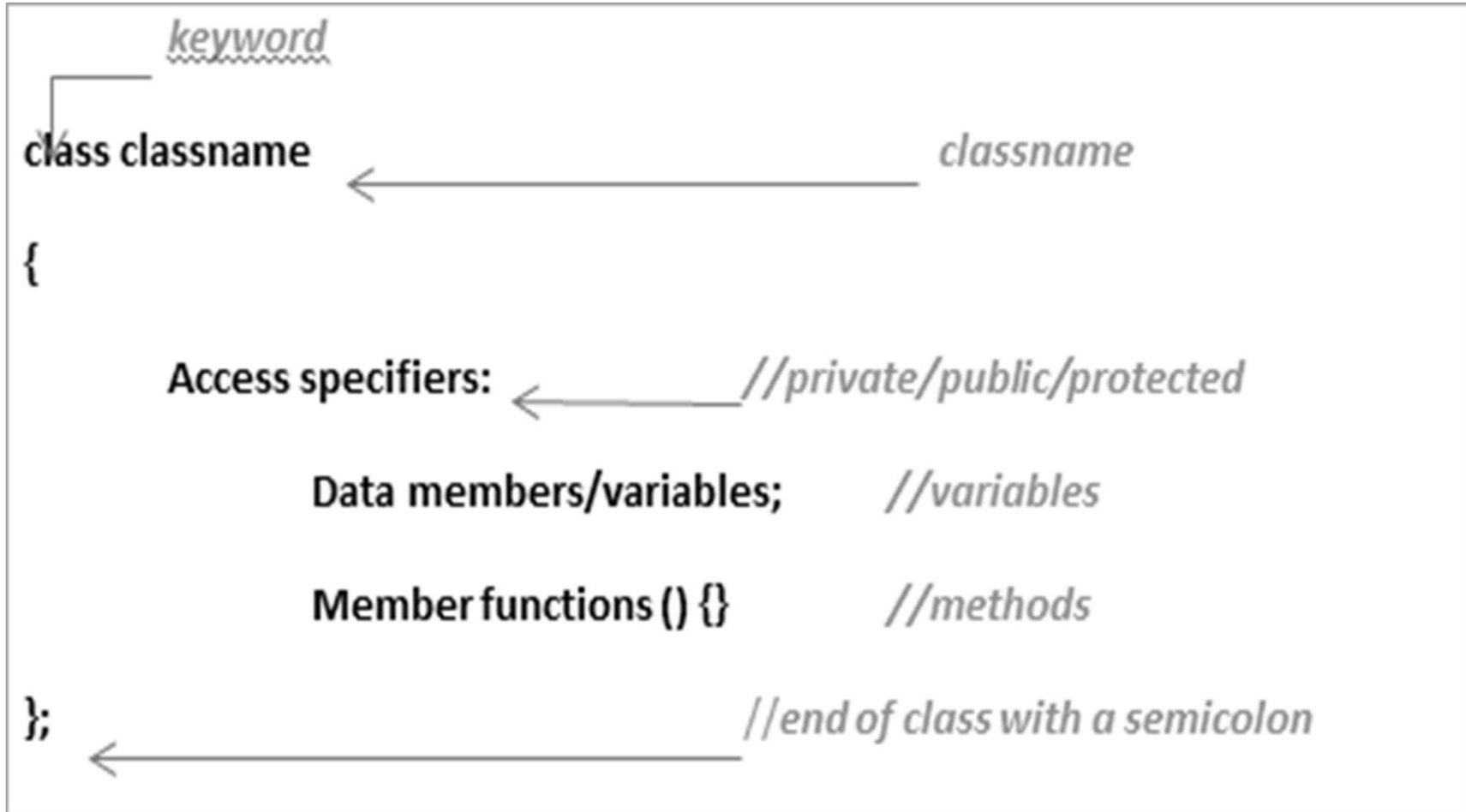


Object Oriented Programming (OOP)

Class Declaration:

Declaration of class must start with the keyword *class* followed by



```
#include<iostream>
using namespace std;
class student
{
private :
    int id;
    char name[20];
public :
    Void Getdata(void);
    Void display (void)
    {
        cout << id <<'\t' << name << endl;
    }
};
int main( )
{
```



Data Members



**Member
Functions**

Data Member And Member Function

Data Members

- The variables declared inside the class are known as **data members**.
- The private data of a class can be accessed only through the member functions of that class.
- Data members Can be of any type, built-in or user-defined

Member Function

- The functions declared inside the class are known as **member functions**.
- Member functions access the values of the data members and perform operations on the data members .
- Their definition can be placed inside the class body, or outside the class body.
- Can access both public and private members of the class

```
#include<iostream>
using namespace std;
class student
```

```
{
```

```
private :
```

```
int id;
```

```
char name[20];
```



Data Members

```
public :
```

```
Void Getdata(void);
```

```
Void display (void)
```

```
{
```

```
cout << id << '\t' << name << endl;
```

```
}
```



**Member
Functions**

```
};
```

```
int main( )
```

```
{
```

Defining Objects

- Defining an object is similar to defining a variable of any data type
- When an object of a class is created, the space for all data members defined in the class is allocated in the memory according to their data types.

Classes & Objects

```
class Rectangle
{
  private:
    int width;
    int length;
  public:
    void set(int w, int l);
    int area();
}
```

```
Rectangle r1;
Rectangle r2;
Rectangle r3;
```

⋮

```
int a;
```



Define C++ Object

- Using Reference variable
- Syntax for creating an object of class

Class-Name Object-Name;

- Example
- Following statements declare two objects of class Student

```
Student Std1;  
Student Std2;
```

Classes & Objects

```
class Box
{
  private:
    // data members
  public:
    // member functions
};
```

← **Class**

Box b1

Box b2

Box b3

← **Objects**

Accessing Data Members and Member functions

- Data members and member functions can be accessed in similar way the member of structure is accessed using member operator(.).
- **Syntax for accessing the member function of class**

Object_Name . Function_Name;

- **Syntax for accessing the Data member of class**

Object_Name . Data_Memeber

Note : We cannot access the data member of the class in other class if the data members are private or protected.

Program Example

```
class Student{
    public:
    void show(){
        cout<< "Hello Students";
    }
};

int main(){
    // object created with std name
    Student std;
    object.show();
}
```

Creating an Object
of Class Student

Calling show Function
using Object of Class
Student

```
class Student
{
    public:
    int rollno;
    string name;
};

int main()
{
    Student A;
    Student B;

    // setting values for A object
    A.rollno=1;
    A.name="Adam";

    // setting values for B object
    B.rollno=2;
    B.name="Bella";

    cout <<"Name and Roll no of A is: " << A.name << "-" << A.rollno;
    cout <<"Name and Roll no of B is: " << B.name << "-" << B.rollno;
}
```

