Don Petersen

S2 Discussion Questions

1) What is the function of the *new* operator?

According to the Microsoft Developer Network website, the *new* operator is used to create objects and invoke constructors. An example of the *new* operator creating an object: Class1 obj = new Class1(); This makes the object obj from the class Class1. An example of the *new* operator invoking a default constructor: int i = new int(); In this statement i is initialized to 0. If no argument is provided with the class name, then the default constructor is used.

Microsoft Developer Network(msdn): *New Operator (C# Reference).* Retrieved from http://msdn.microsoft.com/en-us/library/fa0ab757.aspx.

2) What is a C# namespace? What purposes does it serve? How will you use namespaces in your programs?

The namespace keyword is used to declare a scope that contains a set of related objects. A namespace allows you to use related types or services that are under the same namespace. The textbook states, “For example, all text-based services are grouped in the Text namespace.”(Lewis, 2007) Namespaces are most commonly used using the *using* declaration. With using declarations you can still access the services in a namespace. The using declaration saves time because it allows you to not have to write the namespace for each line of code. So, instead of writing System.Text.Stringbuilder phrase = new System.Text.Stringbuilder(“words”); you can put the line using System.Text as one of the first lines in the program and that allows you to just write Stringbuilder.Text.Stringbuilder phrase = new Stringbuilder(“words”); This is how I imagine I will be using namespace for the most part.

 Lewis, John (2007, p120). C# Software Solutions: Foundations of Program Design. Boston, MA: Pearsons, Inc.

3) What is the scope of a variable? Do all variables have scope? Why are globally scoped variables frowned upon?

The scope of a variable is the area where that variable can be referenced. “The location at which a variable is declared defines its scope.”(Lewis, 2007) All variables have scope. Since the location defines the scope and the variable can only be referenced within its scope then it is in a sense protected from anything that is outside that scope. To make a variable’s scope global is to expose it to being referenced by all area of the program. The few things that have access to a variable the easier it will be to maintain control over that variable.

 Lewis, John (2007, p.165). C# Software Solutions: Foundations of Program Design. Boston, MA: Pearsons, Inc.

4) What are the "visibility modifiers?" To what may they be applied? What function do they serve?

Visibility modifiers are modifiers that control access to the members of a class. The reserved words public and private are visibility modifiers. If a member of a class has public visibility, it can be directly referenced form outside of the object, public variables violate encapsulation. If a member of a class has private visibility, it cannot be referenced outside of the object but only used anywhere inside the class definition, private variables enforce encapsulation. The function of visibility modifiers is to make the variable, method, or class only available to areas of the program that it would need to be.

 Lewis, John (2007, p167-168). C# Software Solutions: Foundations of Program Design. Boston, MA: Pearsons, Inc.

5) What is a constructor? What objective(s) should a constructor fulfill? Can you think of anything else a constructor might do?

A constructor is a method in the class which gets executed when its object is created. A constructor fulfills the role of initializing an object. “Whenever a class or struct is created, its constructor is called. A class or struct may have multiple constructors that take different arguments. Constructors enable the programmer to set default values, limit instantiation, and write code that is flexible and easy to read.” (msdn, 2014) Constructors give us a way to initialize default values for the class’s attributes so that the class contains the data to accomplish its purposed without causing an error.

 Microsoft Developers Network(msdn): *Constructors(C# Programming Guide)*. Retrieved from http://msdn.microsoft.com/en-us/library/ace5hbzh.aspx

Lewis, John (2007, p109). C# Software Solutions: Foundations of Program Design. Boston, MA: Pearsons, Inc.