Abstraction in C++

Data 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.

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.

```
class Abstract
{
        private:
                int a, b;
        public:
                void set(int x, int y)
                {
                        a = x;
                        b = y;
                }

                void display()
                {
                        cout<<"a = " <<a << endl;
                        cout<<"b = " << b << endl;
                }
};
```

```
void main()
{
        Abstract a1;
        a1.set(10, 20);
        a1.display();
        getch();
}
```

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.