HW2 Solution – Penn State CMPSC 461 Spring 2022
Homework 2 Solution. G. Tan.

1.
    a. **Main - <**a,15>, <b,15>, <j,1>, <i,1>, <h,1>;
       A - <x,8>, <y,8>, <i,9>, <j,1>, <h,1>;
       B - <w,2>, <j,3>, <i,1>, <h,1>
    b. <w,2>, <j,3>, <a,15>, <b,15>, <h,1>, <i,1>
    c. <x,8>, <y,8>, <i,9>, <a,15>, <b,15>, <j,1>, <h,1>
    d. <w,2>, <j,3>, <x,8>, <y,8>, <i,9>, <a,15>, <b,15>, <h,1>

2. A few key points are as follows
    a. Namespace names have to be unique
    b. Nesting of namespaces is allowed
    c. Classes within different namespaces can have the same name, they are referenced in tandem with the name of the namespace as shown below
    d. Sample global namespace usage below -

```
namespace SampleNamespace;

class AnotherSampleClass
{
    public void AnotherSampleMethod()
    {
        System.Console.WriteLine(
            "SampleMethod inside SampleNamespace");
    }
}
```

    e. The below code gives you a glimpse into why the same class name within different namespaces can be referenced as shown. This means that similar named variables, classes and functions within namespaces can be differentiated using the name of the namespace.

```
// main namespace
namespace Main_name
{

    // nested namespace
    namespace Nest_name
    {

        // class within nested namespace
        class Example
        {

            // Call to print
        public static void Main(string[] args){
            System.Console.WriteLine("Hello");
```

```
            NestWithinNest_name.Example.func();
          }

        }
            namespace NestWithinNest_name
        {
          class Example {
            ....
              static void func(){
              ...
            }
          }
        }
      }

}
```

f. References – https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/namespaces#136-namespace-member-declarations OR https://www.youtube.com/watch?v=mw254_XAnGU&ab_channel=Covalence

3.

A. The lifetime of a static local variable is the execution of the entire program; the storage for a static local variable is allocated at the beginning of program execution. However, a static local variable is invisible outside the function where the variable is declared. A static local variable has the scope of the function or block in which the variable is declared.

B. The scope of a static global variable is the scope of the file in which the variable is declared. It is invisible outside the file it is declared. If a local variable has the same name as the global variable, the global variable will be invisible in the scope of the local variable. The lifetime of a static global variable is the entire program execution. The extern keyword can be used to extend its scope to outside the file it is declared in. Note that all functions in the entire file can edit this variable.