



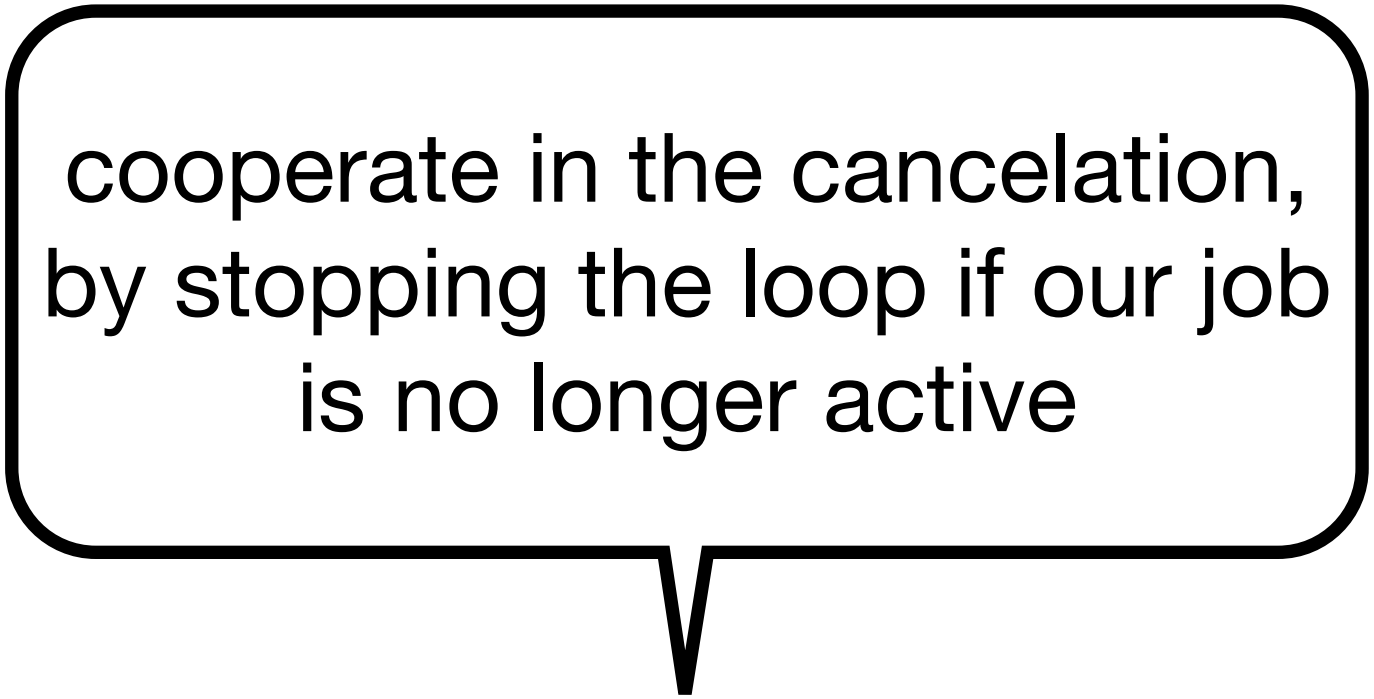
**Cooperative Cancellation**

3

5

```
fun CoroutineScope.cashDesk() = launch(IO) {  
    while (isActive) {  
        println("How many pretzels?")  
        val count = readLine()?.toIntOrNull()  
        //receive pretzel  
    }  
}
```

```
fun main() = runBlocking {  
    val job = cashDesk()  
    delay(1000)  
    job.cancelAndJoin()  
}
```



cooperate in the cancelation,  
by stopping the loop if our job  
is no longer active



by using a multi-threaded  
dispatcher yield isn't needed

# Cooperative Cancellation

cooperate in the cancellation,  
by stopping the loop if our job  
is no longer active

by using a multi-threaded  
dispatcher yield isn't needed

```
fun main() = runBlocking {  
    val job = cashDesk()  
    delay(1000)  
    job.cancelAndJoin()  
}
```

```
fun CoroutineScope.cashDesk() = launch(IO) {  
    while (isActive) {  
        println("How many pretzels?")  
        val count = readLine()?.toIntOrNull()  
        //receive pretzel  
    }  
}
```



# Calling Blocking Code