BASICS OF CLASSES

What is a class

- A building block of an application.

Anatomy of a Class

- **Data:** represented by fields
- **Behaviour:** represented by methods/functions

```
class Person
{
    // fields/attribute which represent Data
    string Name;
    byte Age;
    float Height;
    byte Weight;

    // method/function which represent Behaviour
    void Walk() { }
    void Talk() { }
    void Eat() { }
    void Sleep() { }
}
```

```
class Post
{
    // fields/attribute which represent Data
    string Title;
    string Description;
    DateTime Date;

    // method/function which represent Behaviour
    void Publish() { }
    void Like() { }
    void Comment(message) { }
}
```

Object

- An instance of a class

Creating Object

```
Person person = new Person();
// using var
var person = new Person();
```

Class Members

- **Instance:** accessible from an object.
- **Static:** accessible from the class.

Why use static members?

- To represent concepts that are singleton. e.g DateTime, Console.WriteLine()

Constructor

- A method that is called when an instance of a class is created.

Why Constructor?

- To put an object in an early state.
- To iniatialize some of the fields in the class.

Object Initializers

- A syntax for quickly initialising an object without the need to call one of its constructors.

Why?

- To avoid creating multiple constructors

Methods

- Signature of Methods
- Method Overloading
- Params modifier
- Ref modifier
- Out modifier

Access Modifier

- A way to control access to a class and/or its members.

Why?

- To create safety in our programs.

```

```

```
public class Person
{
    private DateTime _birthdate;

    public void SetBirthdate(DateTime birthdate)
    {
        this._birthdate = birthdate;
    }

    public DateTime GetBirthdate()
    {
        return _birthdate;
    }

}
```

Properties

- A class member that encapsulates a getter/setter for accessing a field.

Why?

- To Create a getter/setter wth less code.

```
public class Person
{
    private DateTime _birthdate;

    public DateTime Birthdate
    {
        get { return _birthdate; }
        set { _birthdate = value; }
    }
}
```

Indexer

- A way to access elements in a class that represents a list of values.

How?

```
public class HttpCookie
{
```

```
    public string this[string key]
    {
        get { }
        set { }
    }
}
```