***Name-Gaurav Kumar Singh***

**Properties in C#**

\*A property acts as a wrapper around a field (a field is a variable of any type that is declared directly in a class).

\*We have two main type of properties i.e,

1. **Read Only Property** – this property only allows you to retrieve the value of the field. To create a read-only property, you should define the accessor. For example:-

Class Example

{

String name;

Public string Name

{

get {return name}

}

}

1. **Write only Property-** This property allows you to only change the value of field. To create this property set accessor must be defined. For example:-

Class Example

{

String name;

Public string Name

{

Set {name=value;}

}

}

1. **Read Write Property-** This property allows to assign and read values of a field. To create this property both get and set accessor must be defined. For example

Class Example

{

String name;

Public string Name

{

get {return name;}

set {name = value;}

}

}

**\*Auto implemented Property:**

Public int Name { get; set; }

**\*Static property:**

public class Singleton

{

private static Singleton instance = new Singleton ();

private Singleton () { }

public static Singleton GetInstance

{

get { return instance;}

}

}

**Constructors in C#**

\*Constructor is a special method that is used to initialize object.

\*Constructor is use to Create object and initialize it.

\*It invokes automatically at the time of object creation and there is no return type of a constructor.

Types of Constructors are :-

1. **Default Constructor :-**

* There is a default constructor if no constructor is defined by user and it automatically invoked at the time of object creation.

1. **Parameter less Constructors :-**

* Constructor which has no argument is known as parameter less constructror and it is invoked at the time of creating object.

1. **Parameterized Constructor:-**

* Constructor which has parameters are known as parameterized constructors.
* It is used to provide different values to different objects.

1. **Static Constructors:-**

* It is the constructor that should be called before the first object created.
* It is only called once and we cannot overload static constructors.
* Only single static constructors can be used in a class.

**5.Private Constructor:**-

* These are used to prevent creating instances of a class when there r when there are no instance fields or methods such as Math Class.

**6.Copy Constructors :-**

* Constructor that takes reference of another object and copies the data of one object into another object.