Answer the discussion questions in writing.

1. Declare an enum for military ranks, either officer or enlisted. Name it Ranks. What are the symbols, like Private, PFD, Corporal, or 1stLt, 2ndLt, Capt?

enum Ranks { 1LT, 2LT, CPT } // use short to save memory, max 256 literals

1. Using the Ranks enum, assign a rank to yourself or a friend.

Ranks Pearce = Ranks.CPT;

1. Determine the numeric index of particular ranks, using the Ranks enum.

Console.WriteLine((int)Pearce); // writes out ‘2’

1. How do you select the type of an enum?

Use a colon after the enum name and then specify the type. For example:  
enum Ranks : byte { 1LT, 2LT, CPT } // declares the enum as type byte

1. Are structs stored on the stack or on the heap? What about enums?

Enums and structs are both stored on the stack.

1. Declare a struct named DOD with four branches.

struct DOD  
{  
 public string “Army”, “Air Force”, “Navy”, “Coast Guard”;  
}

1. Why can’t you create a default constructor for a struct?

In a struct, the compiler always generates a default constructor for you, so you will get a compile-time error if two default constructors (yours and the compiler’s) try to run at the same time. In a class, for comparison, the compiler only generates a default constructor if you don’t write a constructor yourself.

1. What is CIL? What does the CLR do to the CIL?

CIL is the Common Intermediate Language. It is pseudo-machine code that the compiler converts C# code into when compiled. The instructions in the CIL are stored in an assembly. When the C# app is run, the Common Language Runtime (CLR) converts the CIL instructions into real machine instructions that the processor can execute.