Why OOPs?

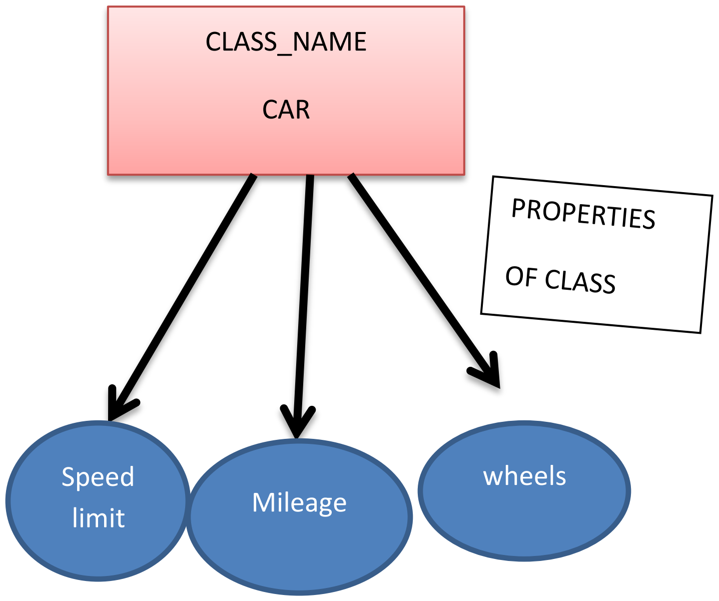
Object-oriented programming:-

- As the name suggests , it uses the objects.
- .The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

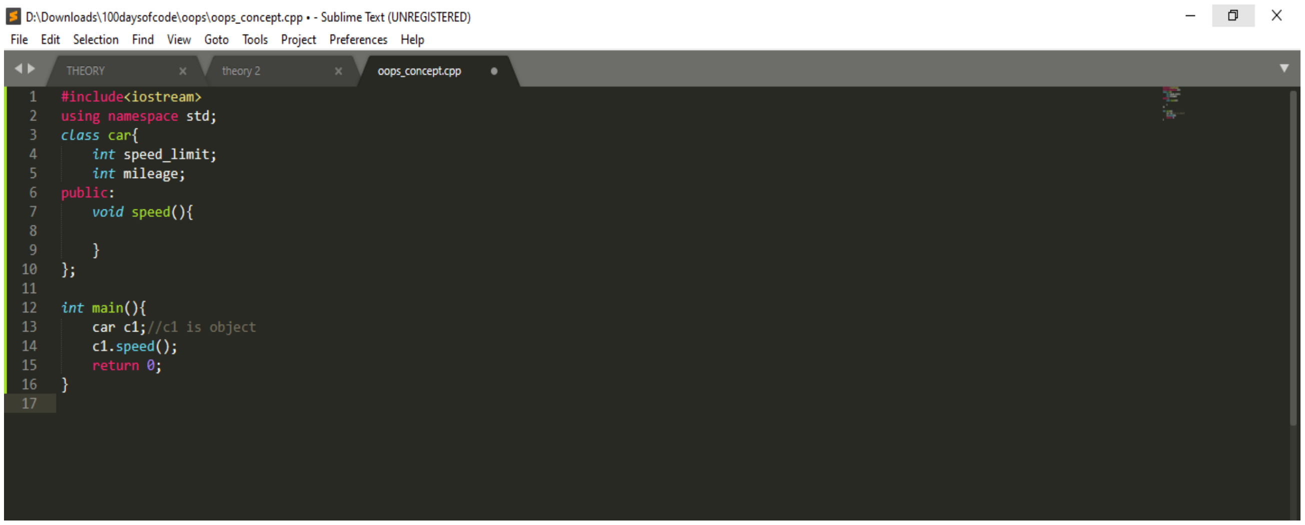
Building block of oops:-CLASS

CLASS:-A CLASS IS A USER-DEFINED DATA TYPE, WHICH HOLDS ITS OWN DATA MEMBERS AND FUNCIONS.

Lets take example of class:-



An Object is an identifiable entity with some characteristics and behaviour. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

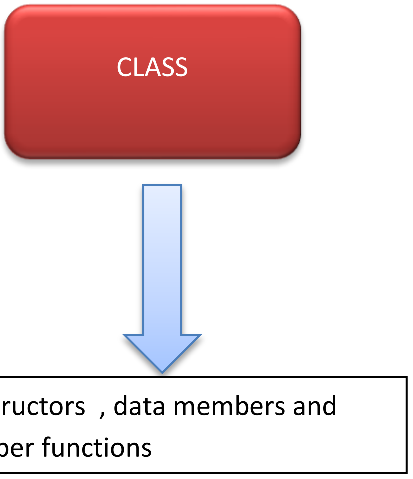

D:\Downloads\100daysofcode\oops\oops_concept.cpp • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
THEORY
theory 2
oops_concept.cpp

```cpp
#include<iostream>
using namespace std;
class car{
    int speed_limit;
    int mileage;
public:
    void speed(){

    }
};

int main(){
    car c1;//c1 is object
    c1.speed();
    return 0;
}
```

CONSTRUCTORS AND DESTRUCTORS:-

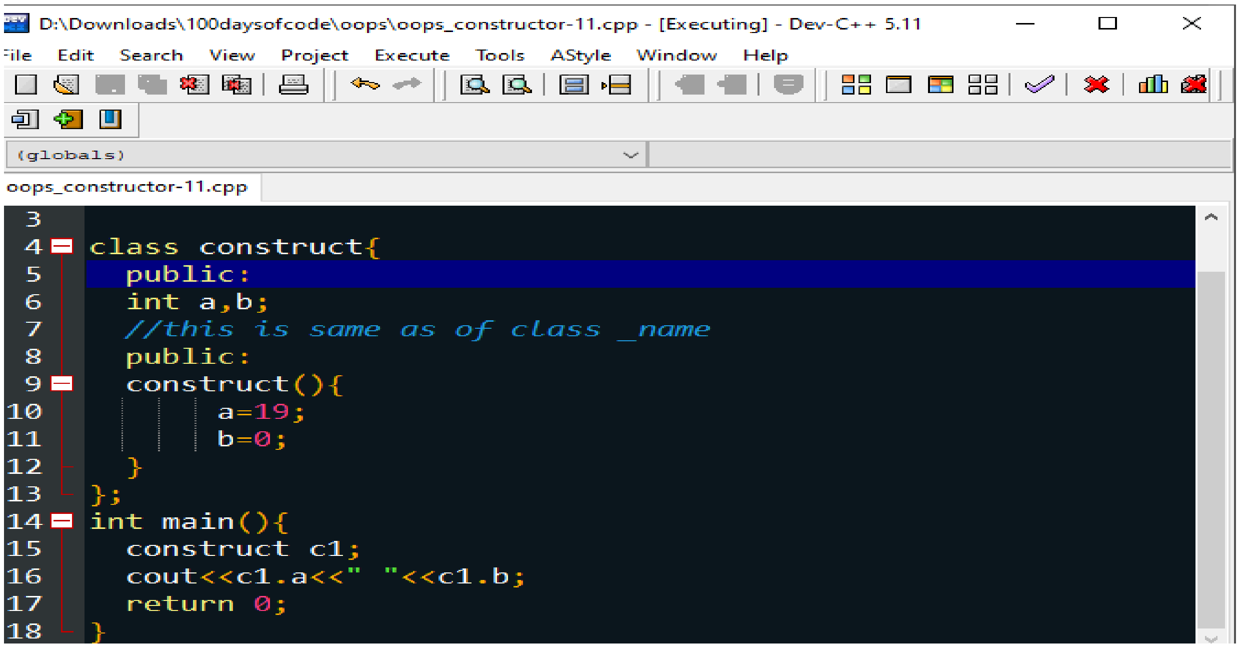
CONCEPT OF CONSTRUCTORS AND DESTRUCTORS

‘



HOW CONSTRUCTOR IS DIFFERENT FROM NORMAL MEMBER FUNCTION???

- **Constructor has same name as the class itself**
- **Constructors don’t have return type**
- **A constructor is automatically called when an object is created.**
- **If we do not specify a constructor, C++ compiler generates a default constructor for us (expects no parameters and has an empty body).**


D:\Downloads\100daysofcode\oops\oops_constructor-11.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
oops_constructor-11.cpp

```
class construct{
  public:
  int a,b;
  //this is same as of class _name
  public:
  construct(){
      a=19;
      b=0;
  }
};
int main(){
  construct c1;
  cout<<c1.a<<" "<<c1.b;
  return 0;
}
```

IMPORTANT TIPS!!!!

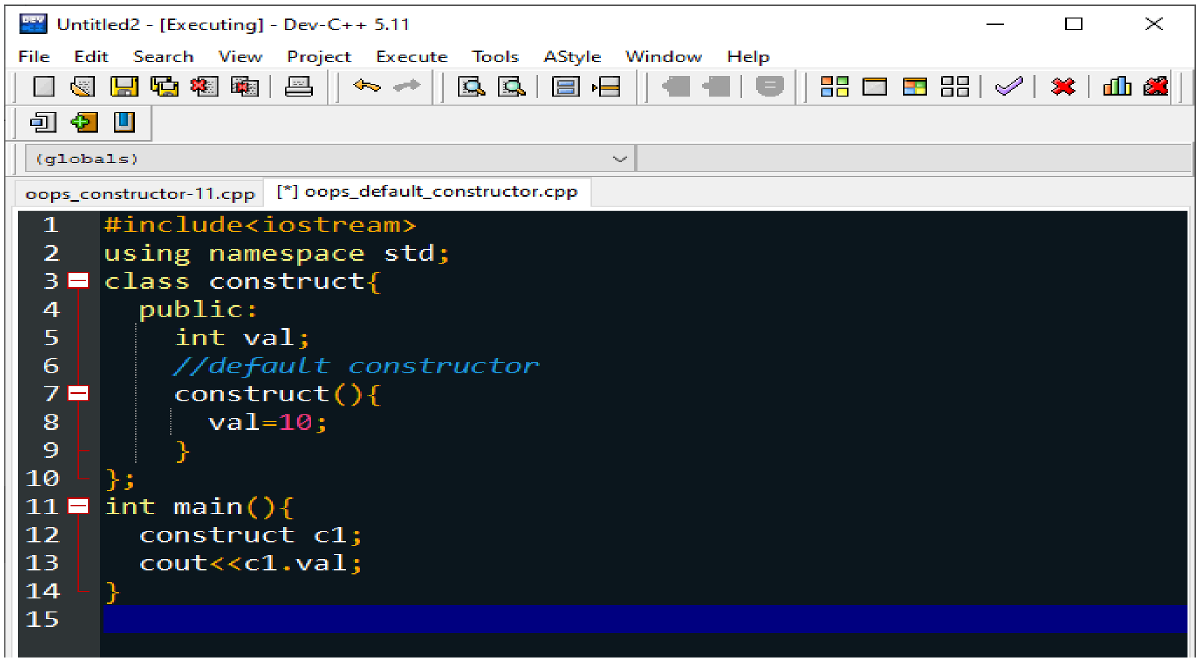
1. They should be declared in the public section
2. They do not have any return type, not even void
3. They get automatically invoked when the objects are created
4. They cannot be inherited though derived class can call the base class constructor
5. Like other functions, they can have default arguments
6. You cannot refer to their address
7. Constructors cannot be virtual

TYPES OF CONSTRUCTORS::->>

- **DO-NOTHING CONSTRUCTOR**
- **DEFAULT CONSTRUCTOR**
- **PARAMETERIZED CONSTRUCTOR**
- **COPY CONSTRUCTOR**

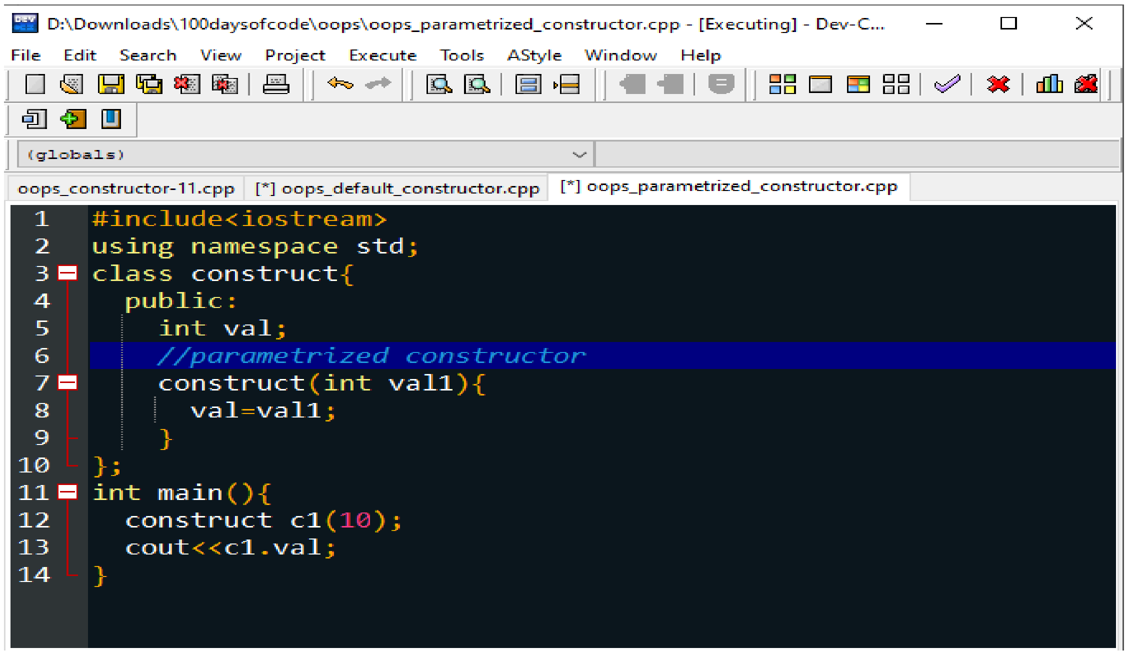
Do nothing constructors are that type of constructor which does not contain any statements. Do nothing constructor is the one which has no argument in it and no return type.

The default constructor is the constructor which doesn't take any argument. It has no parameter but a programmer can write some initialization statement there.


Untitled2 - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
oops_constructor-11.cpp
[*] oops_default_constructor.cpp

```
#include<iostream>
using namespace std;
class construct{
  public:
    int val;
    //default constructor
    construct(){
      val=10;
    }
};
int main(){
  construct c1;
  cout<<c1.val;
}
```

A default constructor does not have any parameter, but programmers can add and use parameters within a constructor if required. This helps programmers to assign initial values to an object at the time of creation


D:\Downloads\100daysofcode\oops\oops_parametrized_constructor.cpp - [Executing] - Dev-C...
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
oops_constructor-11.cpp
[*] oops_default_constructor.cpp
[*] oops_parametrized_constructor.cpp

```cpp
#include<iostream>
using namespace std;
class construct{
  public:
    int val;
    //parametrized constructor
    construct(int val1){
      val=val1;
    }
};
int main(){
  construct c1(10);
  cout<<c1.val;
}
```

COPY CONSTRUCTOR:-C++ provides a special type of constructor which takes an object as an argument and is used to copy values of data members of one object into another object. In this case, copy constructors are used to declaring and initializing an object from another object