race.go

This program has two goroutines, one which adds to a variable and one that subtracts from a variable. Both print the value after this operation and both repeat the operation in a loop 10 times.

Because the two goroutines share a common pointer to a variable they are both simultaneously changing the value. This results in a race condition as the numbers printed will likely be different every time as the go runtime schedules the tasks differently everytime.

For example all of additions could happen first resulting in the program printing 0 to 9 and then 9 to 0. Or the subtractions could all happen first resulting in 0 to -9 and then -9 to 0 printing. Or the order could be varied with some additions happening then some subtractions then the remainder of the additions and subtractions. This means the sequence it prints can’t be predicted due to the race conditions of both functions modifying the same variable.