**M1.2.2**

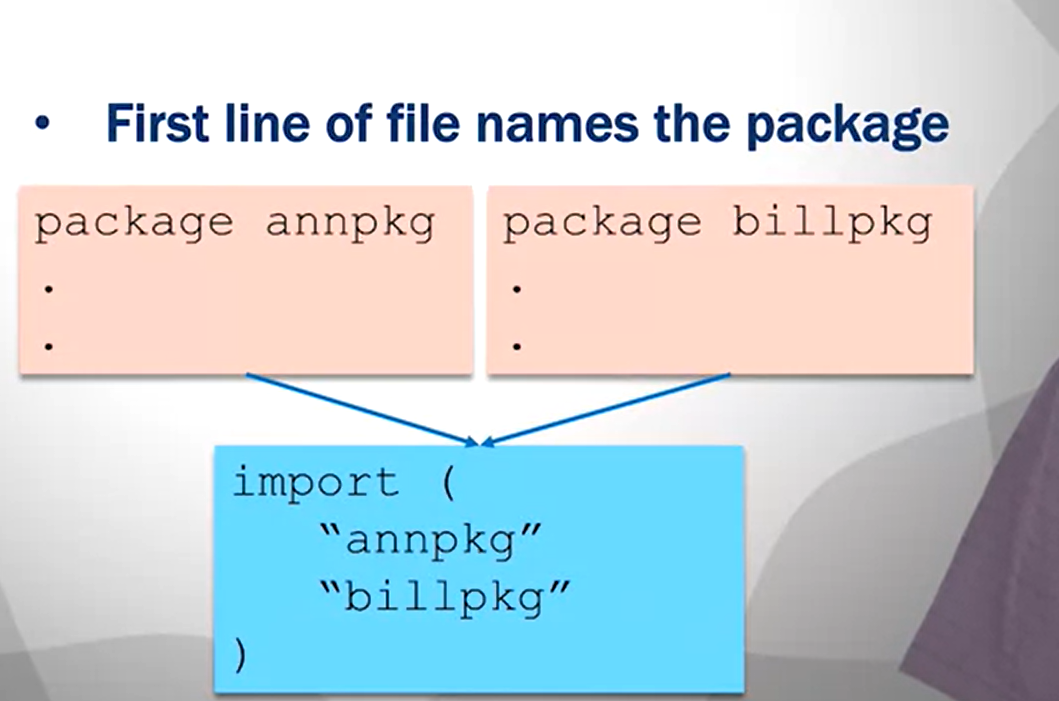
Go Workspace suggestions/recommendations

Three Subdirectories within your workspace:

1. Src – contains source code files
2. Pkg – contains packages (libraries)
3. Bin – contains executables

User will typically have one workspace for many projects

* Workspace directory defined by GOPATH environment variable
* GOPATH is defined during installation
* Go tools assume that your code is in GOPATH



Blue box is some src file example

\*\*\* There must always be one package called main 🡪 where code executation starts

* Building the main package generates an executable program
* Main package needs a main() function
* Main() in where code execution starts of course



#### M1.2.3 - Go Tool

* Import keyword is used to access other packages
* Searches directories specified by GOROOT and GOPATH
* Standard library includes many packages (including fmt)

**go build:**

* Compiles the program
* Arguments can be a list of packages or a list .go files
* Creates executable for the main package
* .exe suffix for executable in windows

**go doc:**

* Prints documentation for a package

**go fmt:**

* Formats source code files

**go get:**

* Downloads packages and installs them

**go list:**

* Lists all installed packages

**go run:**

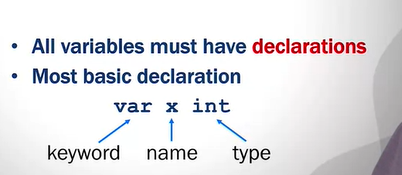
* Compiles .go files and runs the executable

**go test:**

* Runs test files in \_test.go

#### M1.3.1 - Variables

Names must start with a letter, and they cannot contain keywords



Can declare multiple on the same line:

**var x, y int**

#### M1.3.2 - Variable Initialization

You can define aliases for types:

Type Celsius float64

* Whenever you set variable to type Celsius, you’re setting to type float 64

You can initialize in the declaration

Var x = 100 🡪 you’ll get an inferred type of ‘int’

**Uninitialized variables have a zero value**

* **Int = 0**
* **String = ‘’**

Can perform a declaration and initialization together with the := operator:

* X := 100
* Variable is declared as type of expression on the right hand side
* Can only do this inside a function