```cpp
#include <iostream>
using namespace std;

int main()
{
   //! Variables -->
     //? are containers for storing data values
     //?  to use it later on in the program in different places

    //! Data types -->
       //^ Different types of variables (defined with different keywords)
        // 1) Fundamentals(primitive data type)
        // 2) Drived
        // 3) User defined

    //! Fundamentals: (6)
            //%1)Integral Type:
               //? a) int
                  //(numbers that do not contain decimal numbers) keyword--> ((int))
               //? b) char
                  //(Used to store only one character and the character is placed inside '' ) keyword--> ((char))
               //?# note -->  The character is within the integral group because each character has a value in the ASCII code table
            //% 2)Floating Type:
               //? a)float
                  //(numbers that  contain decimal numbers) keyword--> ((float))
               //? b)double
                  //(numbers that  contain decimal numbers) keyword--> ((double))
               //# note --> the size of double is bigger than the size of the float and it use within the biggest program
            //% 3)Boolean Type:
               //? bool
                  //(It stores a single value (true or false)) keyword--> ((bool))
                  //$ 0 mean false , 1 mean true
                  //^ any number is true but the 0 is false
            //% 4)String Type:
               //? String
                  //is an array of character (the characters / text put inside " ") keyword--> ((String))
            //% 5)void Type:
               //? void
                  //it is an empty type (dont return any value)--> ((void))
            //% 6)wide character Type
               //? wide character
                  //(Used to store only one character and the
                  //character is placed inside '' for unicode characters) keyword--> ((wchar_t))

     //! Drived:(4)
            //% 1)Function
            //% 2)Array
            //% 3)Pointer
            //% 4)Reference

```

```
    //! User-Defined:(5)
        //% 1) Class
        //% 2) Structure
        //% 3) union
        //% 4) enum
        //% 5) Type Def

    //! Declaring variables (Creating)
        //? To create a variable , specify the type & assign it a value
        //^ Syntax--> dataType identifier = value ;
        //~ The identifier is case sensitive
        //$ note --> you can also declare a variable without assigning the value
        //$ and assign the value later
        //? dataType identifier ;
        //? identifier = value ;

    //! note:
       //* 1) if you assign a new value to an existing variable , it will overwrite the previous value
       //* 2) the cout object is used together with  the (<<) operator to display variables
            //* to combine both text & a variable , separate them with the << operator

    //! Declare many variables in same data type  -->
       //$ to declare more than one variable of the same datatype use a comma to separated list
       //~ ex --> \tdataType identifier1 , identifier2 , identifier3 = value , identifierN ,.... ;

    //!note:
     //? if the value i dont want to use it later  on in the program
     //? so get the value or expression and use it directly without storing it in variable

    //! Naming variables --> (identifiers) the name variable
        //# dataType identifier = value (literal constant);
            //~ All c++ variables must be identified with unique names
            //~ these unique names are called identifiers
            //~ identifiers can be short names (like x & y) or more descriptive names (sum , age)

    //! important note -->
       //% it ie recommended to use ((((( descriptive names ))))) 
       //%  in order to create understandable & maintainable code

    //! Naming Rules :-
        //? the general rules for naming variables are -->
          // 1) names can contain letters , digits & underscores
          // 2) names are case sensitive
          // 3) names must begin with a letter or an underscore (_)
          // 4) names can not contain whitespace or special character
          // 5) reserved words (keywords) can not be used as names

    //! Constants (Read Only)
        //? to declare constants -->
           //const dataType identifier = value (literal constant);
               //~ when you dont want others (or yourself) to override existing variables values
               //~ use the const keywords (this will declare the variables as "constant")
               //# which means unchangeable and read only
               //# you should always declare the variable as constant
               //# when you have values that are unlikely to change
}
```