Objects and Classes

Object Oriented Programming

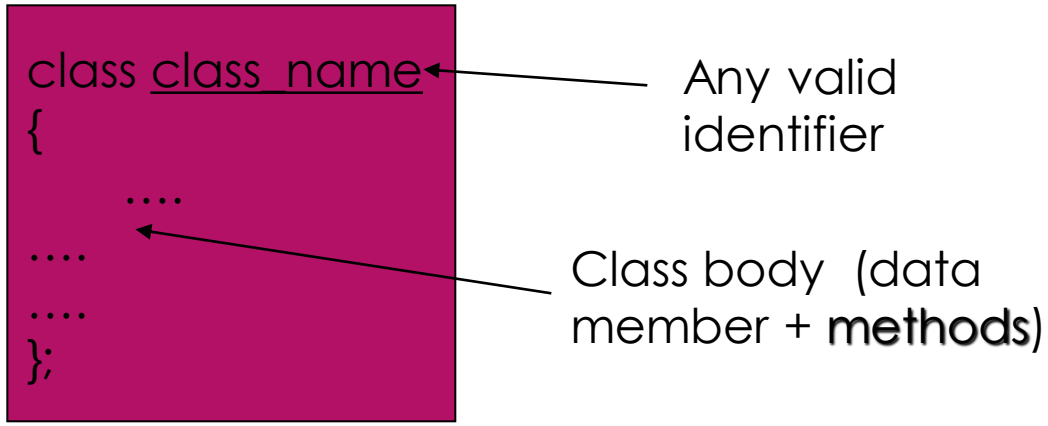
- Object-oriented programming (OOP)
  - Encapsulates data (attributes) and functions (behavior) into packages called classes.
- So, Classes are user-defined (programmer-defined) types.
  - Data (data members)
  - Functions (member functions or methods)
- In other words, they are structures + functions

Classes and objects in C++

- **Class:** A class in C++ is the building block, that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.
  For Example: Consider the Class of **Cars**. There may be many cars with different names and brand but all of them will share some common properties like all of them will have *4 wheels*, *Speed Limit*, *Mileage range* etc. So here, Car is the class and wheels, speed limits, mileage are their properties.
- A Class is a user defined data-type which has data members and member functions.
- Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions defines the properties and behavior of the objects in a Class.
- In the above example of class *Car*, the data member will be *speed limit*, *mileage* etc and member functions can be *apply brakes*, *increase speed* etc.
- An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Classes in C++

- A class definition begins with the keyword *class*.
- The body of the class is contained within a set of braces, { } ; (notice the semi-colon).

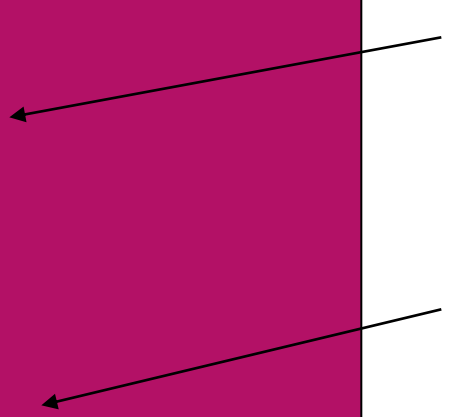


Classes in C++

- Within the body, the keywords *private:* and *public:* specify the access level of the members of the class.
  - the default is private.
- Usually, the data members of a class are declared in the *private:* section of the class and the member functions are in *public:* section.
- Don't confuse data hiding with the security techniques used to protect computer databases

Classes in C++

```
class class_name
{
    private:
      …
      …
      …
    public:
      …
      …
      …
};
```



private members or methods

Public members or methods

Lets start……

Demonstrates a small, simple object

```
#include <iostream>
class smallobj
{
private:
      int somedata;
public:
  void setdata(int d)
  {     somedata = d;
  }
  void showdata()
  {  cout << "Data is : " << somedata << endl;
  }};
int main()
{  smallobj s1, s2;
  s1.setdata(1106);
  s2.setdata(1425);
  s1.showdata();
  s2.showdata();
  return 0;
     }
```

Data members

Member Functions

Calling Member Functions

Output of the Program:

Data is : 1106
Data is : 1425

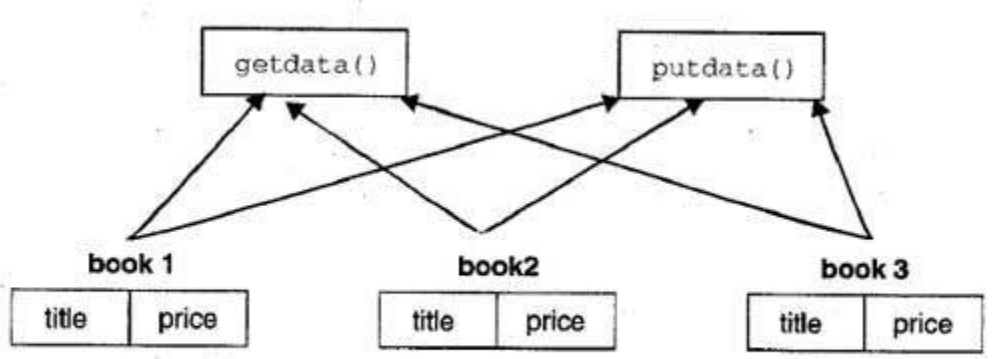
Classes and Objects

- An object has the same relationship to a class that a variable has to a data type
- An object is said to be an instance of a class
  - Chevrolet is instance of a vehicle
  - In smallobj example, s1, s2 are instances of smallobj class

Memory Allocation: Book class example

```
class Book
{
private:
char title[20];
float price;
public:
void getdata(){.....}
void putdata(){.....}
};

int main(){
Book book1,book2,book3;
book1.getdata();
book2.getdata();
....
book1.putdata();
book2.pudata();
....
}
```



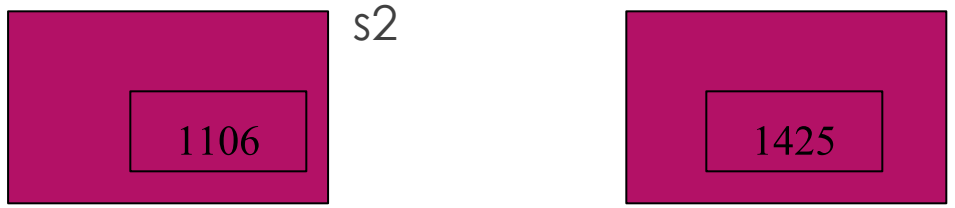
Memory Allocation for the Objects of the Class book

Functions are public, Data is private

- Usually the data within a class is private and functions are public
- Data is hidden so it will be safe from accidental manipulation
- Functions that operate on the data are public so they can be accessed from outside the class
- However, there is no rule that says data must be private and functions must be public

Objects

Objects of class smallobj




```
smallobj s1, s2;
  s1.setdata(1106);
  s2.setdata(1425);
```

```
// Objpart.cpp

#include <iostream.h>
using namespace std;
class part
{
private:
        int modelnumber;
         int partnumber;
        float cost;
public:

            void setpart(int mn, int pn, float c)
             {
                modelnumber = mn;
                partnumber = pn;
                cost = c;
             }

             void showpart()
             {
             cout << "Model: " << modelnumber << endl;
             cout << "Part: " << partnumber << endl;
             cout << "Cost: " << cost << endl;
             }
};
```

```
int main()
{
            part part1;

          part1.setpart(6244,329,55
          );
            part1.showpart();

            return 0;
}
```