Module 3: Threads in Go
Topic 3.1: Communication

Goroutine Communication

- Goroutines usually work together to perform a bigger task
- Often need to send data to collaborate
- Example: Find the product of 4 integers
  - Make 2 goroutines, each multiplies a pair
  - Main goroutine multiplies the 2 results
- Need to send ints from main routine to the two sub-routines
- Need to send results from sub-routines back to main routine

Channels

- Transfer data between goroutines
- Channels are typed
- Use `make()` to create a channel

```
c := make(chan int)
```

- Send and receive data using the **<-** operator
- Send data on a channel

```
c <- 3
```

- Receive data from a channel

```
x := <- c
```

Channel Example

```
func prod(v1 int, v2 int, c chan int) {
   c <- v1 * v2}
func main() {
   c := make(chan int)
   go prod(1, 2, c)
   go prod(3, 4, c)
   a := <- c
   b := <- c
   fmt.Println(a*b)
}
```

UCI Division of Continuing Education