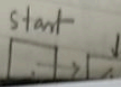
Sorted Linked list :

If some element inserted at the proper
list then this type of linked list is kn
order linked list.

There will be two cases for adding th
i) list is empty or element value is les
first node.
ii) The element value lies in between or

i)



Date

even by mistake.
There are mechanisms to access even private data using friends, pointer to member etc from outside the class.

Access specifiers.

Access specifier	Accessible to	
	own class members	Object of a class.
private :	Yes	No
protected :	Yes	No
public :	Yes	Yes.

fig: Visibility of class members.

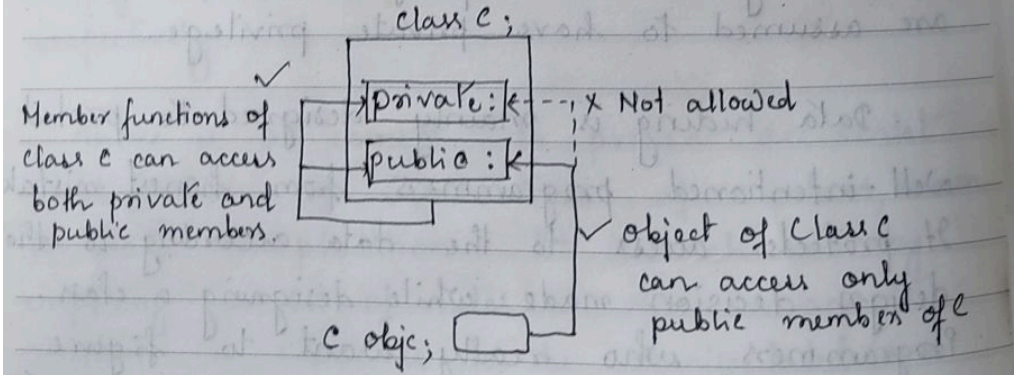


fig: Class member accessibility.

Double linked list : Double linked list is a l
in which each node has address of previous and n
the data structure for doubly linked list will

Date

examples.

Private members.

```
class person
{
    private :
            int age ;          private data
            int getage ();     private function.
};
person p1;
a = p1. age ;    x cannot access private data
p1. getage ();   x  "      "  private function.
```

fig : Private members accessibility.

Protected Members.

```
class person
  { protected :
            int age;
            int getage();
  };
    Person p1;
    a = p1. age ;   } → cannot access protected memb
    p1. getage();     (same as private).
```

Date

Public members.

```
class person
  {
    public:
        int age;
        int getage();
  };
person p1;
a = p1.age;  → can access public data
p1.getage(); → can access public function
```

fig: Public members accessibility.

Date

Member functions :

A function declared within a class definition is called member function. Member functions are mostly declared in the public section of a class, because they have to be called outside the class definition. The member functions that have been declared in the private section of a class can be accessed only from within the class.

In C++, the member functions can be coded in two ways →
- (a) Inside class definition
- (b) Outside class definition using scope resolution operator.

(a) Inside Class definition :

When a member function is defined inside a class, we do not require to place a membership label along with the function name. We use only small functions inside the class definition and such functions are known as inline functions.

In case of inline function the compiler inserts the code of the body of the function at the place where it is invoked (called) and in doing so the program execution is faster but memory penalty is there.

Date

eg.

```
class student
{
    int rollno;
    char name[20];
public: void read data()
        { function body; }
        void disp()
        { function body; }
};
```

(b) Outside class definition using scope resolution operator (::)

Member functions that are declared inside a class definition can be defined outside the class definition.

The general form of a member function definition is

```
return type class name :: function name (argument list)
{
    function body;
}
```

Date

A member function of a class is defined using the scope resolution operator (::).
Here the operator :: known as scope resolution operator helps in defining the member function outside the class.
It tells the compiler that the function belongs to that particular class. Without this operator the function difinition would behave as an ordinary function.

eg.

```
class student
{
    int rollno;
    char name[20];
  public :
        void readdata();
        void disp();
};
void student :: readdata()
{   function body;
}
void student :: disp()
{   function body;
}
```