**C# Assignment Questions**

**Que1= What is the case used for constant/global variable, camelCase or Pascal Case and also define it.**

Ans1= The naming conventions for constants and global variables are:

Pascal Case: The first letter of each word is capitalized, and there are no spaces or underscores between words.

Eg:- const int MaxValue = 100;

**Que2= Difference b/w Variable & property and when should use variable & property?**

Ans2= **Variable:** A variable is nothing but a name given to a storage area that our programs can manipulate.

Eg:- int i, j, k;

char c, ch;

float f, salary;

double d;

Property:- A property is a member of a class which is used to set and get the data from a data field of a class. They use accessors through which the values of the private fields can be read, written or manipulated. Properties do not name the storage locations. Instead, they have accessors that read, write, or compute their values.

Note:- In property we can put some validations or can do any operation on the field before set or get of value, but in case of member variable it is not there.

**Que3= Explain the read-only variable.**

## Ans3= Runtime constants (readonly): The Run time constants are declared by using the readonly keyword, whose value can not be changed during the execution of the program.

Eg:- int readonly a=0;

Or

int readonly a;

a=0;

1. Readonly allows readonly constant and non read-only constant variables into the expression.

Eg:- int readonly a=10;

int b=1;

int readonly c=a+b;

1. Readonly can be declared only at the class level, not inside the method.
2. Readonly can not be declared using static keywords because they are, by default, static.
3. Readonly constant's value can be set through the reference variable.
4. Readonly constant variables are runtime time constant variables.

Note:- We can use **Readonly**when its value is not an absolute constant, which means it can be changed frequently, such as dollars vs INR.

**Que4= Difference b/w Functions and Subroutines.**

Ans4= **Function:** A function is used when a value is returned to the calling routine.

**Invoking a Function:** A function by definition has a return value. Therefore, a function call must be assigned to a variable of the type that the function returns.

**Subroutine:** A subroutine is used when a desired task is needed, but no value is returned.

**Invoking a Subroutine:** A subroutine is used when a series of steps are required but no value is returned to the routine that called the subroutine. Subroutines are invoked using a subroutine name

**Que5= Write a program on bitwise operators (at least one, we can use more than one).**

Ans5= The File has been attached in the mail for this program.

**Que6= What is +12 and -12 Time zone ? Are they same?**

Ans6= **+12 Time Zone:** This refers to a time zone that is 12 hours ahead of Coordinated Universal Time (UTC+12).

These regions are ahead of UTC by 12 hours. For example, when it is 12:00 PM UTC, it is 12:00 AM (midnight) in the +12 time zone.

**-12 Time Zone:** This refers to a time zone that is 12 hours behind Coordinated Universal Time (UTC-12). It is one of the time zones furthest behind UTC. The region most commonly associated with this time zone is:

In the -12 time zone, when it is 12:00 PM UTC, it is 12:00 AM (midnight) the previous day in the -12 time zone.

So, +12 and -12 time zones are not the same; they are on opposite sides of the world and have a 24-hour time difference between them.

**Que7= Why Main method has arguments and why we used arguments ?**

Ans7= In C#, the Main method of a console application can have arguments because it allows us to pass command-line arguments to our program when we run it. These command-line arguments provide a way to input data or configure the behavior of our program from outside the application code.

**Passing Input Data:** Command-line arguments allow us to pass input data to our program when we execute it. This input data can be used by our program to perform specific tasks or make decisions. For example, we can pass a file path, a user name, or any other data as arguments.

**Configuring Program Behavior**: By providing different arguments when running the program, we can change how it operates.

**Accessing External Data**: Sometimes, we want to provide our program with data from an external source or another program.

**Que8= Difference b/w ref and out keyword.**

Ans8= **ref keyword:**

1. ref keyword is used when a called method has to update the passed parameter.
2. ref keyword is used to pass data in bi-directional way.
3. Before passing a variable as ref, it is required to be initialized otherwise compiler will throw error.
4. In called method, it is not required to initialize the parameter passed as ref.

**Out keyword:**

1. out keyword is used when a called method has to update multiple parameter passed.
2. out keyword is used to get data in uni-directional way.
3. No need to initialize variable if out keyword is used.
4. In called method, it is required to initialize the parameter passed as out.