Classes and Objects

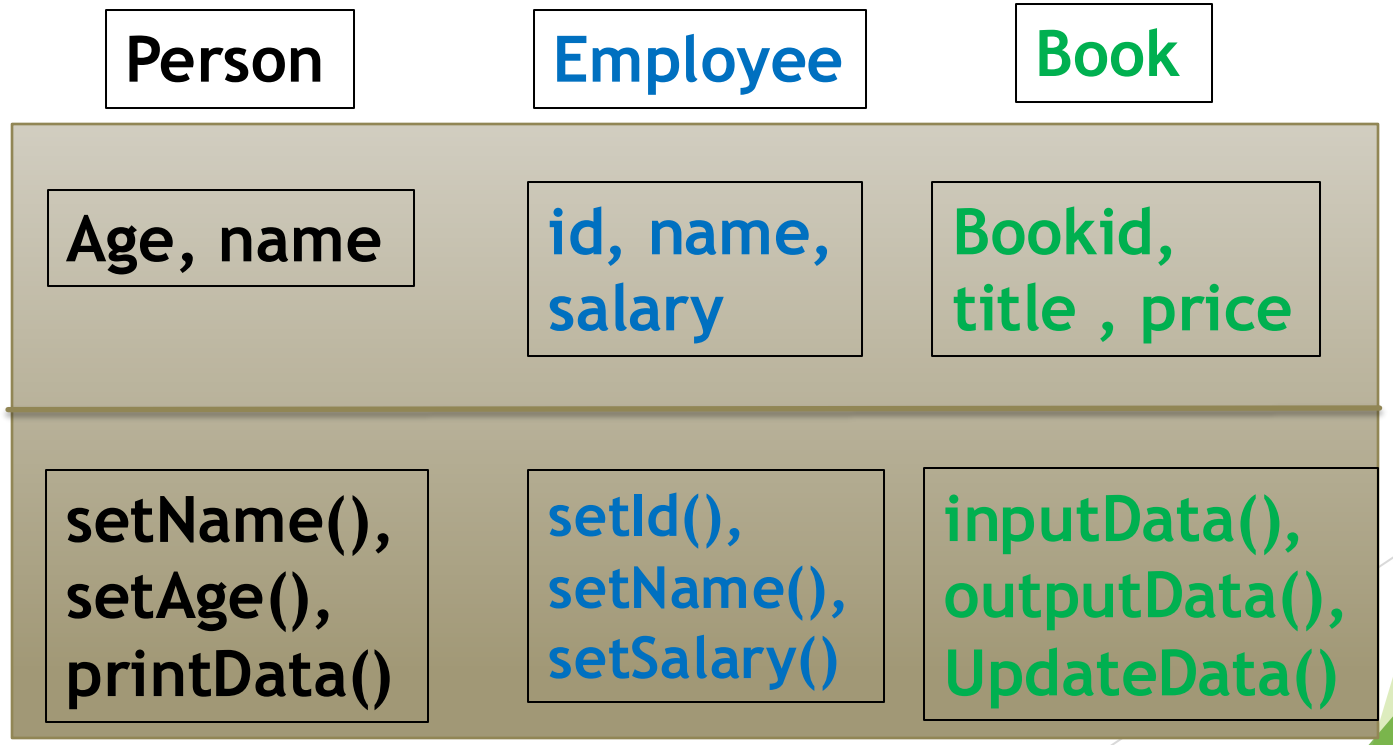
Mohammad Tasin

Agenda

- Encapsulation
- Implementation of Encapsulation
- Class
- Object
- Example

Encapsulation

- An act of combining properties and methods related to the same entity is known as Encapsulation.



- **Properties :-**
  - **Data members**
  - **Fields**
  - **Attributes**
  - **member variables**
- **Methods :-**
  - **function**
  - **Procedure**
  - **action**
  - **service**

How to implementation of Encapsulation in C++

- There are two ways :- 👇

1. struct
2. class

- By default members of a structure are public and members of a class are private

class

- class is a keyword to create a custom data type. (just like struct)
- class is a group of a variables, functions and operators
- class is a description of an object
- class is a common noun
- class provides a blueprint for its objects
- class is a way to implement concept of encapsulation

Syntax of class

```
class ClassName
{
    //Variables
    //Functions
    //Operators
};
```

object

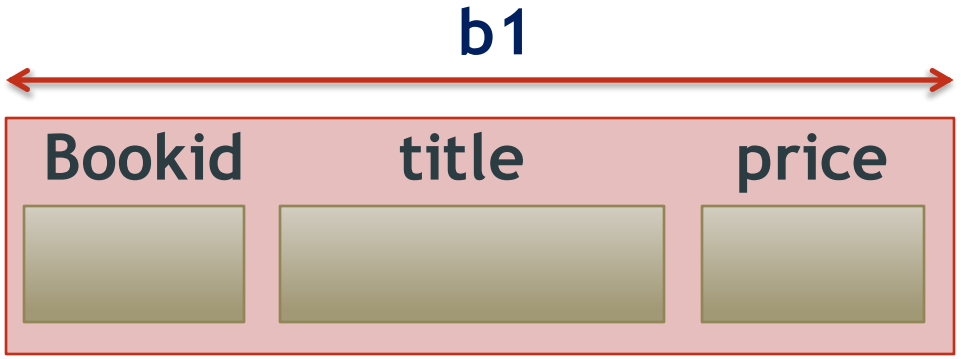
- Object is a real world entity
- Object is an instance of a class
- object is a proper noun
- object has a state and behaviaur

State → object state is a set of values at particular instant.

Behaviaur → Behaviaur of an object is set of actions it can perform

Book → class

Book b1; → b1 is an object



b1.inputData()

b1.outputData()

b1.UpdateData()

- **state of an object should be changed only via its methods.**

Example

```
#include<iostream>
#include<string.h>
using namespace std;
class Book
{
    private:
        int Bookid;
        float Price;
        char Name[20];
    public:
        void setId(int id) {    Bookid = id; }
        void setPrice(float pri)  { Price = pri;  }
        void setName(char *naam)  { strcpy(Name, naam); }
        void ShowData()  {
            cout<<Bookid<<endl;
            cout<<Price<<endl;
            cout<<Name<<endl;
        }
};
```

```
int main()
{
    Book b1;
    b1.setId(1);
    b1.setPrice(450.0F);
    b1.setName("Python");
    b1.ShowData();
    return 0;
}
```