**Assignment Week 1 Day 1**

Question 2 – **Difference between “==” operator, object.Equals methods and object.ReferenceEquals method.**

Ans - **Object.ReferenceEquals() Method** is used to determine whether the specified Object instances are the same instance or not. This method **cannot be overridden**. So, if a user is going to test the two objects references for equality and he is not sure about the implementation of the Equals method, then he can call the ReferenceEquals method.

class Program

{

static void Main(string[] args)

{

object v1 = new object();

person p1 = new person();

object v3 = v1;

object v2 = new object();

bool s = ReferenceEquals(v1, p1);

Console.WriteLine(ReferenceEquals(v1, v2));

Console.WriteLine(ReferenceEquals(v1, v3));

Console.WriteLine(s);

Console.WriteLine(v1==p1);

Console.WriteLine(v1.Equals(p1));

}}

class person

{

}

// Output

False

True

False

False

False

class Program

{

static void Main(string[] args)

{

// Declaring and initializing value1

object value1 = null;

// Declaring and initializing value2

object value2 = null;

bool status = Object.ReferenceEquals(value1, value2);

Console.WriteLine($"Value 1 is equal to Value 2 = {status}");

}

}

// Output

Value 1 is equal to Value 2 = true

The == Operator compares the reference identity while the Equals() method compares only contents. Both of them can be overridden.

Eg

static void Main(string[] args)

{

string name = "Tavish";

string myName = name;

Console.WriteLine("== operator result is {0}", name == myName);

Console.WriteLine("Equals method result is {0}", name.Equals(myName));

Console.ReadKey();

}

// Output

== operator result is True

Equals method result is False

static void Main(string[] args)

{

object name = "tavish";

string myName = new string("tavish");

Console.WriteLine("== operator result is {0}", name == myName);

Console.WriteLine("Equals method result is {0}", myName.Equals(name));

}

//Output

== operator result is False

Equals method result is True

Regarding null objects –

static void Main(string[] args)

{

object v1 = null;

object v2 = null;

Console.WriteLine("== operator result {0}", v1 == v2);

Console.WriteLine("Equals method result is {0}", v1.Equals(v2));

Console.WriteLine("Reference Equals result is {0}", ReferenceEquals(v1,v2));

}

// True

// Null Reference Exception

Equals is an instance method that takes one parameter (which can be null). Since it is an instance method (must be invoked on an actual object), it can't be invoked on a null-reference.

ReferenceEquals is a static method that takes two parameters, either / both of which can be null. Since it is static (not associated with an object instance), it will not throw a NullReferenceException under any circumstances.

==is an operator, that, in this case (object), behaves identically to ReferenceEquals. It will not throw a NullReferenceExceptioneither.