

Solution Review: Temperature Conversion

This lesson discusses the solution to the challenge given in the previous lesson.

```
package main
import (
    "fmt"
)

// aliasing type
type Celsius float32
type Fahrenheit float32

// Function to convert celsius to fahrenheit
func toFahrenheit(t Celsius) Fahrenheit {
    return Fahrenheit((t*9/5 )+ 32)
}

func main() {
    var tempCelsius Celsius = 100

    tempFahr := toFahrenheit(tempCelsius) // function call
    fmt.Printf("%f °C is equal to %f °F",tempCelsius,tempFahr)
}
```



Celsius to Fahrenheit

We *aliased* the types of the temperature, i.e., `float32` to `Celsius` and `Fahrenheit`. At **line 7**, we alias the type `float 32` by giving the name `Celsius`. We need another type called `Fahrenheit`, so we alias the type `float32` again at **line 8**. Now, we have to write a function that does the conversion from Celsius to Fahrenheit.

Look at the function header at **line 11**: `func toFahrenheit(t Celsius) Fahrenheit`. `toFahrenheit` is the function *identifier*. Parameter `t` with type `Celsius` is passed, and the return type of function is `Fahrenheit`. Inside the function, a one-line implementation is given. The simple formula is applied, and then the result is type-casted to `Fahrenheit` before returning as `Fahrenheit((t*9/5)+32)`.

In the `main` function, we declare a variable `tempCelsius` of type `Celsius` at **line 16**. Next at **line 18**, we call the function `toFahrenheit`, and send `tempCelsius` as the parameter to it and store the result in `tempFahr` a variable of `Fahrenheit` type. Finally, at **line 19**, we print the temperature to view the result.

That's it about the solution. In the next lesson, you'll study strings, a non-elementary yet an important datatype of Go.