Encapsulation

- Encapsulation refers to bundling data and the methods that operate on that data into a single unit.
- Many programming languages use encapsulation frequently in the form of classes.
- A class is an example of encapsulation in computer science in that it consists of data and methods that have been bundled into a single unit.
- Encapsulation may also refer to a mechanism of restricting the direct access to some components of an object, such that users cannot access state values for all of the variables of a particular object.
- Encapsulation can be used to hide both data members and data functions or methods associated with an instantiated class or object.
- In other words: Encapsulation is about wrapping data and methods into a single class and protecting it from outside intervention.
- The general idea of this mechanism is simple.
- For example, you have an attribute that is not visible from the outside of an object.
- You bundle it with methods that provide read or write access.
- Encapsulation allows you to hide specific information and control access to the object’s internal state.

Example:

```
#include <iostream>
using namespace std;
class Student {
    // private data members
    private:
    string
    studentName; int
    studentRollno; int
    studentAge;
    // get method for student name to access
    // private variable studentName
    public:
        string
            getStudentName() {
            return studentName;
        }
    // set method for student name to set
    // the value in private variable studentName
    void setStudentName(string
        studentName) { this -> studentName
        = studentName;
    }
    // get method for student rollno to access
    // private variable studentRollno
    int getStudentRollno() {
        return studentRollno;
    }
    // set method for student rollno to set
    // the value in private variable studentRollno
    void setStudentRollno(int
        studentRollno) { this ->
        studentRollno = studentRollno;
```

```
    }
    // get method for student age to access
    // private variable studentAge
    int getStudentAge() {
        return studentAge;
    }
    // set method for student age to set
    // the value in private variable studentAge
    void setStudentAge(int studentAge)
        { this -> studentAge =
        studentAge;
    }
};
int main() {
    Student obj;
    // setting the values of the variables
    obj.setStudentName("Avinash");
    obj.setStudentRollno(101);
    obj.setStudentAge(22);
    // printing the values of the variables
    cout << "Student Name : " << obj.getStudentName() << endl;
    cout << "Student Rollno : " << obj.getStudentRollno() << endl;
    cout << "Student Age : " << obj.getStudentAge();
}

Output:
Student Name : Avinash
Student Rollno : 101
Student Age : 22
```