**Output Parameters (out)**

* Used to return multiple values.
* The method must assign a value to the parameter before returning.

void GetValues(out int a, out int b)

{

a = 10;

b = 20;

}

int x, y;

GetValues(out x, out y);

Console.WriteLine($"x: {x}, y: {y}"); // Output: x: 10, y: 20

**Difference between ref and out**:

* ref requires initialization before passing.
* out does not require initialization before passing.

**Multiple Values using out Parameters**

void Divide(int dividend, int divisor, out int quotient, out int remainder)

{

quotient = dividend / divisor;

remainder = dividend % divisor;

}

int q, r;

Divide(10, 3, out q, out r);

Console.WriteLine($"Quotient: {q}, Remainder: {r}");

// Output: Quotient: 3, Remainder: 1

**Multiple Values using Tuples (Preferred Way)**

(string, int) GetPersonInfo()

{

return ("Alice", 25);

}

var person = GetPersonInfo();

Console.WriteLine($"Name: {person.Item1}, Age: {person.Item2}");

// Output: Name: Alice, Age: 25

**Returning Objects (Best for Complex Data)**

class Person

{

public string Name { get; set; }

public int Age { get; set; }

}

Person GetPerson()

{

return new Person { Name = "Alice", Age = 25 };

}

Person p = GetPerson();

Console.WriteLine($"Name: {p.Name}, Age: {p.Age}");