

# Exercise on Composition

Here is an exercise you can solve to test your understanding!

## WE'LL COVER THE FOLLOWING ^

- Question

## Question #

Looking at the `User` / `Player` example, you might have noticed that we composed `Player` using `User`. This means that a `Player` should be able to access methods defined in the `User` struct. In the code given below, add additional code to the `GreetingsForPlayer` function so that it uses the `Greetings` function from the `User` struct to print the string that the `Greetings` function is printing right now:

### Environment Variables



Key:	Value:
GOPATH	/go

```
package main
import "fmt"
import "encoding/json"

type User struct {
    Id      int
    Name, Location string
}

func (u User) Greetings() string {
    return fmt.Sprintf("Hi %s from %s",
        u.Name, u.Location)
}

type Player struct {
    u User
    GameId int
}
```



```
func GreetingsForPlayer(p Player) string{  
    //insert code  
  
    return ""; //modify the statement to return the required string  
}
```

