QUIZ

When would building an executable file using `go build` be preferable to `go run`?

When we want to run the file WITHOUT creating an executable file, then `go build` is preferred.

If the code is being used in production (users are interacting with it) then using an executable file is preferred.

👏 Correct, rather using `go run` multiple times, using `go build` once and executing the binary file is better for performance.

When we want to run a program built with another language we use `go build`.

When a program contains multiple compilation errors, its easier to debug using `go build`.

Fill in the code to only print out `Just Right.`.

```
/*
fmt.Println("Too hot!")
fmt.Println("Too cold!")
*/
fmt.Println("Just right.")
//  fmt.Println("I'll sleep here")
```

👏 You got it!

What is the purpose of a library in Go?

A library is a Go package that won't create an executable file but contains code that we can use for our program.

👏 Right! We can import a package (library) that contains code that we need for our program.

A library is a keyword that is used to import packages.

A library determines whether a Go program is executable or not.

A library contains multiple executable files grouped together to run concurrently.

Based on the code provided, what would be printed to console?

```
package main

import "fmt"

func main () {
  fmt.Println("Hello")
  // fmt.Println(",")
  /*
  fmt.Println("World")

  fmt.Println("!")
  */
}
```

`World !`

`Hello World`

`Hello`

👏 Correct, all the other print statements are commented out!

What is the importance of the package declaration: `package main`?

A package with a `package main` declaration will be a library file.

A package with the `package main` declaration will compile into an executable file.

👏 Correct!

The `package main` declaration ensures that a program compiles regardless of any errors.

Including `package main` will speed up a Go program by 20%.

True or False:

The command `go run` (followed by a Go program) will build an executable file that we can keep and also run the program.

- True
- False

The `go run` command will run the program, but it will not build an executable file that we can keep.

What is the purpose of Go's compiler?

- Go's compiler pools together all of our Go program's code into one pile for the computer.
- Go's compiler imports the neccessary packages for code to run.
- Go's compiler makes it easier for humans to read the code by adding in syntax highlighting.
- The purpose of Go's compiler is to translate Go code into binary code that computers understand.

Correct! Go code does not directly make sense to computers, however, binary does.

What command would reveal more information ONLY about the `Scan` function in the `fmt` package?

- `go doc`
- `go doc fmt.Scan`

Correct, the `go doc` command followed by the package and the function name will print out information about the function's usage.

- `go doc fmt`

What commands are needed to build an executable file and then run the executable file?

```
go build main.go
./ main
```

You got it!

Fill in the code to import both the `fmt` and `math` packages and alias `fmt` as `f`.

```
import (
    f "fmt"
    "math"
)
```

You got it!