

CLASS , MEMBER FUNCTION AND OBJECT




PRESENTED BY – ANJALI SONA
DEPARTMENT OF COMPUTER SCIENCE

Introduction of Class and Object

- ▶ OOP के लिए Class और Object बहुत ही महत्वपूर्ण हिस्सा है ।
- ▶ OOP का concept Objects और Classes पर निर्भर है ।
- ▶ Class के अन्दर बहुत सारे member function और data members हो सकते है । जिसको object के जरिये access किया जाता है ।
- ▶ Class के अन्दर जो variables होते है उन्हें data members कहते है और जो functions होते है उन्हें member functions कहते है ।

for eg.

- ▶ अगर कोई प्राणी है तो उस प्राणी का behavior और properties जैसे कि, उसका भौकना, चलना, देखना , उसके शरीर की रचना को data members और member functions के जरिये लिखा जाता है ।
- ▶ हर एक से अधिक प्राणी को अलग-अलग नाम से उनके objects भी बनाये जाते है ।



Class क्या होता है ?

- ▶ Class एक structure के जैसा होता है | जिसमे variable(data members) और function(member functions) को एक ही जगह पर इकट्ठा किये जाते है | Data और functions class के members होते है | Class data को hold करने का काम करता है | Class ये object का layout होता है |

Defining Class

- ▶ Class को define करने के लिए class keyword का इस्तेमाल किया जाता है । इसके साथ 'Classname' जाता है । Class के नाम का पहला character Uppercase या Lowercase दोनों भी हो सकता है । लेकिन Class के नाम का पहला अक्षर Uppercase में होना एक 'Good Programming' कहलाता है ।
- ▶ Class के लिए और भी कुछ महत्वपूर्ण हिस्से हैं । जैसे कि, Access Specifier
- ▶ Access Specifier / Modifier का use Access Control करने के लिए किया जाता है ।

Access Specifier तीन प्रकार के होते हैं ।

1. private
2. public
3. protected



member function

- ▶ जैसे कि हम जानते हैं class में data member और member function शामिल होता है। क्लास के data members केवल जानकारियाँ हैं यह किसी कार्य को प्रदर्शित नहीं कर सकता है जबकि member function के द्वारा data member का उपयोग करके class के आवश्यक कार्य को किया जाता है। उपयोगकर्ता से data प्राप्त करना, उसे हेरफेर करना, स्क्रीन पर data दिखाना आदि member function के समान्य कार्य हैं।

Defining Member function

- ▶ member function का उपयोग, एक class को एक structure से अलग करता है। data member को class के अंदर declare किया जाता है, जबकि member function को class के अंदर और class के बाहर दोनों जगह define किया जा सकता है, इसलिए आप इनमें से कोई एक तरीका चुन सकते हैं। member function को define करने का syntax दोनों तरीकों में अलग-अलग होता है।

member function definition inside the class:

SYNTAX-

```
returntype functionName(argument declaration)
{ // Body of Function
};
```


EXAMPLE-

```
Class Customer
{
public:
char name[15];
char address[10];
double payable_amount;
void input_data(void)
{
cout<<"Enter customer name";
cin>>name;
cout<<"Enter customer address";
cin>>address;
cin<<"Enter amount to be paid";
cin>>payable_amount;
}

};
```


Member function definition outside the class:

- ▶ member functions को class के बाहर define करने लिए के निम्नलिखित कार्य करने की आवश्यकता होती है:
- ▶ क्लास डेफिनिशन के अंदर member function के prototype को declare करें।
- ▶ member function definition को उसके class के साथ scope resolution operator (::) का उपयोग करके बांधें। (अर्थात् जब member functions को class के बाहर define करते हैं तो compiler को कैसे पता चलता है कि member function किस class से संबंधित है, इसलिए scope resolution operator (::) का उपयोग करके फंक्शन और क्लास को एक साथ bind किया जाता है।



```
Class Customer
```

```
{  
  public:  
  char name[15];  
  char address[10];  
  double payable_amount;  
  void input_data();           //Prototype of member function  
};
```

```
\ member function definition outside the class.
```

```
void Customer::input_data(void)           // :: tells the compiler that the input_data function belongs to Customer class  
  
{  
  cout<<"Enter customer name";  
  cin>>name;  
  cout<<"Enter customer address";  
  cin>>address;  
  cout<<"Enter amount to be paid";  
  cin>>payable_amount;  
}
```

1. private :

- ▶ private में class के members मतलब variables लिखे जाते है | private के बिना private के members लिखे जाते है वो default private member होता है | private member सिर्फ उसके class के लिए ही काम करते है | ये class के बाहर access नहीं होते |
- ▶ एक class के private members केवल उसी class के members के लिए उपलब्ध होते हैं। private members, class के बाहर के functions के लिए accessible नहीं होता है, यानी private access specifier, members तक सीधी पहुंचने से रोकता है, जिससे class का data बाहरी एक्सेस से छिपा होता है, इस गुण को data hiding कहते हैं। data hiding किसी ऑब्जेक्ट के data members को पूरी तरह से encapsulate करने में सक्षम बनाता है।
- ▶ **Class के members को private के रूप के declare करने के की प्रक्रिया को Employee क्लास के उदहारण से समझते है:**

```
class Employee
{
private:
int empcode;
char empname[15];
char emppost[10];
float empsalary;
public:
void display();
};
```

2. public :


- ▶ public के members class के अंदर और बाहर दोनों ही जगह पर access किये जाते हैं | इसके अन्दर data members को भी लिखा जाता है |
- ▶ नीचे दिए गए Employee class के द्वारा सीखते कि कैसे members को public member के रूप में declare करते हैं:
- ▶

```
class Employee
{
public:
int empcode;
char empname[15];
char emppost[10];
float empsalary;
void display();
};
```





3. protected

- ▶ protected access specifier, inheritance में महत्वपूर्ण भूमिका निभाता है। जब नए class को मौजूदा class से उत्पन्न(derived) किया जाता है, तो नए class को subclass कहा जाता है। protected member की उपलब्धता(accessibility) की दायरा private के समान है, केवल अंतर यह है कि private members, subclass के द्वारा एक्सेस नहीं किया जा सकता है लेकिन protected members सबक्लास द्वारा एक्सेस किया जा सकता है।



```
class Employee
{
private:
int empcode;
char empname[15];
float empsalary;
};
class Manager: private Employee
{
Void input(void)
{
cout<<"\n Enter employee code :";
cin>> empcode;
cout<<"\nEnter employee Name :";
cin>> empname;
cout<<"\nEnter employee Salary :";
cin>>empsalary;
};
```

- 
- 
- ऊपर के code segment में एक Employee क्लास बनाया गया है और इस class से Manager नाम का subclass derived किया गया है। Employee क्लास के members, protected member के रूप में declare है।
 - Manager क्लास के member function यानि input() के द्वारा Employee class की data members को access किया गया है जो संभव है, क्योंकि यह protected data members है। अगर यह private या पब्लिक member होता तो इसे एक्सेस नहीं किया जा सकता था।



THANK YOU