

# Tastes Like Chicken (*An experiment with interfaces in C++*)



---

By Leandro Santiago  
<[github.com/leandrosansilva](https://github.com/leandrosansilva)>



**Chuck Norris Facts**

@chuck\_facts

Follow



If it looks like chicken, tastes like chicken,  
and feels like chicken but Chuck Norris says  
its beef, then it's f [redacted] beef.

11:59 AM - 24 Dec 2009

4 Retweets 3 Likes



[https://twitter.com/chuck\\_facts/status/7009867023](https://twitter.com/chuck_facts/status/7009867023)



# What's a Chicken?




```
struct Look {};  
struct Taste {};  
struct Feeling {};
```

```
struct LooksLikeChicken {  
    virtual Look lookOfChicken() = 0;  
    virtual ~LooksLikeChicken() noexcept = default;  
};
```

```
struct TastesLikeChicken {  
    virtual Taste tasteOfChicken() = 0;  
    virtual ~TastesLikeChicken() noexcept = default;  
};
```

```
struct FeelsLikeChicken {  
    virtual Feeling feelingOfChicken() = 0;  
    virtual ~FeelsLikeChicken() noexcept = default;  
};
```



# Eating Chicken



```
void look(implements<LooksLikeChicken>& l) {  
    const auto look = l.lookOfChicken();  
}  
void taste(implements<TastesLikeChicken>& t) {  
    const auto taste = t.tasteOfChicken();  
}  
void feel(implements<FeelsLikeChicken>& f) {  
    const auto feeling = f.feelingOfChicken();  
}  
void serveChicken(implements<LooksLikeChicken, TastesLikeChicken,  
FeelsLikeChicken>& chicken) {  
    look(chicken);  
    taste(chicken);  
    feel(chicken);  
}
```



According to delishably.com

View 3+ more



Chuck  
steak



Beef  
shank



Brisket



Ribs



Beef plate



Flank  
steak

### The most commonly used types of beef are:

- Chuck.
- Shank.
- Brisket.
- Rib.
- Short plate.
- Flank.
- Loin.
- Sirloin.

More items...

What Are the Different Cuts of Beef? | Delishably

<https://delishably.com/meat-dishes/Cuts-of-Beef>

# F\*\*\*ing Beef



```
struct BeefCut {  
    virtual int cutType() = 0;  
    virtual ~BeefCut() noexcept = default;  
};
```

```
struct Beef { /* Concrete things of a beef */ };
```

```
struct ChuckSteak final: /* is */ Beef,  
/* not actually :-) */ implements<LooksLikeChicken, TastesLikeChicken,  
FeelsLikeChicken, BeefCut> {  
    Look lookOfChicken() final { return {}; }  
    Taste tasteOfChicken() final { return {}; }  
    Feeling feelingOfChicken() final { return {}; }  
    int cutType() final { return {}; }  
};
```

---

# Lunch Time



```
void lunch() {  
    auto steak = ChuckSteak{};  
    serveChicken(steak);  
}
```

```
void  
serveChicken(implements<LooksLikeChicken,  
    TastesLikeChicken, FeelsLikeChicken>&);
```





imgflip.com





# Sometimes with no taste



```
void  
serveChickenWithNoTaste(implements<LooksLikeChi  
cken, FeelsLikeChicken>& c) {  
    look(c);  
    //taste(c); // ERROR, do not have taste  
    feel(c);  
}
```

```
auto steak = ChuckSteak{};  
serveChickenWithNoTaste(steam);
```

---

# Issues and Bugs



- *implements*<A, B> can be used as *implements*<A, B, C>, but cannot as *implements*<A, C, B>, or *implements*<C, A, B>
  - *implements*<C, A, B>  $\neq$  *implements*<A, B, C>;
  - *using AB = implements*<A, B>;
  - *implements*<AB, C>  $\neq$  *implements*<A, B, C>;
  - Currently the max sizeof...(Ts) is 5, as they are hardcoded.
  - Extremely ABI fragile
-



Source Code available at

[https://github.com/leandrosansilva/random-things/tree/master/experiment\\_interface\\_composition](https://github.com/leandrosansilva/random-things/tree/master/experiment_interface_composition)



Questions?

