



High School Programming

Lecture: 15

WELCOME TO



Mahinur Rahaman Hridoy

*BSc in CSE, Diploma
Software Developer*



Recap Previous Lecture



Method

Agenda

 Methods Overloading

Methods

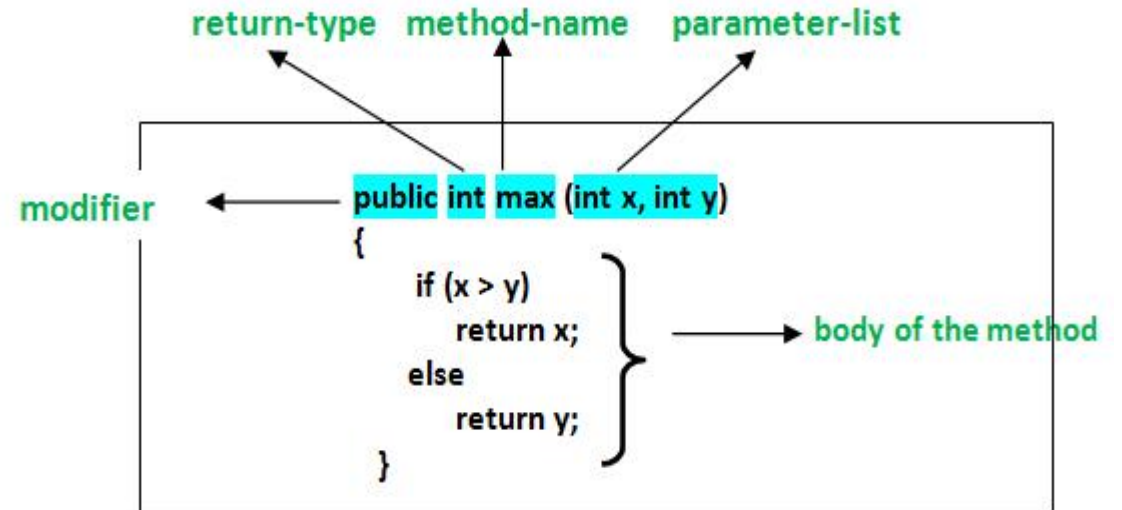
-- Methods are used to perform certain actions, and they are also known as functions. A method is a block of code which only runs when it is called.

Why use methods?

- ✓ To reuse code: define the code once, and use it many times.

There are two types of methods:

1. Parameterized Method
2. Non Parameterized Method
3. Method Overloading



Method Syntax

```
//Non parameterized  
<access-specifier> <return-type>  
MethodName()  
{  
  // function body  
  // return statement  
}
```

```
//Parameterized  
<access-specifier> <return-type>  
MethodName(parameters)  
{  
  // function body  
  // return statement
```

Method Example

//Non parameterized

```
private void Show()  
{  
    Console.WriteLine("Put your name here");  
}
```

//Parameterized

```
private void Show(string name)  
{  
    Console.WriteLine(name);  
}
```

Method Overloading

-- With method overloading, multiple methods can have the same name with different parameters:

Example:

```
int MyMethod(int x)
```

```
float MyMethod(float x)
```

```
double MyMethod(double x, double y)
```


Method Parameters

Parameters and Argument:

--When a **parameter** passed to the method, it is called an **argument**.

Example:

```
static void MyMethod(string fname)
{
    Console.WriteLine(fname + " Refsnes");
}
```

```
static void Main(string[] args)
{
    MyMethod("Liam");
    MyMethod("Jenny");
    MyMethod("Anja");
}
```

```
// Liam Refsnes
// Jenny Refsnes
// Anja Refsnes
```

Multiple Parameters

--You can have as many parameters as you like, just separate them with commas:

Example:

```
static void MyMethod(string fname, int age)
{
    Console.WriteLine(fname + " is" + age );
}
```

```
static void Main(string[] args)
{
    MyMethod("Liam", 5 );
    MyMethod("Jenny", 8 );
    MyMethod("Anja", 31 );
}
```

```
// Liam Refsnes 5
// Jenny Refsnes 8
// Anja Refsnes 31
```

Default Parameters Value

--You can also use a default value, by using equal sign(=).

Example:

```
static void MyMethod(string country = "Norway")
{
    Console.WriteLine( country );
}
```

```
static void Main(string[] args)
{
    MyMethod("Sweden");
    MyMethod("India" );
    MyMethod();
    MyMethod("USA" );
}
```

```
// Sweden
// India
// Norway
// USA
```

Return Values

--If you want to method to return s value, you can use the return keyword inside the method:

Example:

```
static int MyMethod(int x)
{
    return 5 + x;
}

static void Main(string[] args)
{
    Console.WriteLine(MyMethod(3));
}

// Outputs 8 (5 + 3)
```

Named Arguments

- It is also possible to send arguments with the *key:value* syntax.
- That way, the order of argument does not matter.

Example:

```
static void MyMethod(string child1, string child2, string child3)
{
    Console.WriteLine("The youngest child is: " + child3);
}

static void Main(string[] args)
{
    MyMethod(child3: "John", child1: "Liam", child2: "Liam");
}

// The youngest child is: John
```

Useful Resource



Contact Me

Phone

+8801321869515

Website

www.mrhridoymc.com

Mail

mrhridoymc@gmail.com

Facebook

[Mahinur Rahaman Hridoy](#)

Thank You