Module 3: Threads in Go
Topic 2.1: Basic Synchronization

Synchronization

- Using **global events** whose execution is viewed by all threads, simultaneously

1: `x = 1`	
	1: `print x`
2: `x = x + 1`	

1: `x = 1`	
2: `x = x + 1`	
	1: `print x`

- Want print to occur after update of x

Synchronization Example

Task 1   **Task 2**

```
x = 1
x = x + 1
GLOBAL EVENT
```

```
if GLOBAL EVENT
      print x
```

- **GLOBAL EVENT** is viewed by all tasks at the same time

- Print must occur after update of x

- Synchronization is used to restrict bad interleavings