METHODS

* Class contains METHODS and VARIABLES.
* Variables hold the data and values with which we are going to process.
* Methods are the actions that we are going to do with the Values.
  + - Ex: main method; Println

Methods are the blocks of code which will help me to do an action on the codes level.

Variable = datatype method = Action

How to write a method:

We should conclude few things:

* + - * Access Modifier
      * Return Type
      * Methodname(arg)

Syntax of a Method:

Text

Description automatically generated with medium confidence

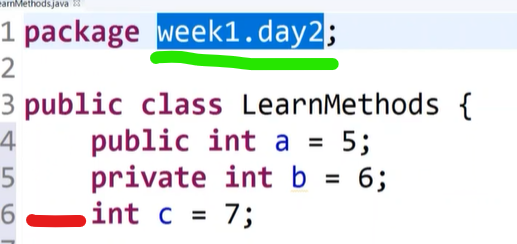
* To perform Actions
* Things that are required to create a method.
  + **Access Specifier** (public, private, default, protected)
  + **Return Type** (void, integer, string, Boolean ,….)
  + **Method name** (launchBrowser)
  + Input arguments.

**Access Modifier**:

Accessing something restriction of accessing of a particular thing is called Accessmodifier.

* In java we have a four diff accessmodifier

1. Public 🡪 If you a declare a method or a variable as public it can be used by anyone in the project level.
2. Private 🡪 If you a declare a method or a variable as private it can be used within the class.
3. Package/no modifier or default 🡪 if you declare a variable without any method then can be accessed within the package.



1. Protected 🡪week no 3. --> involves some OOP’s concept.

Return Type:

What you’re getting as the output is said to be return type.

* For every action there will be a return. 🡪 Method with return
* For certain actions that will not give you any return. 🡪 Method without return

VOID: the things without output🡪action without output🡪there is nothing to return.

A close up of a computer screen

Description automatically generated

Note: Here the Return type is said to be VOID, so we are print the name

Print is an action which is acting on the return type of VOID.

A screenshot of a computer program

Description automatically generated

* There is no output for the VOID type.

If you’re getting an output then we should be able to say what type of output that we are getting, if you’re not getting an output then we are going to say it is a VOID.

Method Name:

* The Methodname should start with lowercase and should be followed by camel case.

(NOTE🡪method name should be in a meaning full way)

🡪Get: the 1st part is the action.

🡪Text: the 2nd part is on whom you’re performing the action.

Graphical user interface, text, application

Description automatically generated Note: Move to element

(args): ARGS means the inputs, or sometimes we may not give the inputs.

* Input—Action🡪 Choco -- eat. /Both input and action
* Action 🡪 switch off the FAN. /Only action

Args can be multiple or one.

Depends on the requirements U may pass the args or may not pass the args.

Syntax Description:

Methods🡪access this method-> return you something or not return something🡪method name🡪 inputs that your giving may not giving.

* In a method return statement should be the last line of the code if not that becomes unreachable code

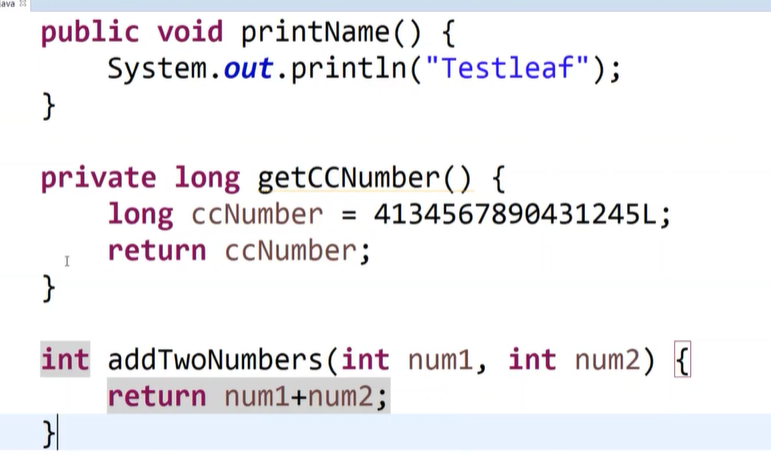
Graphical user interface, text

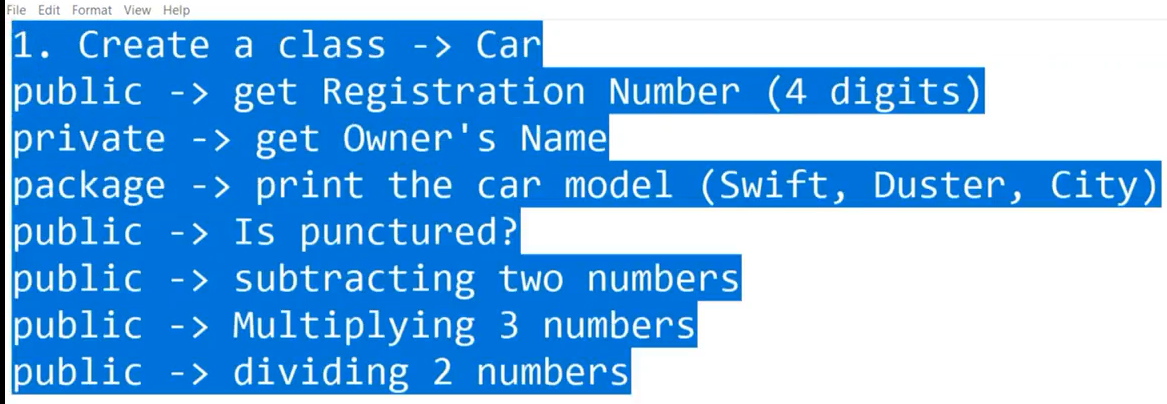
Description automatically generated

Code:

Graphical user interface, text, application

Description automatically generated





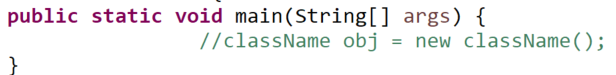
OBJECT

how to call a method or variable -> will be under reference of a class >outside the class will not have any codes. methods and variables will be in the class.

Object: Representative member of the class, which will access all the methods and variables in the class.

Accessing Syntax for OBJECT:

className obj = new className ();



Obj can be renamed as our usage.

Ex: className met = new className ()

Obj=**met** have been replaced as the Obj here.

Accessing the obj over the methods and variables:

Syntax: 🡪 Obj.methodname();

When we want to store the return output of a method then you can use this shortcut: crt+2+l

Task to print the data type string 