**Constants & Literals In C# Programming**

A constant has a fixed value that remains unchanged throughout the program while a literal provides a mean of expressing specific values in a program.

**Example**

* Consider a code that calculates the area of the circle.
* To calculate the area of the circle, the value of pi and radius must be provided in the formula.
* The value of pi is a constant value.
* This value will remain unchanged irrespective of the value of the radius provided.
* Similarly, constants in C# are fixed values assigned to identifiers that are not modified throughout the execution of the code.
* They are defined when you want to preserve values to reuse them later or to prevent any modification to the values.

**Constants In C#**

* A constant has a fixed value that remains unchanged throughout the program.
* In C#, you can declare constants for all data types.
* You have to initialize a constant at the time of its declaration.
* Constants are declared for **value types** rather than for **reference types**.
* To declare an identifier as a constant, the **“const”** keyword is used in the identifier declaration. The compiler can identify constants at the time of compilation, because of the **“const”** keyword.

**The following syntax is used to initialize a constant:**

const<data type><identifier name> = <value>;

**where,**

* **const**: Keyword denoting that the identifier is declared as constant.
* **data type:** Data type of constant.
* **identifier name:** Name of the identifier that will hold the constant.
* **value**: Fixed value that remains unchanged throughout the execution of the code.

**The following code declares a constant named \_pi and a variable named radius to calculate the area of the circle:**

const float \_pi = 3.14F;

float radius = 5;

float area = \_pi \* radius \* radius;

Console.WriteLine(“Area of the circle is ” + area);

**In Above Code,**

* A constant called \_pi is assigned the value 3.14, which is a fixed value. The variable, radius, stores the radius of the circle. The code calculates the area of the circle and displays it as the output.

**Literals In C# Programming**

* A literal is a static value assigned to variables and constants.
* Numeric literals might suffix with a letter of the alphabet to indicate the data type of the literal.
* This letter can be either in upper or lowercase.

**Example**

**For example, in the following declaration,**

string bookName = “Csharp”

Csharp is a literal assigned to the variable bookName of type string.

**Types Of Literals**

* Boolean Literal
* Integer Literal
* Real Literal
* Character Literal
* String Literal
* Null Literal

**Boolean Literal**

* Boolean Literal: Boolean literals have two values, true or false. For example,
  + bool val = true;
* where,
  + true: Is a Boolean literal assigned to the variable val.

**Integer Literal**

* Integer Literal: An integer literal can be assigned to int, uint, long, or ulongdata types. Suffixes for integer literals include U, L, UL, or LU. U denotes uint or ulong, L denotes long. UL and LU denote ulong. For example,
  + long val = 53L;
* Where,
  + 53L: Is an integer literal assigned to the variable val.

**Real Literal**

* Real Literal: A real literal is assigned to float, double (default), and decimal data types. This is indicated by the suffix letter appearing after the assigned value. A real literal can be suffixed by F, D, or M. F denotes float, D denotes double, and M denotes decimal. For example,
  + float val = 1.66F;
* Where,
  + 1.66F: Is a real literal assigned to the variable val.

**Character Literal**

* Character Literal: A character literal is assigned to a char data type. A character literal is always enclosed in single quotes. For example,
  + char val = ‘A’;
* Where,
  + A: Is a character literal assigned to the variable val.

**String Literal**

* String Literal: There are two types of string literals in C#, regular and verbatim. A regular string literal is a standard string. A verbatim string literal is similar to a regular string literal but is prefixed by the ‘@’ character. A string literal is always enclosed in double quotes. For example,
  + string mailDomain = “@gmail.com”;
* Where,
  + @gmail.com: Is a verbatim string literal.

**Null Literal**

* Null Literal: The null literal has only one value, null. For example,
  + string email = null;
* Where,
  + null: Specifies that e-mail does not refer to any objects (reference).