(422)

Faculty Name :- Hemant Kumar Singh

Subject :- OOP's

Code :- 3EE4

Batch & Branch :- A & B (EE Branch)

Unit - 2

Department C.S./I.T.

Semester (A+B)

26/10/13

26/10/13

27/10/13

Department of Computer Science
Modern Institute of Technology
& Research Centre

③ Polymorphism :→ "poly" means many and "Morphism" means form. Thus polymorphism means more than one form.

It is work on the concept of ~~one interface~~ multiple method. In here method name are same but they are different in their parameter and perform different behaviour.

Polymorphism is of two types :→

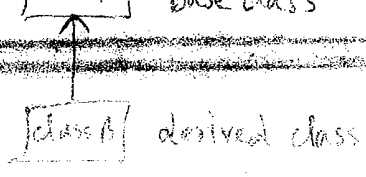
i) compile time (early binding, static binding)

ex :→ method overloading, constructor overloading, Operator overloading.

ii) run time (late binding, dynamic binding)

ex :→ method overriding, virtual function

④ Inheritance :→ It is work on the concept of code reusability. In here one class access the feature of another class mean inherit the property another class (code once written can be used again again in number of few classes.



Message Passing :→ In Java objects communicate each other by passing messages to each other.

Message passing involves specifying the name of the object, the name of the method (message) and the information to be sent.

example :→ Employee . salary (name) ;

- Employee → object
- salary → method name
- (name) → information (parameter value)

Message passing is nothing but calling the method of the class and sending parameters. The method in turn executes in response to a message.

Data hiding :→ Data hiding hides the data from external access by the user. In OOP's have special keywords like private, protected, which hides the data.

Dynamic Binding :→ Binding means linking. If linking of function call to function definition i.e. a place where control has to be transferred is done at compile time it is known as static binding or early binding.

(5) Assignment Operator

+=, -=, *=, /=, %=, =

```
x += b
x = x+b;
```

(6) Bitwise Operator :→ They operate on bits.

(1) Bitwise AND (&)

```
int a=2, b=3;
int c = a&b;
cout << "c=" << c;
```

output :- (2)

a = 0 0 1 0

b = 0 0 1 1

0 0 1 0

It takes two bits as operands and return the value 1 if both are 1. If either of them is 0, the result is 0.

(2) Bitwise OR (|)

It returns 1 if any of the bit is 1. If both are 0 then result is 0.

```
int a=12, b=7;
int c = a|b;
cout << "c=" << c
```

output : (15)

a = 1 1 0 0

b = 0 1 1 1

c = 1 1 1 1

Call by reference

```
void main()
{
 int a,b;
 cout<<"enter two no.";
 cin>>a>>b;
void swap(int &x, int &y)
 swap(a,b);
 getch();
}
```

```
void swap(int &x,int &y)
{
 int t;
 t = x;
 x = y;
 y = t;
 cout<< x;
 cout<<y;
 }
```

Constructor :→ A constructor is a special member function whose name is same as the name of its class in which it is declared and defined. The purpose of the constructor is to initialize the object of the class. So when object is created it automatic call to the constructor.

```
#include <iostream.h>
class demo
{
 public :
  demo( )
  {
   cout<<" Hello" ;
  }
};
void main( )
{
 demo d;
}
```


following points of constructor :→

(1) The constructor is always public. but if declared in the private then object can only be created inside the class but serve no purpose.

```
class Demo
{
  int a;
  int b;
  public:
  void show()
  {
    a=10;
    b=20;
    cout<<a<<b;
  }
};
```

```
void main()
{
  Demo d1;
  d1.show();
  getch();
}
```

(2) Parameterized Constructor →

Parameterized Constructor → Those constructor which have ~~parameter~~ one or more parameter.

```
class Demo
{
  int a;
  int b;
  public:
  Demo(int c, int d)
  {
    a=c;
    b=d;
  }
  void show()
  {
    cout<<a<<b;
  }
};
```

```
void main()
{
  Demo d1(10,20);
  d1.show();
  Demo d2(30,40);
  d2.show();
  getch();
}
```

Demerits of friend function

1. friend function cannot access the class members and functions directly. they need to have a class object which using dot can call the members of the class.
2. Creating friend classes and functions defy the idea of encapsulation and create exception in the ways of data hiding.
3. usage of many friend functions sometimes make you think to redesign your program.