Don Petersen

S2 Discussion Questions

1) Describe the process of creating an array. When is memory allocated for an array? Are the rules the same for all kinds of arrays?

To create an array, the reference to the array must be declared and then the array must be instantiated using the new operator. An example of the creation of the array is: int[] height = new int[11]; This example creates an array called height that contains 11 elements. When the new operator is used to instantiate an array, memory space is allocated to store the array. These rules can be applied to all arrays, arrays only differ in the type of variables and the parameters. (Lewis, 2007) Example: new int[11] is an integer array, while new float[5,10] is a two dimensional float array.

2) Can an entire array be passed to a method as an argument? If it is possible, how can this be accomplished?

Arrays are objects in C# and objects can be passed as parameters, so, yes, an entire array can be passed to a method as an argument. It is also possible to pass just an element of the array as an argument as well. This is done by very similarly to passing any argument to a method, example: Void PrintArray(int[] height) { }. In this example the array height is being passed as an argument to the method PrintArray. You can also initialize and pass a new array at once. (Lewis, 2007) Example: PrintArray(new int[] {1,2,3,4});

3) Explain the similarities and differences between arrays and ArrayLists. When might you use one in preference to the other?

Both Arrays and ArrayLists are objects that store a list of values. The main difference between an Array and an ArrayList is that an array can store a list that remains a fixed size. An ArrayList object can dynamically grow and shrink as needed. Also, the ArrayList is part of the Collections namespace so it requires a using line to access the class. An Array, then, would be used when you know the exact number of a particular set of variables you wish to store and you do not want that list to change. An ArrayList object is better suited when you either do not know how the list of elements will be or you want to list to be flexible in its size. (Lewis, 2007)

4) How does inheritance support software reuse and software maintenance?

Inheritance supports software reuse by allowing us to take a class and create another class from it that extends the use of the original class. Creating a derived class allows the new class to reuse, modify and extend the behavior that is defined in the parent class. Because the members of a parent class are inherited by the child class, all changes to the parent class members, change the members inherited by the child. This allows for easier maintenance because instead of having to make change to each class, you only have to make changes to the base class. (Lewis, 2007)

5) Are all members of a parent class inherited by its child classes? Explain your answer in detail and justify your reasoning.

“When you define a class to derive from another class, the derived class implicitly gains all the members of the base class, except for its constructors and destructors.” (MSDN 2010) All members of the parent class are inherited by the child class except for any variables and methods that are declared private in the parent class. However, the child class can still indirectly reference the private members of the parent class through the parent’s constructor by calling the base reference. (Lewis, 2007)

Lewis, J. (2007). C# Software Solutions: Foundations of Program Design.Boston, MA:Pearson Education.

Microsoft (2014). Inheritance (C# Programming Guide). Retrieved http://msdn.microsoft.com/en-us/library/ms173149.aspx