C++ Note 3:

Now we move to the next topic under the concepts of OOP i.e. Constructors and Destructors. They sound deadly but are very easy.

So we have learned about functions defined inside a class which we also call as member functions. If you remember we say functions/behaviours are actions that can either use the properties/arguments that the class holds or change the values of the properties.

So let us say we don't give any initializing value to the properties and don't write any function that can give it any values but declare an object. What happens then?

Then CONSTRUCTOR happens.

The Constructor is a special member function of every class that sets an initializing value to all the properties that are declared inside a class. Some key characteristics of the constructor are:

1. The constructor is of the same name as that of the class
2. It has no return type
3. It is always public

So know we know what a constructor is but when is it called? Whenever we declare an object of a class the constructor is called to give the object default values for the characteristics that are inside it.

You may have noticed previously that whenever we were writing classes or creating objects we didn't write any function that resembled a constructor but still the program ran and we got our desired output. This could be done because whenever we do not declare a constructor explicitly the C++ compiler makes a constructor by itself and gives it some default values.

In real life system design, not giving a constructor is considered a bad programming practice, but in your standard of studying, its Okay!.

Here is an example on how to declare a constructor:
https://drive.google.com/file/d/15a2fDrB8D_NMj411u-v1kxyo1Ao70S4o/view?usp=sharing

There are several constructors depending on various characteristics that they possess:

a. Default Constructor: A constructor which has no argument is known as default constructor. It is invoked at the time of creating object. So the constructor we defined in the above example is a default constructor. When C++ automatically defines a constructor it is a default constructor.

b. Parameterized Constructor: A constructor which has parameters is called parameterized constructor. Different objects will have different values so as to cater to that particular need we should be able to give specific values to the objects that we define.
The declaration of parameterized constructor is as follows:
https://drive.google.com/file/d/1NI4c1RDUuKmIfbWxUxxTeY8maBBKMDzt/view?usp=sharing

c. Overloaded Constructor: We have previously seen function overloading while studying polymorphism. Overloading Constructors are similar to overloading functions just keeping in in the different special characteristics of constructors. Let us see an example and understand this:
https://drive.google.com/file/d/1wLrpTg6SFRfEJsKN20szr1tD9bR74JKx/view?usp=sharing

d. Copy Constructor : We know constructors initialize the attributes of a class on creation of an object but with some distinct values. But in copy constructors we use other objects of the same class to initialize any new object that is being created.
Let us say you post a photo that has a caption “Aesthetic Morning” and a song let’s say “Antariksh by Anuv Jain”. So this photo/post becomes an object of the class InstaPost. Now if you want to post another photo, which is also an object of the same class, and you want that photo to have the same caption and music you do not necessarily need to pass the values again, rather you can use the previous post/object you created to initialize this post.
Lets see this in a program:
https://drive.google.com/file/d/1i69DXx6wLv0U2S8brPV9VCpMNua77rGw/view?usp=sharing

e. Constructor with default arguments: A constructor that holds default values for its parameters is known as a constructor with default arguments. The calling of a constructor with default arguments can be done with either one argument or no argument at all.

ASSIGNMENT: Write a C++ program that includes all the different types of constructors and also write a display function to display the values of the data members you use.

DESTRUCTOR:

Destructors are the “Destroyers of objects” that is it destructs the objects in a class, which is an exact opposite of what a constructor does. It defined same as a constructor that is it has the same name as the class it is defined in and also has no return type. But a difference between a constructor and destructor is that a destructor is always declared using the “~” sign in front of it.

Another special characteristic of a destructor is that it has no parameters thus it can be only declared once inside a class and there can be no overloading of a destructor.

They are called automatically when the class object goes out of scope such as when the function ends, the program ends, a delete variable is called etc.

For example: https://drive.google.com/file/d/1l56G6hJjSQMExUbYs48Ztb_w-nm7sf_U/view?usp=sharing

~ -> Tilde