**C# Homework03**

**Question 1**

What is a method?

**Answer**

A method is a code block that contains a series of statements. In C# every executed instruction is performed in the context of a method. A named block of code.

**Question 2**

What is the difference between a function and a procedure?

**Answer**

A function must return value but in stored procedures it is optional, a procedure can return 0 or n values. Functions can have only input parameters for it, whereas procedures can have input/output parameters. Functions can be called from a procedure whereas procedures cannot be called from a function.

**Question 3**

What does the return statement do?

**Answer**

The return statement terminates execution of the method in which it appears and returns control to the calling method. It can also return an optional value. If the method is a void type, the return statement can be omitted.

**Question 4**

What is an expression bodied method?

**Answer**

An expression bodied method consists of a single expression that returns a value whose type matches the method’s return type, or, for methods that return void, that performs some operation. It also defines a DisplayName method that displays a name to the console.

**Question 5**

What is the scope of a variable?

**Answer**

The scope of a variable is a region of code that indicates where the variables are being accessed.

**Question 6**

What is an overloaded method?

**Answer**

Two or more than two methods having the same name but different parameters are what we call method overloading. It can be performed by changing the number of arguments and the data type of the arguments. You can only overload the method with a different parameter list

**Question 7**

How do you call a method that requires arguments?

**Answer**

When a caller invokes the method, it provides concrete values, called arguments, for each parameter. The arguments must be compatible with the parameter type, but the argument name, if one is used in the calling code, does not have to be the same as the parameter named defined in the method.

**Question 8**

How do you write a method, that is, specify the method definition, that requires a parameter list?

**Answer**

By using the params keyword to indicate that a parameter is a parameter array, you allow you method to b called with a variable number of arguments. The parameter tagged with the params keyword must be an array type, and it must be the last parameter in the method’s parameter list.

It has to have a return type, method name, a parameter list, and a body.

**Question 9**

How do you specify a parameter as optional when defining a method?

**Answer**

Optional parameters are defined at the end of the parameter list, after any required parameters. If the caller provides an argument for any one of a succession of optional parameters, it must provide arguments for all preceding optional parameters. Comm-separated gaps in the argument list are not supported. By using the assignment operator.

**Question 10**

How do you pass an argument to a method as a named parameter?

**Answer**

Named arguments enable you to specify an argument for a particular parameter by associating the argument with the parameter’s name rather than with the parameter’s position in the parameter list. Optional arguments enable you to omit arguments for some parameters. Use a colon

**Question 11**

How do you return values from a method? Can you return multiple values from a method, and if so, how?

**Answer**

They use out and ref parameters and keyvaluepair. Methods return only one value. This value can be an object with multiple fields. Many options are available for returning multiple values from a method. If you need to return multiple values, you can use out parameter or return a type instance containing all the values. We benchmark and compare methods that return multiple in values. Use a return statement.

**Question 12**

What is a tuple? How do you define a method that returns multiple values? Give an example of a method that returns multiple values other than the example in the book.

**Answer**

A tuple is a data structure that contains a sequence of elements of different data types. It can be used where you want to have a data structure to hold an object with properties, but you don’t want to create a separate type for it.

**Question 13**

Examine the method definition on page 83 of the book. Desk check the execution of this method. What do you discover? This is called recursion.

**Answer**

Desk checking is a manual (non computerized) technique for checking the logic of an algorithm.

**Question 14**

How does the compiler resolve an ambiguity between named arguments and optional parameters?

**Answer**

Named arguments enable you to specify an argument for a particular parameter by associating the argument with the parameter’s name rather than with the parameter’s position in the parameter list. Optional arguments enable you to omit arguments for some parameters.