**Questions for this assignment**

What are parameter modifiers in C#?

What is the 'ref' modifier?

What is the ‘out’ modifier?

What is the syntax for ‘ref return’ in C#?

What is a local function in C#?

What is the syntax for defining a local function in C#?

What is recursion in C#?

What is the base case in a recursive method?

What is method overloading in C#?

What is the advantage of method overloading?

What is the difference between an instance method and a static method in C#?

When should you use an instance method and when should you use a static method?

What are parameter modifiers in C#?

Parameter modifiers are used to modify the behavior of a parameter in a method. There are three types of parameter modifiers in C#: ref, out, and in.

What is the 'ref' modifier?

The ref modifier is used to pass a parameter by reference. This means that any changes made to the parameter within the method will also be reflected in the calling code.

What is the ‘out’ modifier?

The out modifier is used to pass a parameter as an output parameter. This means that the parameter does not have to be initialized before it is passed to the method, and any changes made to the parameter within the method will be returned to the calling code.

What is the syntax for ‘ref return’ in C#?

The syntax for ‘ref return’ in C# is:

ref [return\_type] [method\_name] (parameters)

{

// Method body

}

What is a local function in C#?

A local function is a method that is defined within the body of another method. It has access to the local variables and parameters of the containing method.

What is the syntax for defining a local function in C#?

The syntax for defining a local function in C# is:

[return type] LocalFunctionName(parameters)

{

// Function body

}

What is recursion in C#?

Recursion is a technique in programming where a method calls itself to solve a problem. It is useful for solving problems that can be broken down into smaller sub-problems.

What is the base case in a recursive method?

The base case is the condition in a recursive method that stops the recursion. It is the simplest case that can be solved without recursion.

What is method overloading in C#?

Method overloading is a feature in C# that allows multiple methods with the same name but different parameters to be defined in the same class.

What is the advantage of method overloading?

Method overloading allows developers to write more flexible and expressive code by providing different ways to call the same method with different parameters.

What is the difference between an instance method and a static method in C#?

An instance method is associated with an instance of a class and can access instance fields, while a static method belongs to the class itself and can only access static fields.

When should you use an instance method and when should you use a static method?

You should use an instance method when you need to access instance fields or perform an operation on a specific instance of a class.

You should use a static method when you need to perform an operation that does not depend on a specific instance of a class or when you need to access static fields.