

Class and Object

VISHAL V MEHTRE¹, NIKHIL NALE²

¹ Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India

² Department of Electrical Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India

Abstract -- In computer programming, the object class refers to a class created to group various objects which are instances of that class. Classes are code templates for creating objects. In cases where objects need to be grouped in a certain way, an object class is the "container" for a set of objects built on these templates

I. INTRODUCTION

Classes provide a way of packing together data and functions which work on the data. In Java terminology, this data items are known as fields and functions are known as methods. A class is a blueprint from which we can create objects of the same type.

Class:

The building block of C++ that ends up in Object orienting programming may be a category. it's a user outlined information sort, that holds its own information members and member functions, which may be accessed And employed by making an instance of that category. a category is sort of a blueprint for AN object.[2]

For Example: Consider the Class of Cars. There may be many cars with different names and brand but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range etc. So here, Car is the class and wheels, speed limits, mileage are their properties: [2]

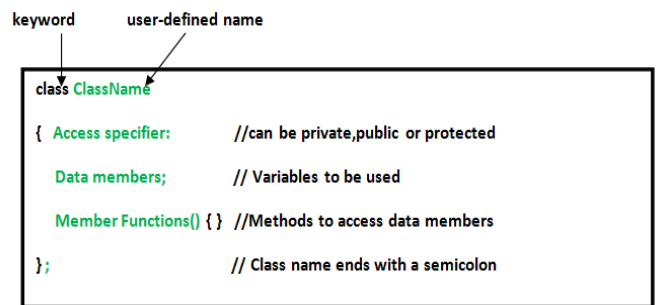
A Class may be a user outlined knowledge-type that has data members and member functions. Data members ar {the knowledge the info the information} variables and member functions ar the functions accustomed manipulate these variables and along these data members and member functions defines the properties and behavior of the objects during a category.

In the on top of example of sophistication automobile, the info member are regulation, mileage etc and member functions is apply brakes, increase speed etc.

An Object is Associate in Nursing instance of a category. once a category is outlined, no memory is allotted however once it's instantiated (i.e. Associate in Nursing object is created) memory is allotted. Object ar basic run time entities in Associate in Nursing object directed system. Programming program ar analysis in terms of objects and nature of communication between them.[1]

II. DEFINING CLASS AND DECLARING OBJECTS

A class is defined in C++ using keyword class followed by the name of class. The body of class is defined inside the curly brackets and terminated by a semicolon at the end.



Declaring Objects: Once a category is outlined, solely the specification for the thing is defined; no memory or storage is allotted. To use the info and access functions outlined within the category, you would like to make objects.

Syntax:

Class Name Object Name;

Accessing information members and member functions The data members and member functions of sophistication will be accessed exploitation the dot('.') operator with the thing. for instance if the name of object is obj and you would like to access the member perform with the name printName() then you'll have to be compelled to write obj.printName().[3]

Accessing information Members

The public information members also are accessed within the same manner given but the non-public information members don't seem to be allowed to be accessed directly by the thing. Accessing {a information|a knowledge|an information} member depends only on the access management of that data member.

This access management is given by Access modifiers in C++. There square measure 3 access modifiers : public, non-public and guarded.

Public:

All the category members declared below public are on the market to everybody. the info members and member functions declared public will be accessed by alternative categories too. the general public members of a category will be accessed from anyplace within the program exploitation the direct member access operator (.) with the thing of that category.[1]

Private:

The class members declared as non-public will be accessed solely by the functions within the category. they're not allowed to be accessed directly by any object or perform outside the category. solely the member functions or the friend functions square measure allowed to access the non-public information members of a class[1].

Protected:

Protected access modifier is analogous thereto of personal access modifiers, the distinction is that the category member declared as Protected square measure inaccessible outside the category however

they will be accessed by any subclass(derived class) of that class[1].

Member Functions in categories

There square measure two ways that to outline a member function:

- Inside category definition
- Outside category definition

To outline amember perform outside {the category|the category} definition we've to use the scope resolution:: operator in conjunction with class name and performance name.

EXAMPLE:

```
#include <iostream.h>

class Geeks
{
    // Access specifier
    private:

    // Data Members
    string geekname;

    public:
    // Member Functions()
    void printname()
    {
        cout << "Geekname is: " << geekname;
    }
};

int main() {

    // Declare an object of class geeks
    Geeks obj1;

    // accessing data member
    obj1.geekname = "Abhi";

    // accessing member function
    obj1.printname();
    return 0;
}
```

OUTPUT:
Geekname is: Abhi

III. CONCLUSION

This lesson has shown you the benefits that OOP brings to a project and to a programmer. You have learned how to:

1. Keep your analysis in the business world, using business words, with encapsulation of related strings and numbers into classes
2. Make teamwork more productive by separating classes that require specialized knowledge from those that do not.
3. Achieve reuse without instruction lists from the programmer who wrote the code you are reusing.
4. Compare the responsibilities of your classes with the overall project scope to ensure nothing's been forgotten.

The next module discusses classes and objects in more detail and illustrates how to create class and object diagrams.

ACKNOWLEDGEMENT

We would like to express our special thanks of gratefulness to **Dr. D.S. Bankar**, Head, Department of electrical engineering for able Guidance and support for completing the research paper. I would like to thank the faculty member of the department of electrical engineering who helped us with extended support

REFERENCES

- [1] "E Balgurusamy Object Oriented Programming with C++"Mc graw hill Education(2016)
- [2] "Sumit Arora Object Oriented Programming with C++" Dhanpat rai and C.O. (2016)
- [3] "Object oriented thought process Matt weisfield" Addison Wesley (2017)
- [4] "A brief history of the mean value theorem" Besenyei, A (September 17, 2012).
- [5] "Extended Mean-Value Theorem". W. Weisstein, Eric Retrieved 2018-10-08.
- [6] "Prof. Danny, Object Oriented Programming with C++" Green Hills Publication(2013)