**C# Homework 08**

**Question 1**

What is the difference between deep copy and shallow copy?

**Answer**

Shallow Copy: Creating a new object and then coping the value type fields of the current object to the new object. Shallow Copy only copies the reference. Deep Copy: It is a process of creating a new object and then copying the fields of the current object to the newly created object to make a complete copy of the internal reference types. Deep Copy copies what the reference points to.

**Question 2**

What is the value of a reference after you declare and initialize it?

**Answer**

It is a memory address. A location of a heap.

**Question 3**

How do you declare a value type?

**Answer**

You can use literals to provide a value of a simple type. For example, ‘A’ is a literal of the type char and 2001 is a literal of the type int. You can declare constants of the simplest types with the const keyword. It’s not possible to have constants of other structure types.

**Question 4**

How do you declare a reference type?

**Answer**

If we define and assign a value to the variable like string name = “Suresh Dasari”; then the system will store the variable “name” in another location along with the memory address of the variable type. The memory for the actual Circle object is allocated only when the new keyword is used to create the object.

**Question 5**

Does C# allow you to assign NULL to a value type?

**Answer**

A value type cannot be assigned a null value. For example, int I = null will give you a compile error. C# 2.0 introduced nullable types that allow you to assign null to value type variables. Nullable<int> can be assigned any value from -2147483648 to 2147483648, or a null value.

**Question 6**

Can you assign a nullable value type to a non-nullable variable of the same type? Why or why not?

**Answer**

No, you cannot. If it were reversed you could. 2 pistol 22 and 44 2 cartridges 22 and 44 will the 22 cartridge fit into the 44 technically yes, but vice versa the answer is no.

**Question 7**

What is the difference between the stack and the heap?

**Answer**

In C# there are two places where an object can be stored—the heap and the stack. Objects allocated on the stack are available only inside of a stack frame (execution of method), while objects allocated on the heap can be accessed from anywhere.

**Question 8**

What does it mean when we say that all classes are specialized types?

**Answer**

When you create your own class you are creating a type that is doing its own thing.

**Question 9**

What does ref do?

**Answer**

The ref keyword indicates a value that is passed by reference. It is used in four different contexts: In a method signature and in a method call, to pass an argument to a method by reference.

**Question 10**

What does out do?

**Answer**

The out is a keyword in C# which is used for the passing arguments to methods as a reference type. It is generally used when a method returns multiple values. But before it returns a value to the calling method, the variable must be initialized in the called method.

**Question 11**

Describe boxing and unboxing in your own words.

**Answer**

Boxing is the process of converting a value type to the type object or to any interface type implemented by this value type. When the common language runtime (CLR) boxes a value type, it wraps the value inside a System. Object instance and stores it on the managed heap. Unboxing extracts the value type from the object.

**Question 12**

What does cast do?

**Answer**

Cast is a method by which a value is converted from one data type to another. Cast is an explicit conversion by which the compiler is informed about the conversion and the resulting possibility of data loss.