Topic 10
Subroutines
Ch 9 in Scott
   return value — function
   no return — procedure
   associated with structure/class — method (regardless of return)

Skip 9.1 & 9.2 Mostly.
Quickly, What happens when you make a subroutine call

   1. Compute (as needed) value of arguments to called subroutine
   2. if a nested subroutine — the “static link” ie what variables are available to nested sub
      from its surrounding subroutine

      how is static link distinct from closure?  Closures are about functions that are
      defined, but then executed later.  Static link is about functions executed now.  So they refer to
      largely the same thing; but with different uses.  Also closures can get outside the function in
      which they were defined.
      the callee — on starting:
         3. Pass return location of the calling function
         4. Add all of this to the “call stack”

Parameter passing:
   Two basics
     pass by value
     pass by reference
   These generally follow the from value-model and reference-model languages.
   Exceptions: Java — primitive types
    Go — pointers
       Can change the thing pointed to but if try to
   change the passed pointer, that change does not survive!
      see passpoint_go/
   So go is essentially passing the pointer by value!
   Kotlin: does not allow change the ref of ANY passed var.  All params are “val”.
This makes kotlin (and Java) “call by sharing”
   see passref_kt
   see alos passobj_kt
   Note that is kotlin, when you cannot change the location of the pointer, you can
   change what is inside
      What have pass by reference
      effectively allows function to return multiple values (dangerous and NOT
      functional)
      PbV on large structs can be expensive

Closures — again
when functions are passed in kotlin (and Go) they are passed with their closure (if
closure exists) (when will it not?)
   For instance see clous1_go/1
   Question what happens when change values in closure AFTER function defintion
      drr clous1_go/2

Java — functions are NOT first class so cannot be passed so closures are not a thing
   Instead “Object closures”.  Idea, define an object with method and vars.  Then
   pass the objects.  The receiver knows to use the method defined within the object.  This is
   used a lot in Android to get “callbacks” and more generally in multithreaded programs.
Illustrative example of callback?
For instance, downloading stuff from web. By rule in Android, this must be done in a separate thread. When download complete, want to notify the “main thread” and have it put up a “complete/fail” message. So, pass in an instance of something like

```java
public class Downer {
    String url = "https://getme.com/123.jpg"
    CallbackClass cc = xxx;
    public void onDownLoadDone(String res) {
        xxx.downloadDone(res);
    }
}
```

Optional parameters
“Go has no concept of default parameters nor any way to specify arguments by name”
Why not?
Go does not allow overloading of function names
Why Not?
Kotlin allows both named parameters and overloading.
Do you really need both?
Java overloading only
Variable number of args
Go — YES
the print statement
see also varadic_go
clever user of functions — but is this clearer??
Kotlin/Java YES
kotlin see varidic_kt
syntax-wise I think this is pretty clear — certainly more so than

Java

Function returns
1. Should functions allow return statement other than at the end?
Go: multiple return values; see return_go

Go: named return values; see return_go
are these good?
Kotlin does not have multiple returns. Why not? Are they needed?

Exceptions:
Kotlin:
try-catch like Java except you are never required to have try/catch!
Why?
https://radio-weblogs.com/0122027/stories/2003/04/01/
JavasCheckedExceptionsWereAMistake.html
Handling in Go
basic approach, lots of functions return two values
desired value, error
see basicexcep_go/
when normal, err value is 0 (nil).
when problem, normal value is 0 (possibly) and error value is non-0
So, how does func set that non-nil error?
1. error prevention — do not just open a file and try to read, then get exception when file does not exist. First check that file exists. Yes, programmer has to do this.

2. PANIC. panics are the equivalent of a thrown exception in Java/Kotlin if not “caught”, program dies. BUT catch without try catch???

Note: “giving up is usually the right response to a panic, but not always”

More generally, problems. Answer, catch with a function that is guaranteed to run at the end of a function kind of like try {} finally{} in Java

in Go “defer” == run just before function completes.

So defer is part of the language inside deferred function call “recover()”. If execute while NOT in panic, return null. if return NOT null, then panicing!!!

So, how do you get a func to return an error value is the only way to catch the error value is done by a function that must run as the last thing in a function???

A By having the actual work done in an inner function!! see except1/go/betterPanic()

OR

use go named return values and set the value of the named returns inside the deferred fun see except1/betterPanic2()

This use case may be logic behind named returns

Text — exception Handling

3 methods of dealing with exceptions without explicit exception handling mechanism

1. Return one value with special values indicating that things went wrong. For instance, if function is supposed to return a positive integer, return -1 to indicate an error

2. Return a status in addition to value (in a multiple return language). The is he general approach taken by Go, with the Panic as a bonus

3. Caller returns a closure containing an error handling function.

From booK: 1 is difficult to generalize, 2 and 3 can introduce new errors and obfuscatory..

So what does Go take option 2?

it does not.

Programmer should do stuff to prevent problems. If unable to prevent, only then, maybe, panic and die.

Problems with catching exceptions someplace other than the func in which they occurred — need to unwind the stack!!!!

3 goals of exception handlers:

A. compensate for exception to allow program to recover
B. Clean up and re-throw so someone else can handle
C. Print an error message.

Events

also “event-driven programming”

Point — the program might be doing other things, it is NOT just waiting on the event events are “something to which a running program needs to responds, but which occurs outside the program, at an unpredictable time.”

GUI input
Network events (read URL)

Synchronous handlers

program is single threaded, event interrupts whatever the program had been doing; does its work, then is gone. Can be handled kind of like any subroutine call.
Threaded Handlers
most modern programming languages
Problem, how to handle data structure use clashes between threads.
Java synchronized. ArrayList vs Vector. All the methods of Vector is synchronized. But, the methods of ArrayList is not synchronized.
See speed_kt/speed.kt
this is a small, but detectable slowdown for vector.

MAYbe MAKE LAST TOPIC OF SEMESTER PARALLEL PROGRAMMING IN GO?