Syntax To Define A Class In C++

```
Class className{
Access_specifier:
Data members;
Member functions()
}
```

Here,

- **Class:** The keyword class is used to define a class in C++
- **Access_specifier**: It is an access controller which is used to control the visibility of the data. There are three specifiers- private, public, and protected. We have discussed these in the section ahead.
- **Data members**: These are variables that hold the data related to the object of the class.
- **Member functions:** These are methods that help perform varied operations on data members and specify the behavior of the object.

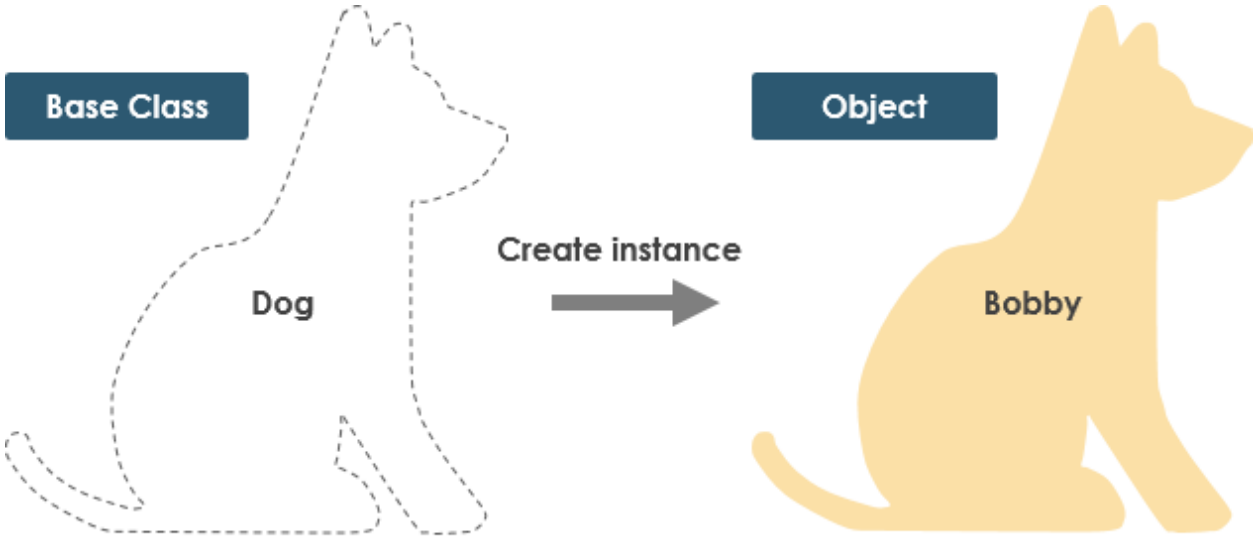
Syntax Of Object In C++

```
Class_name Object_name;
```

Here,

- **Class_name**: It is the name of the class for which the object is created

- **Object_name**: It is the name of the object. Note that having a meaningful object name enhances code readability



Properties	Methods
Color	Sit
Eye Color	Lay Down
Height	Shake
Length	Come
Weight	

Property Values	Methods
Color: Yellow	Sit
Eye Color: Brown	Lay Down
Height: 17 in	Shake
Length: 35 in	Come
Weight: 24 pounds	

Example:

Object:student
Data:

Rollno
Name
Grade
Marks

Function:
Total()
Average()
Display()
Calaculate()

Sample Program

```
#include <iostream>
using namespace std;
class Car {
public:
// Data members
string make;
string model;
int year;
// Method to display car information
void displayInfo() {
cout << "Make: " << make << ", Model: " << model << ",
Year: " << year <<endl;}
};
```

```
int main() {
Car myCar;
// Assigning values to the data members of the object
myCar.make = "Toyota";
myCar.model = "Camry";
myCar.year = 2023;
// Calling the method of the object to display car information
myCar.displayInfo();
return 0;
}
```

Note: Access specifier defines how the members of the class can be accessed. In C++, there are 3 types of access specifiers: public, private, and protected.

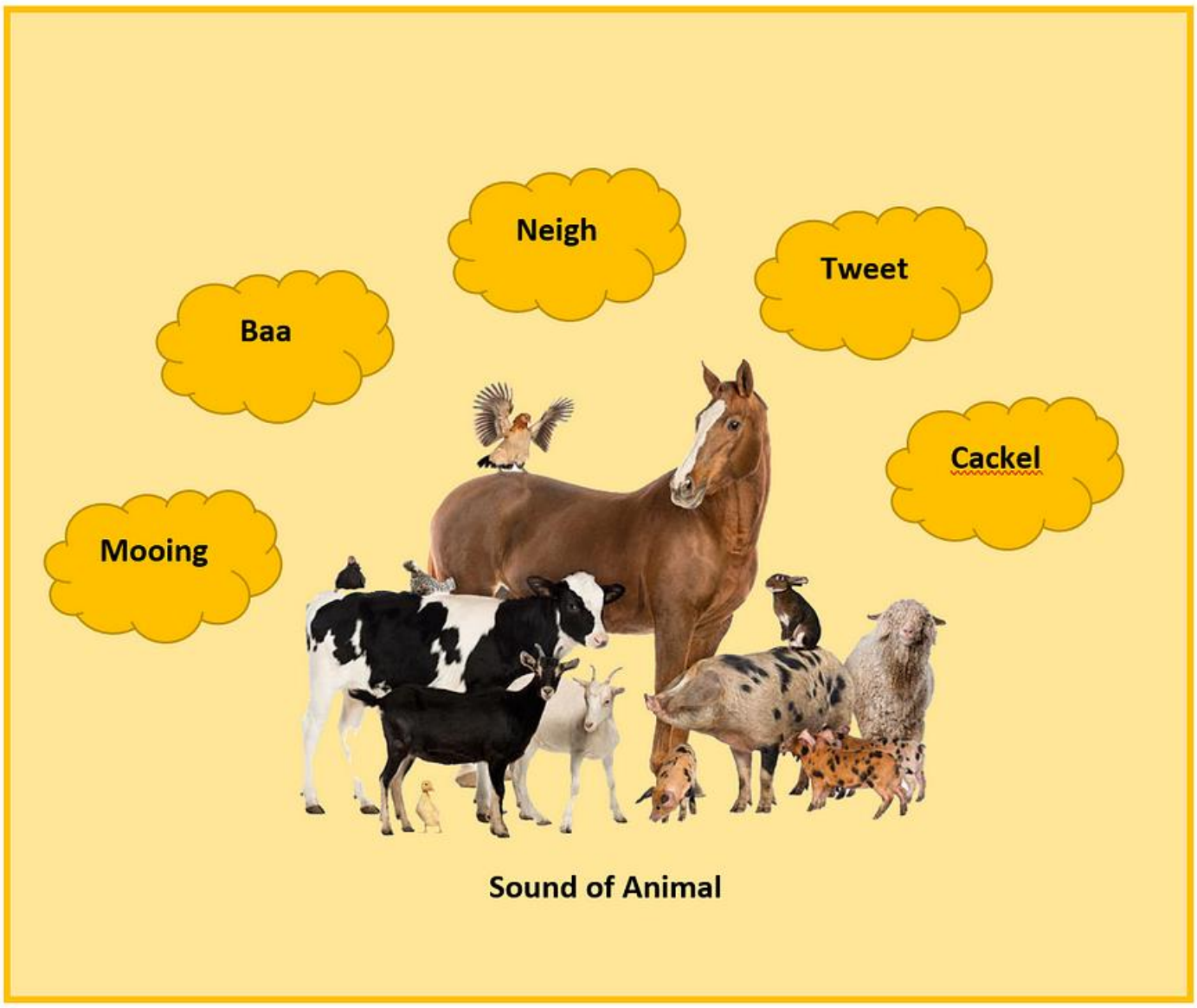
- **public:** members can be accessed outside the class.
- **private:** members cannot be accessed outside the class.
- **protected:** members cannot be accessed(viewed) from outside the class, but can be accessed in inherited classes(subclasses).

1. Polymorphism

Polymorphism is the ability to exist in many forms.

Example 1: We can take a boy as a real-world example. This boy can be a student, a player, and a writer. So that this boy can exist in different ways in different situations.

Example 2: Sound of animals. People have the same sound but different animals make different sounds. The following diagram shows few different sounds make by animals.



This what we called as polymorphism.

2. Inheritance

Inheritance means it allows classes to inherit common properties from the parent class.

Example 1: let's assume that there is a class as Vehicle. All vehicles are not the same. We can inherit common properties like color, size, type from the parent vehicle class and create classes like Car, Bus, Bike.

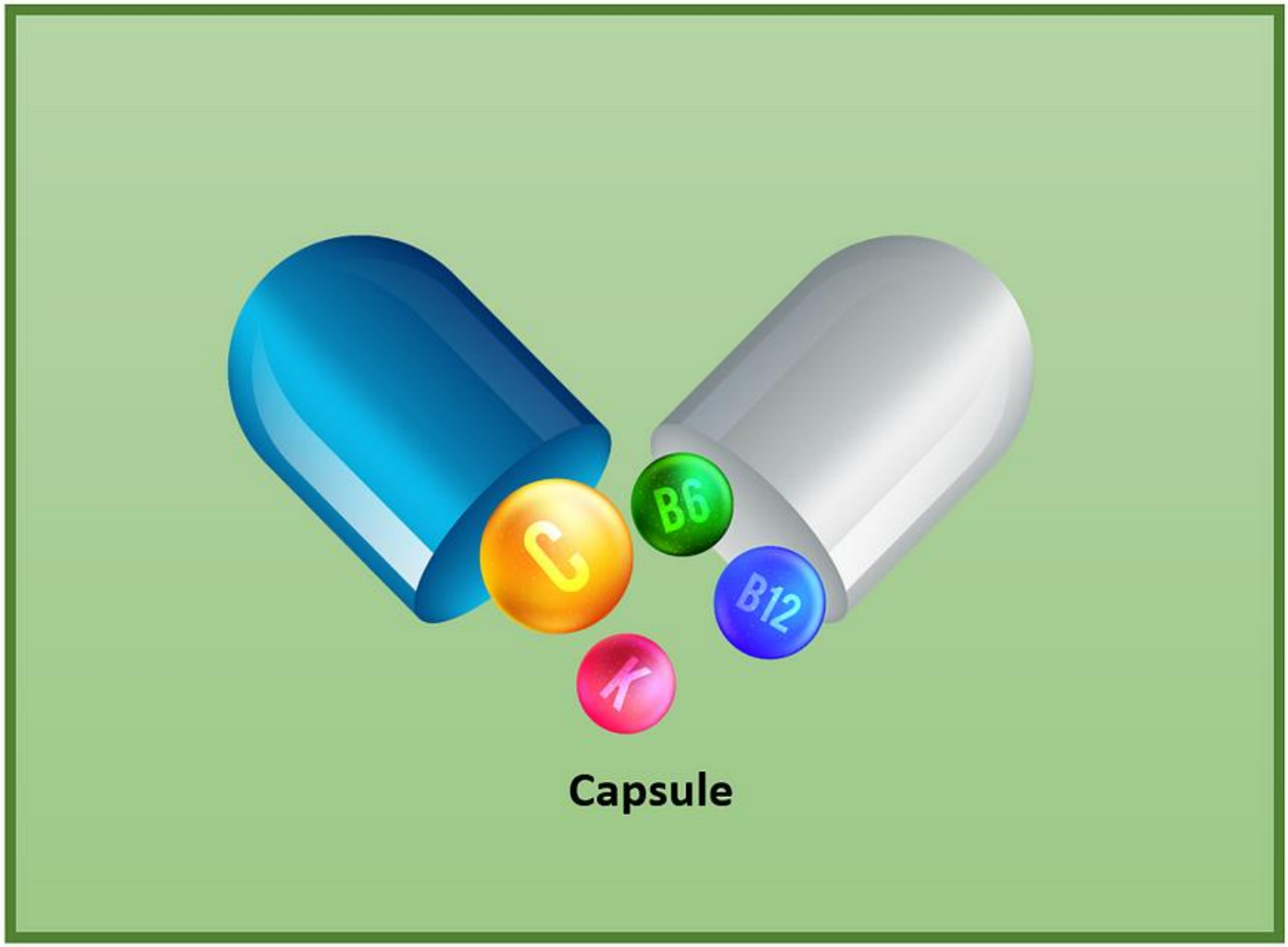
Example 2: let's take another parent class as Animals. Here also we can inherit common properties like name, sound, color, breed from Animal class and create classes like Dog, Cat, Horse and etc.

Animals

3. Encapsulation

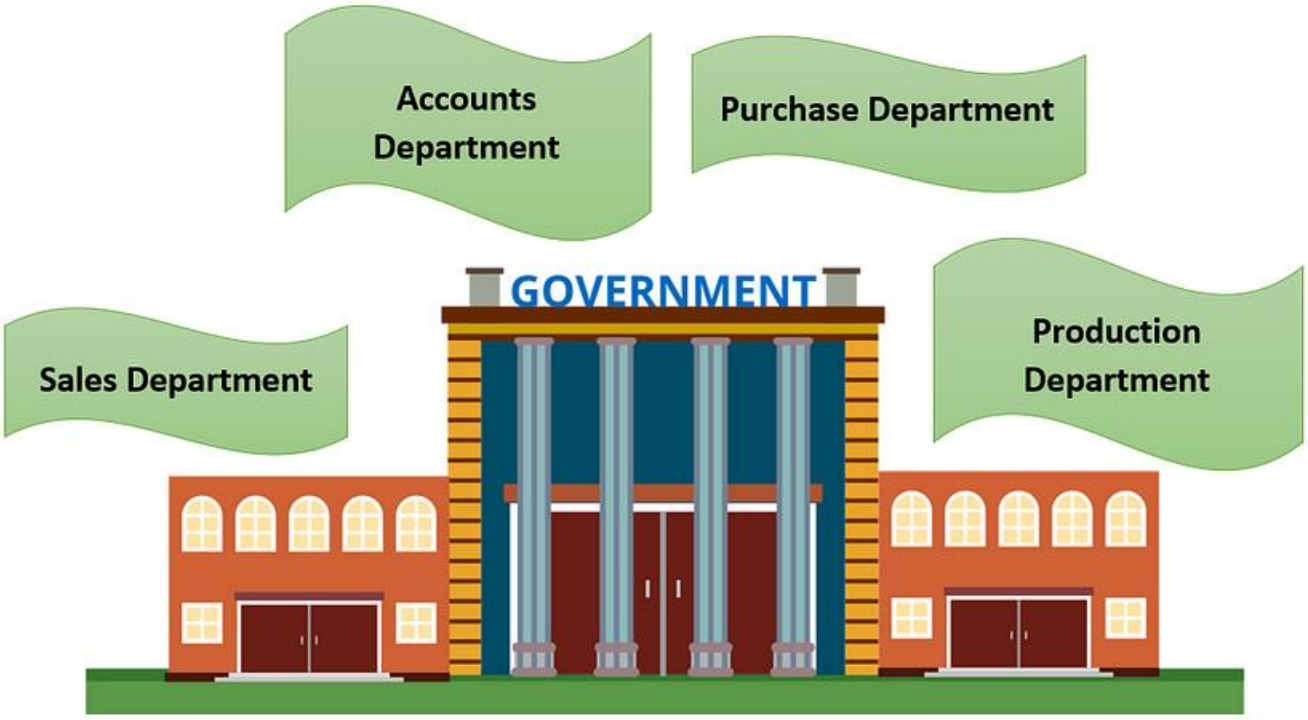
Encapsulation means it binds data and code together into one unit.

Example1; the most commonly used example is the medical capsule. This capsule mixes few types of medicines and stored in one capsule.



Example 2: another example for encapsulation is a large organization. An organization is consists of several departments

like the production department, purchase department, sales department, and Accounts department. It combines all these departments together and had formed the organization.



This is what we called as encapsulation.

4. Abstraction

In abstraction, it displays only the important information by hiding the implementation part.

Example 1: Let's take the ATM machine. In an ATM machine, we can perform functions like withdraw cash, deposit cash, check balance, print bills, and so on. Even though it performs a lot of actions it doesn’t show us the process. It has hidden its process by showing only the main things like getting inputs and giving the output.

Example 2: The next example is the most commonly used mobile phones. On a mobile phone, we can perform so many actions like making a call, sending messages, take pictures, download software and etc. We perform a lot of things but here

also we don't know the inside process of these things. Which means the implementation parts are hidden.