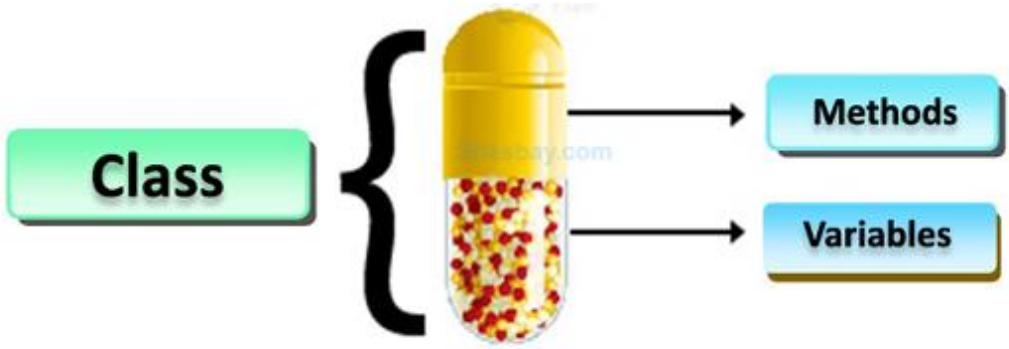
Encapsulation in C++

Encapsulation is a process of combining data members and functions in a single unit called class. This is to prevent the access to the data directly, the access to them is provided through the functions of the class. It is one of the popular feature of Object Oriented Programming(OOPs) that helps in **data hiding**.



Example

In a company, two software projects are going on with two different teams. One team requires data from the other team. But this team cannot access the data from the other team because they do not have the appropriate permissions. This is *Encapsulation*.

How Encapsulation is achieved in a class

1) Make all the data members private.
2) Create public setter and getter functions for each data member in such a way that the set function set the value of data member and get function get the value of data member.

Advantage of Encapsulation in C++

1. The main advantage of using of encapsulation is to secure the data from other methods, when we make a data private then these data only use within the class, but these data not accessible outside the class.
2. The major benefit of data encapsulation is the security of the data. It protects the data from unauthorized users that we inferred from the above stated real-real problem.
3. We can apply the concept of data encapsulation in the marketing and finance sector where there is a high demand for security and restricted access of data to various departments.
4. Encapsulation helps us in binding the data(instance variables) and the member functions(that work on the instance variables) of a class.
5. Encapsulation is also useful in hiding the data(instance variables) of a class from an illegal direct access.
6. Encapsulation also helps us to make a flexible code which is easy to change and maintain.

Example

main.cpp

```cpp
// Encapsulations using classes
#include<iostream>
using namespace std;
class ExampleEncap{
private:
   /* Since we have marked these data members private,
    they have to use getter and setter functions. */
   int num;
   char ch;
public:
   /* Getter functions to get the value of data members.
   thus provide the access to data members through them */
   int getNum()  {
      return num;
   }
   char getCh()  {
      return ch;
   }
   /* Setter functions, they are called for assigning the values
    * to the private data members.
    */
   void setNum(int num) {
      this->num = num;
   }
   void setCh(char ch) {
      this->ch = ch;
   }
};
```

```
29 int main(){
30    ExampleEncap obj;
31    obj.setNum(100);
32    obj.setCh('A');
33    cout<<obj.getNum()<<endl;
34    cout<<obj.getCh()<<endl;
35    return 0;
36 }
```

```
100
A
```

So we hide the data from external environment

Difference Between Data Abstraction and Encapsulation in C++

Data Abstraction	Data Encapsulation
It solves the issue by proposing a design.	It solves the issues by implementing the design proposed by data abstraction.
Helps in hiding the background details and explanation of how the data is derived.	Helps in hiding the data by combining it with functions and objects as a single unity to protect it from unauthorized users.
Shifts your focus on what the object does instead of what it does.	Encapsulation helps you hide the internal detail from the outside world by giving it restricted access.
They are abstract in nature.	They are known as ADT (abstract data types) as they are used to create objects of its own type.