object-oriented programming

(OOP)

KIAN _ ACADEMY

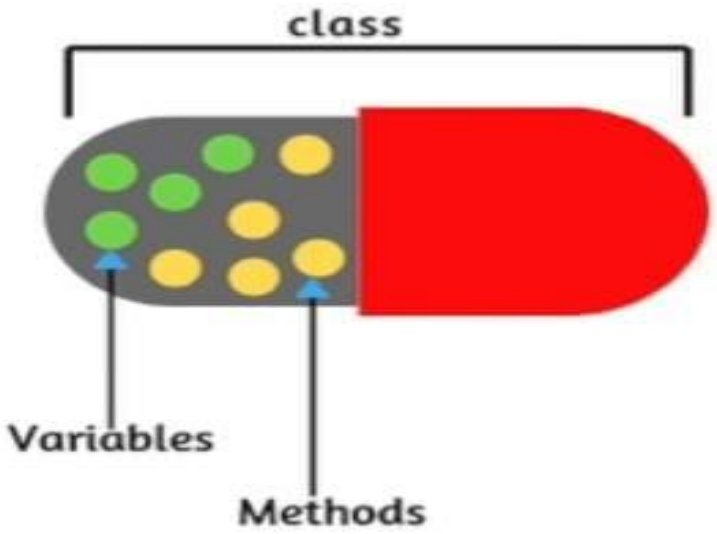
Encapsulation :-

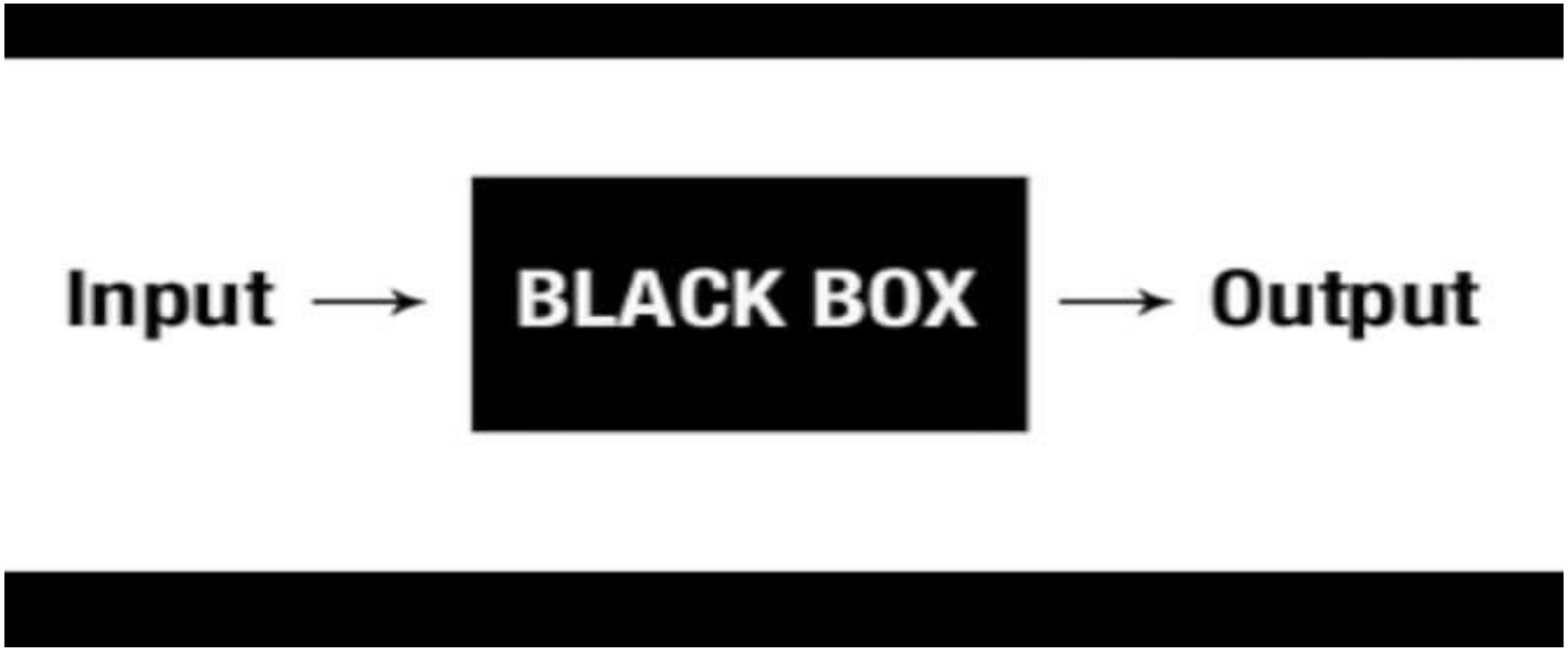
- In Object-Oriented Programming, **Encapsulation** is defined as binding together the data and the functions that operate on those data members into a single class .

```
class
{

  data members
        +
  methods (behavior)

 }
```




Input
BLACK BOX
Output

Example :-

```
class Rectangle {
  public:
    int length;
    int width;
    int getArea() {
      return length * width;
    }
};
```

- using encapsulation also hides(wrapping up) the data
- So can we define **Encapsulation** as : Hiding implementation details of an object from its clients.

Separating Class Code into 2 files :-

1_Header File - .h

- Contains the declaration of all the class members.
- Only attributes declaration and methods prototypes

2-Implementation File - .cpp

- Contains the implementation of the class methods

client code

- is the one that includes the main function. This file should be stored by the name main.cpp

Mutators and Accessors :-

Mutator(setter)

- function that store a value in a private member variable , or changes its value in some way

Accessor(getter)

- function that return a value from a private member variable

Note Setter And Getter should be define as public.

Example :-

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

```
// main function
int main()
{
    Encapsulation obj;
    obj.set(5);  // x=5;
    cout<<obj.get();
    return 0;
 }
```

output:

5

Exam questions:

- **(1) Which among the following best describes encapsulation?**

a) It is a way of combining various data members into a single unit

b) It is a way of combining various member functions into a single unit

c) It is a way of combining various data members and member functions into a single unit which can operate on any data

d) It is a way of combining various data members and member functions that operate on those data members into a single unit

- **(2) If data members are private, what can we do to access them from the class object?**

a) Create public member functions to access those data members

b) Create private member functions to access those data members

c) Create protected member functions to access those data members

d) Private data members can never be accessed from outside the class

- **(3) Which feature can be implemented using encapsulation?**

a) Inheritance

b) Abstraction

c) Polymorphism

d) Overloading

- **(4) which of the following uses encapsulation?**

a) void main(){ int a ; void fun(int a=10 ; cout<<a) ; fun() ; }

b) class student{ int a ; public : int b ; } ;

c) class student{ int a ; public : void disp(){ cout<<a ; } } ;

d) struct topper{ char name[10] ; public : int marks ; }

- **(5) How can Encapsulation be achieved?**

a) Using Access Specifiers

b) Using only private members

c) Using inheritance

d) Using Abstraction

- **(6) Which among the following violates the principle of encapsulation almost always?**

a) Local variables

b) Global variables

c) Public variables

d) Array variables

- **(7) Using encapsulation data security is…….**

a) Not ensured

b) Ensured to some extent

c) Purely ensured

d) Very low

- **(8) Encapsulation is the way to add functions in a user defined structure**.

a) True

b) False

Answer the questions :-

- 1 - d
- 2 - a
- 3 - b
- 4 - c
- 5 - a
- 6 - b
- 7 - b
- 8 - b