Abstraction & Encapsulation in C++

Abstraction

- ***Data abstraction*** is one of the most essential and important feature of object oriented programming in C++.
- Abstraction means displaying only ***essential information*** and ***hiding the details***.
- Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Abstraction Using Class

```cpp
#include <iostream>
using namespace std;
class Abstract
{
    private:
        int a, b;
    public:
        /// method to set values of private members
        void set(int x, int y)
        {
            a = x;
            b = y;
        }
        void display()
        {
            cout<<"a = " <<a << endl;
            cout<<"b = " << b << endl;
        }
};
int main()
{
    Abstract obj;
    obj.set(10, 20);
    obj.display();
    return 0;
}
```

Abstraction

- You can see in the above program we are not allowed to access the variables a and b directly, however one can call the function set() to set the values in a and b and the function display() to display the values of a and b.

Advantages of Data Abstraction:

- Helps the user to avoid writing the low level code
- Avoids code duplication and increases reusability.
- Can change internal implementation of class independently without affecting the user.
- Helps to increase security of an application or program as only important details are provided to the user.

Abstraction in Header File

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
 int n = 4;
   int power = 3;
   int result = pow(n,power);        // pow(n,power) is the  power function
   std::cout << "Cube of n is : " <<result<< std::endl;
   return 0;
}
```

Ways of Abstraction in C++

- **Abstraction using Classes:** We can implement Abstraction in C++ using classes. Class helps us to group data members and member functions using available access specifiers. A Class can decide which data member will be visible to outside world and which is not.
- **Abstraction in Header files:** One more type of abstraction in C++ can be header files. For example, consider the pow() method present in math.h header file. Whenever we need to calculate power of a number, we simply call the function pow() present in the math.h header file and pass the numbers as arguments without knowing the underlying algorithm according to which the function is actually calculating power of numbers.

Ways of Abstraction in C++

- **Abstraction using access specifiers**

✓ Access specifiers are the main pillar of implementing abstraction in C++. We can use access specifiers to enforce restrictions on class members. For example:

✓ Members declared as **public** in a class, can be accessed from anywhere in the program.

✓ Members declared as **private** in a class, can be accessed only from within the class. They are not allowed to be accessed from any part of code outside the class.

Advantages of Data Abstraction

- Helps the user to avoid writing the low level code.
- Avoids code duplication and increases reusability.
- Can change internal implementation of class independently without affecting the user.
- Helps to increase security of an application or program as only important details are provided to the user.

Encapsulation in OOP

Encapsulation

- In normal terms ***Encapsulation is defined as wrapping up of data and information under a single unit.***
- In Object Oriented Programming, ***Encapsulation is defined as binding together the data and the functions*** that manipulates them.
- Encapsulation also lead to data abstraction or hiding. As using encapsulation also hides the data.

Real-Life Example of Encapsulation

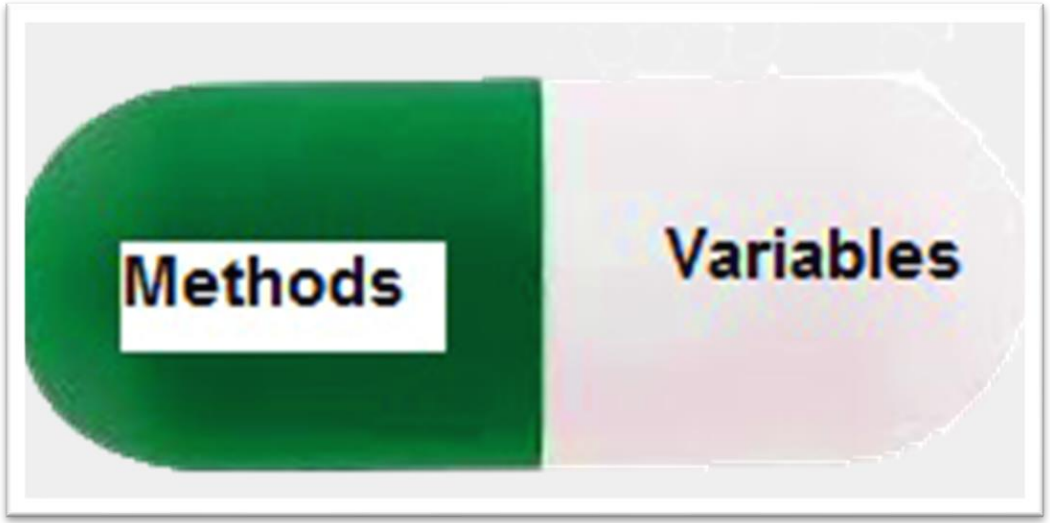
- ***School Bag***

Real-Life Example of Encapsulation

➢ *Classroom/Room*

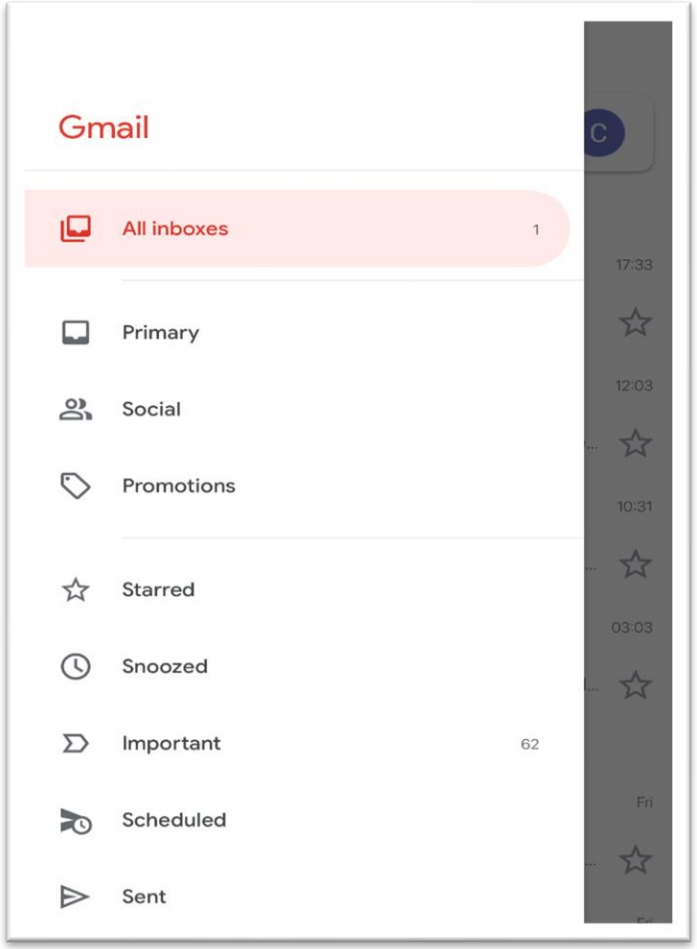
Real-Life Example of Encapsulation

➢ *Capsule*



Real-Life Example of Encapsulation

➢ *Gmail Account*



Real-Life Example of Encapsulation

- ***Bank Account***

Encapsulation

- In normal terms **Encapsulation** is defined as wrapping up of data and information under a single unit. In Object Oriented Programming,
- Encapsulation is defined as binding together the data and the functions that manipulates them.
- Encapsulation also lead to data abstraction or hiding. As using encapsulation also hides the data.

Encapsulation

```
#include<iostream>
using namespace std;
class Encapsulation
{
    private:
        /// data hidden from outside world
        int x;
    public:
        /// function to set value of variable x
        void set(int a)
        {
            x = a;
        }
        /// function to return value of variable x
        int get()
        {
            return x;
        }
};
int main()
{
    Encapsulation obj;
    obj.set(5);
    cout<<obj.get() << endl;
    return 0;
}
```

Encapsulation

- In the above program the variable **x** is made private. This variable can be accessed and manipulated only using the functions get() and set() which are present inside the class. Thus we can say that here, the variable x and the functions get() and set() are binded together which is nothing but encapsulation.

Role of access specifiers in encapsulation:

- As we have seen in above example, access specifiers plays an important role in implementing encapsulation in C++. The process of implementing encapsulation can be sub-divided into two steps:
  - ✓ The data members should be labeled as private using the **private** access specifiers
  - ✓ The member function which manipulates the data members should be labeled as public using the **public** access specifier

Difference Between Abstraction & Encapsulation

ABSTRACTION	ENCAPSULATION
Abstraction is the process or method of gaining the information.	While encapsulation is the process or method to contain the information.
In abstraction, problems are solved at the design or interface level.	While in encapsulation, problems are solved at the implementation level.
Abstraction is the method of hiding the unwanted information.	Whereas encapsulation is a method to hide the data in a single entity or unit along with a method to protect information from outside.
We can implement abstraction using abstract class and interfaces.	Whereas encapsulation can be implemented using by access modifier i.e. private, protected and public.
In abstraction, implementation complexities are hidden using abstract classes and interfaces.	While in encapsulation, the data is hidden using methods of getters and setters.
The objects that help to perform abstraction are encapsulated.	Whereas the objects that result in encapsulation need not be abstracted.

Q: Define a class “**PRIME**” in C++ with the following description:
Private Data Members:
IdNo(integer), Name(string), Result(double)

Public Member Functions:
A **constructor** to assign initial values IdNo as 12, Name as “XYZ”, and Result as 3.90.
Input() to take the input for IdNo, Name and Result.
Display() to display all the data members.

Thank You