

Chapter 6:-

Defining a Method

Def: is a way to make an operation more simple in Maintenance operation and a Method have a signature i contains her name and her Parameters and we can write it as the following Syntax :-

modifier returnType Methodname(Parameters){
method body}

notes: the method Signature can be different in its Parameters and order of them like the following

Syntax :- Overloading
~~max(int num)~~ / ~~max(int num, double n)~~

notes: Parameters Cannot be declared with one declaration as following Syntax :-

~~max(int num1, num2)~~ ✓ max(int num1, int n2)

Def: The Overloading Principle is to make two methods with the same name but with out same Parameters

Calling a Method

Def: is an operation to call or invoke the method and it's consist of Passing The arguments as ~~as~~ parameters and we can invoke our method as the following
Syntax:

method name(arguments);

note any method that return Value must contains return Statement

Void Method

Def: is a method that return Void datatype and it's return statement make it Terminates and it's useful in printing Strings like `Println();` and we can write it as the following Syntax:

modifier void Methodname(Parameters){
body of method;
}

note when two methods have the same name and the same argument without same order it causes Syntax Error

Passing Arguments by Values

Def.

Passing by Value is to assign the value of the argument to the parameter not to assign the variable to it like the following Syntax:

```
int a, b;
```

```
a = 5;
```

```
b = 6;
```

```
max(a, b);
```

```
public static int max(int a, int b) {
```

```
    if (a > b)
```

```
        return a;
```

```
    else
```

```
        return b;
```

Output: ~~return b~~

Notes: Reusability Principle is a fundamental rule for using methods more than once.

Scope Of Variables

[1] Local Variable \Rightarrow is a variable which in a scope {}.

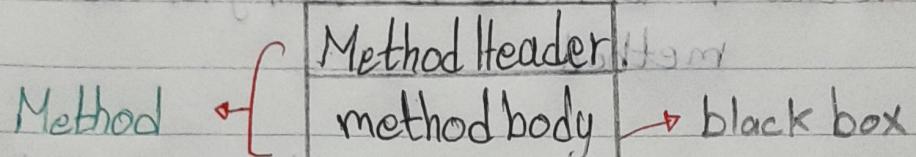
[2] Global Variable \Rightarrow is a variable start with static keyword.

notes

you can't use a local variable out of its scope

Method Abstraction and Stepwise Refinement :

Def: Abstraction has many meanings in Programming but in methods it's mean to separate the implementation of the method from user's sight



And we have some benefits of stepwise Refinement :

- [1] Simpler Program
- [2] Reusing Methods
- [3] Easier Developing debugging and Testing
- [4] Better Facilitating Team work

Recursion Principle :

Def: is to make a method recall it's self by itself as
The following Syntax :

```
public static void Print() {  
    S.O.Println("Hi");  
    Print();  
}
```

notes

The Preceding Code makes a stackOverflowError

and we should make two types of cases to generate a recursive method

- [1] **base Case** \Rightarrow To make method terminates depend on some conditions
- [2] **Recursive Case** \Rightarrow To make re invoking for the method

notes Recursion Principle depends on suspend some instructions to execute them in the last steps