

```
func main() {  
    fmt.Println(getNumber())  
}  
  
func getNumber() int {  
    var i int  
    go func() {  
        i = 5  
    }()  
  
    return i  
}
```

Here, there is a race condition when `getNumber()` is called. As the anonymous function within it runs on a new routine, in the main routine, the function `getNumber` may return `i` with the value of 0 unless `func()` finishes faster!