✓ **Congratulations! You passed!**

TO PASS 75% or higher

Keep Learning

GRADE
100%

Module 3 Quiz

LATEST SUBMISSION GRADE

100%

1. What is the difference between an **object** and a **class**?   `1 / 1 point`

   ○ An object is a field of data inside a class.

   ◉ A class is a template and an object is an instance of that template.

   ○ An object is a particular kind of class.

   ○ An object typically contains more data fields than a class.

   > ✓ **Correct**
   > Correct!

2. What is the difference between a struct in Go and a class in an object-oriented   `1 / 1 point`
   language?

   ◉ A struct contains only data while a class can also contain methods.

   ○ A class describes data fields but a struct does not.

   ○ A struct can only be created inside a class.

   ○ A struct cannot contain another struct.

3. Which of the following refers to data hiding?   **1 / 1 point**

   ○ Instantiation

   ○ Polymorphism

   ○ Inheritance

   ● Encapsulation

4. How do you associate a method with an arbitrary data type on Go?   **1 / 1 point**

   ● Define the method so that its receiver type is the data type of interest.

   ○ Define the method inside the data type definition.

   ○ Include the name of the data type in the name of the method.

   ○ Define the data type and the method in the same file.

5. In Go, how do you hide variables or functions in a package, so that functions outside   **1 / 1 point**
   of the package cannot access them?

   ○ Use the **package** keyword

   ○ Use the **private** keyword.

(●) Give the variable/function a name which starts with a lower-case letter

(○) Define the variable/function inside the package.

✓ **Correct**

Correct!

6. Say that you have defined a type **t** and you have declared an object of that type called **t1**. Assume that the type **t** is the receiver type for a method called Foo(). Which expression shows a proper invocation of the the method Foo()?

1 / 1 point

(○) Foo(t1)

(○) Foo(t)

(●) t1.Foo()

(○) t.Foo(t1)

✓ **Correct**

Correct!

7. Assume that that the type **t** is the receiver type for a method called **Foo()**. Under what conditions would it be better to make the receiver type of **Foo()** a pointer to **t**, rather than itself?

1 / 1 point

I. When the receiver type t uses a large amount of memory.

II. When the method Foo() must modify the data in the object of the receiver type.

(○) Only I

(○) Only II

(●) Both I and II

(○) Neither I nor II

Correct
Correct!