract methods, not meant to be instantiated
- interface = A named collection of method definitions (without implementations)

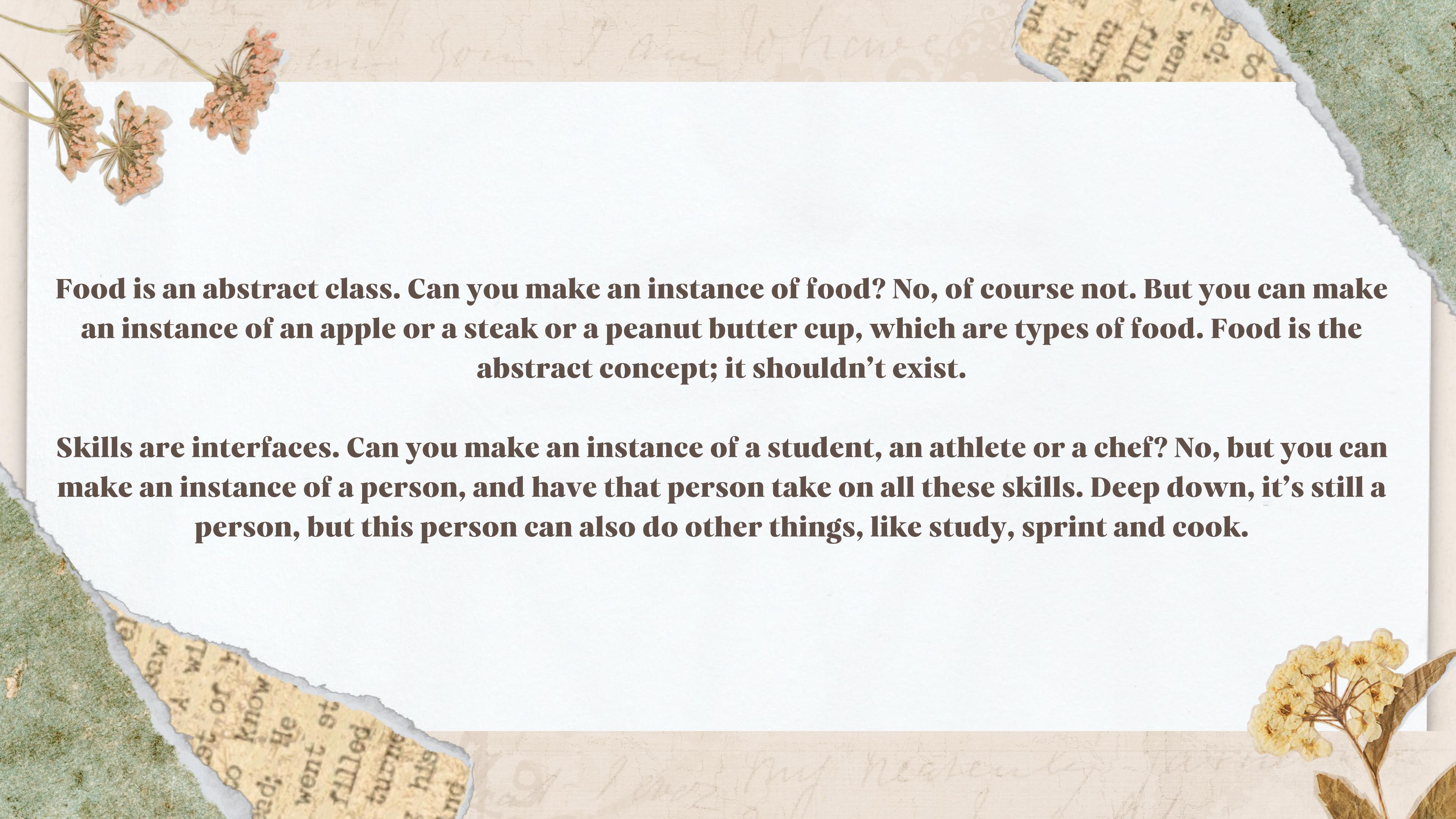
# Syntax

- abstract class:

```
public class Apple extends Food { ... }
```

- interface:

```
public class Person implements Student, Athlete, Chef { ... }
```



**Food is an abstract class. Can you make an instance of food? No, of course not. But you can make an instance of an apple or a steak or a peanut butter cup, which are types of food. Food is the abstract concept; it shouldn't exist.**

**Skills are interfaces. Can you make an instance of a student, an athlete or a chef? No, but you can make an instance of a person, and have that person take on all these skills. Deep down, it's still a person, but this person can also do other things, like study, sprint and cook.**

# Example

```
interface Singer {  
    void sing();  
    void warmUpVoice();  
}  
  
interface Dancer {  
    void dance();  
    void stretchLegs();  
}  
  
interface Talented extends Singer, Dancer {  
    // can sing and dance. Wowwee.  
}
```



Thank  
you