Name:	**Miandad Khan**
ID:	**14130**
Programm:	**BS CS**
Subject	**Visual Programming**
Date:	**18/8/2020**

Q1. How to write Hello program in C# and explain in detail?

"Hello World!" in C#

```
// Hello World! program
namespace Hello World
{
    class Hello {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");
        }
    }
}
```

When you run the program, the output will be:

```
Hello World!
```

`// Hello World! Program`

`//` indicates the beginning of a comment in C#. Comments are not executed by the C# compiler.

`namespace HelloWorld{...}`

The namespace keyword is used to define our own namespace. Here we are creating a namespace called `HelloWorld`.

`class Hello{...}`

The above statement creates a class named - `Hello`in C#. Since, C# is an object-

oriented programming language, creating a class is mandatory for the program's execution.

`static void Main(string[] args){...}`

`Main()` is a method of class Hello. The execution of every C# program starts from the `Main()` method. So it is mandatory for a C# program to have a `Main()` method.

The signature/syntax of the `Main()` method is:

```
static void Main(string[] args)
{
    ...
}
```

`System.Console.WriteLine("Hello World!");`

For now, just remember that this is the piece of code that prints **Hello World!** to the output screen.

Q2. a. Write a simple program C # by taking 2 different strings in text boxes concatenate them and display with the help of message box and explain in detail.

As you know, the best way to concatenate two strings in C programming is by using the strcat() function. However, in this example, we will concatenate two strings manually.

Concatenate Two Strings Without Using strcat()

```c
#include <stdio.h>
int main() {
  char s1[100] = "programming ", s2[] = "is awesome";
  int length, j;

  // store length of s1 in the length variable
```

```
  length = 0;
  while (s1[length] != '\0') {
    ++length;
  }

  // concatenate s2 to s1
  for (j = 0; s2[j] != '\0'; ++j, ++length) {
    s1[length] = s2[j];
  }

  // terminating the s1 string
  s1[length] = '\0';

  printf("After concatenation: ");
  puts(s1);

  return 0;
}
```

```
After concatenation: programming is awesome
```

Here, two strings `s1` and `s2` and concatenated and the result is stored in `s1`.

It's important to note that the length of `s1` should be sufficient to hold the string after concatenation. If not, you may get unexpected output.

b. Write about different types of type conversions in available in C#?

What is Typecasting in C?

Typecasting is converting one data type into another one. It is also called as data conversion or type conversion. It is one of the important concepts introduced in 'C' programming.

'C' programming provides two types of type casting operations:

1. Implicit type casting
2. Explicit type casting

Implicit type casting

Implicit type casting means conversion of data types without losing its original meaning. This type of typecasting is essential when you want to change data types **without** changing the significance of the values stored inside the variable.

Implicit type conversion happens automatically when a value is copied to its compatible data type. During conversion, strict rules for type conversion are applied. If the operands are of two different data types, then an operand having lower data type is automatically converted into a higher data type.

Explicit type casting

In implicit type conversion, the data type is converted automatically. There are some scenarios in which we may have to force type conversion. Suppose we have a variable div that stores the division of two operands which are declared as an int data type.

```
int result, var1=10, var2=3;
result=var1/var2;
```

In this case, after the division performed on variables var1 and var2 the result stored in the variable "result" will be in an integer format. Whenever this happens, the value stored in the variable "result" loses its meaning because it does not consider the fraction part which is normally obtained in the division of two numbers.

To force the type conversion in such situations, we use explicit type casting.

It requires a type casting operator. The general syntax for type casting operations is as follows:

```
(type-name) expression
```

Here,

- The type name is the standard 'C' language data type.
- An expression can be a constant, a variable or an actual expression.

Let us write a program to demonstrate implementation of explicit type-casting in 'C'.

```
#include<stdio.h>
int main()
{
	float a = 1.2;
	//int b  = a;
	//Compiler will throw an error for this
	int b = (int)a + 1;
	printf("Value of a is %f\n", a);
```

```
	printf("Value of b is %d\n",b);
	return 0;
}
```

Output:

```
Value of a is 1.200000
Value of b is 2
```

Q3.

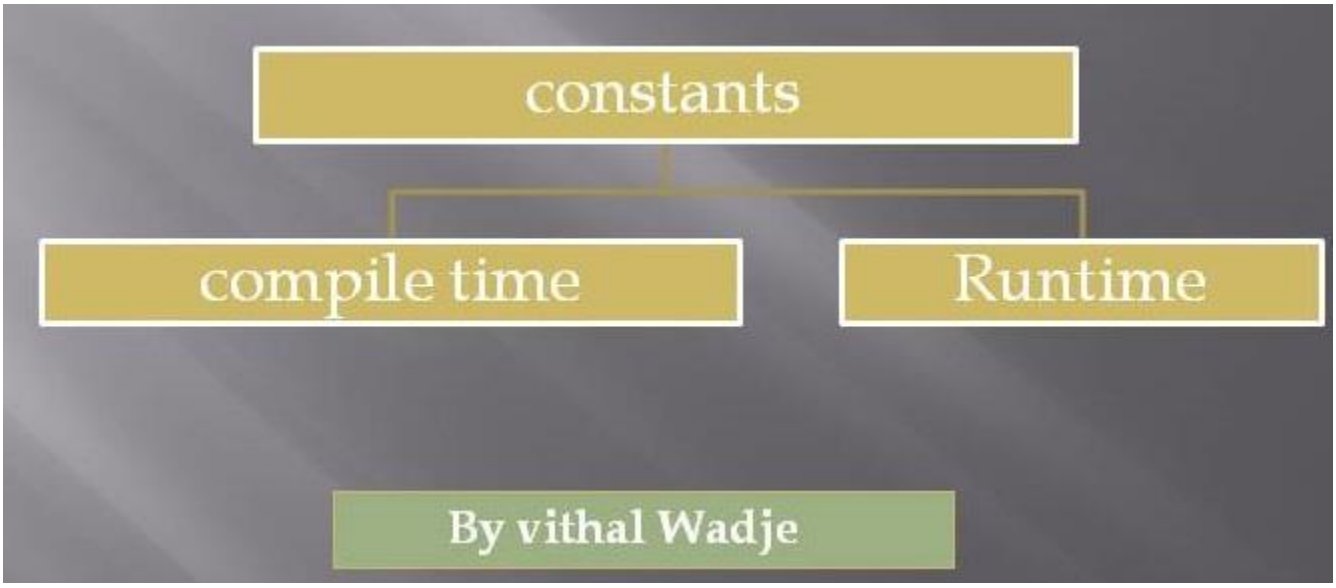
A: What is constant in C# discuss in detail ?

What is a Constant variable:-

A variable whose value cannot be changed during the execution of the program is called a constant variable.

In the above definition, the value cannot be changed during execution of the program, which means we cannot assign values to the constant variable at run time. Instead, it must be assigned at the compile time.

Constants are of the following types:



In the above diagram, we can see the Constants are of two types

1. **Compile time constants** (const)
2. **Runtime constants** (Readonly)

- **Compile time constants**

The compile time constants are declared by using the const keyword, in which the value cannot be changed during the execution of the program.

Syntax

```
int const a=10;
```

Runtime constants (Readonly)

The Run time constants are declared by using the Readonly keyword which value cannot be changed during the execution of the program.

Syntax
int Readonly a; or
int Readonly a=0;

B: Write a program on string literal and explain in detail?

In c#, the string is a keyword that is useful to represent a sequential collection of characters that is called a text and the string is an object of System.String type.

In c#, we use string keyword to create string variables to hold the particular text which is a sequential collection of characters.

The string variables in c# can hold any kind of text and by using Length property we can know that the number of characters the string variable is holding based on our requirements.

C# String Declaration and Initialization

The following are the different ways of declaring and initializing string variables using **string** keyword in c# programming language.

```
// Declare without initializing.
string str1;
```

```
// Declaring and Initializing
string str2 = "Welcome to Tutlane";
String str3 = "Hello World!";
// Initialize an empty string.
string str4 = String.Empty;
// Initialize to null.
String str5 = null;
// Creating a string from char
char[] letters = { 'A', 'B', 'C' };
string str6 = new string(letters);
```

If you observe above code snippet, we created a string variables using **string** and **String** keywords with or without initializing a values based on our requirements.