
CAP444

OBJECT ORIENTED PROGRAMMING

USING C++

Unit- 3



Created By:
Kumar Vishal
(SCA), LPU

Topics covered

Run-time polymorphism and virtual functions :

- virtual base classes,
- abstract classes,
- pointer to object,
- this pointer,
- pointer to derived class,
- virtual function,
- pure virtual function,
- early vs late binding

Situation

Student

Exam

Project

Result

regNo, Name
studentDetails()

regNo, Name
studentDetails()

regNo, Name
studentDetails()

regNo, Name
studentDetails()

ambiguity arises as to which
data/function member would
be called?



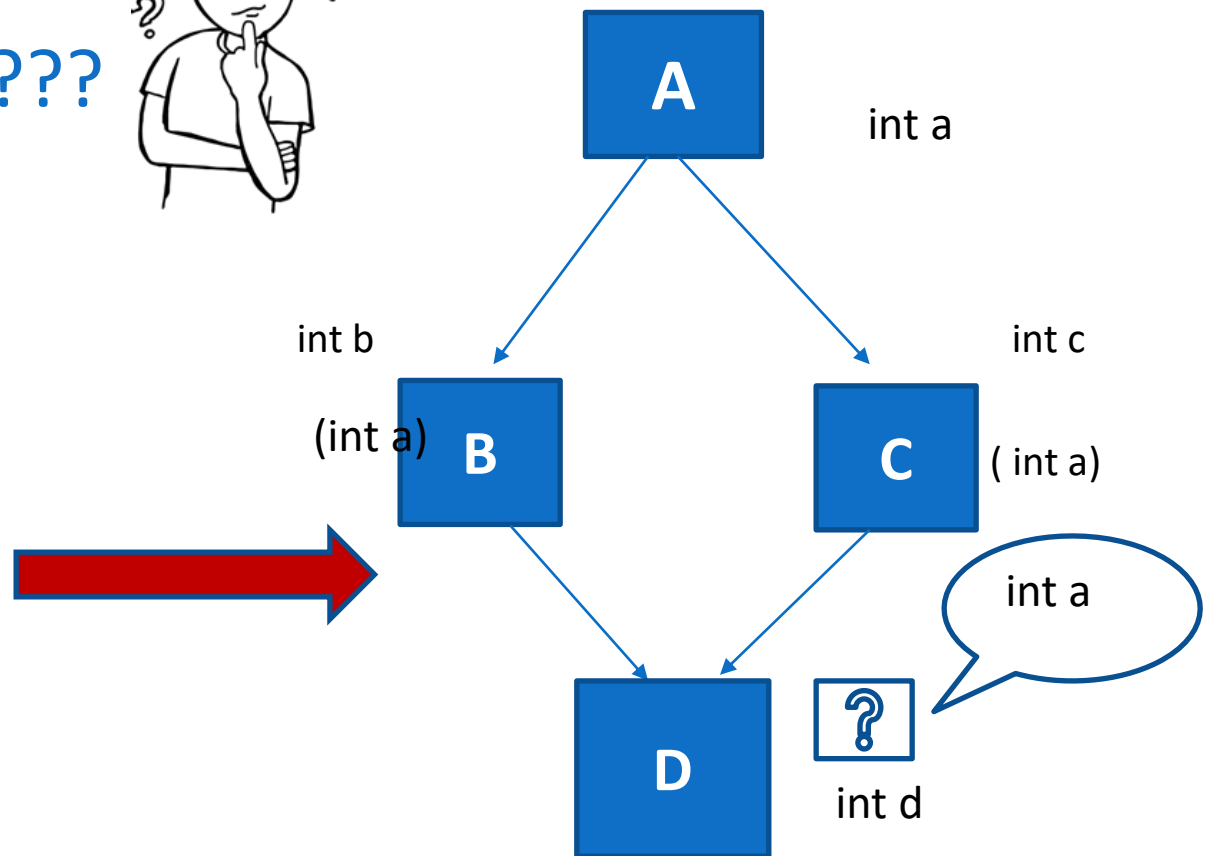
virtual base class introduce

virtual base classes

- It means we are making base class as virtual
- But why ???
- In situation???



Let's assume



Situation

Media Player

PlayMovie()
PlaySong()

Punjabi Movie

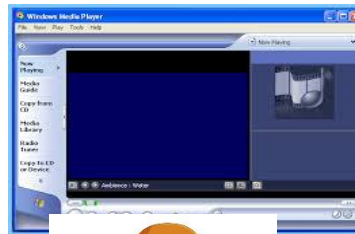
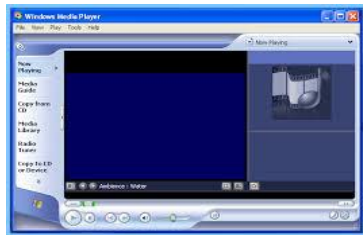
PlayMovie()
PlaySong()

Telugu Movie

PlayMovie()
PlaySong()

Hindi Movie

PlayMovie()
PlaySong()



What's meaning it....



Its method overriding



- Problems in method overriding
 - Overriding using base class pointer compiler don't know about pointer which address is pointing because address will be decided at run time when memory will be allocated.
 - To over come this problem we use virtual function
 - With the help of virtual function, we can override function at run time

Situation



Transportation:
Vehicle_Registration()

Punjab_Transport
Vehicle_Registration()



**Punjab
Transport
Department**

PB081234

Himachal_Transport
Vehicle_Registration()



**Himachal Road
Transport
Corporation**

HP014567

Karnataka_Transport
Vehicle_Registration()



**Karnataka State Road Transport
Corporation**

KA023333

Bihar_Transport
Vehicle_Registration()



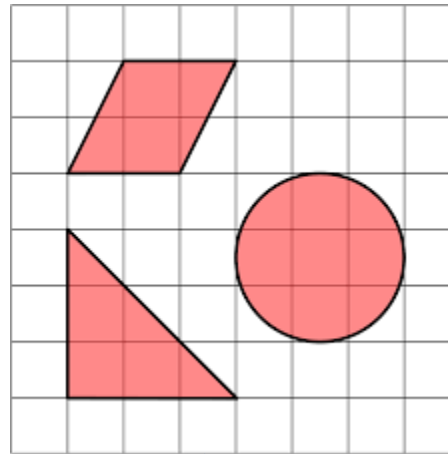
**Transport Department
Government of Bihar**

BR012345

Abstract Class

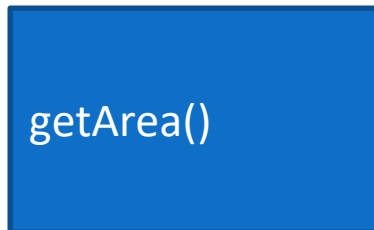
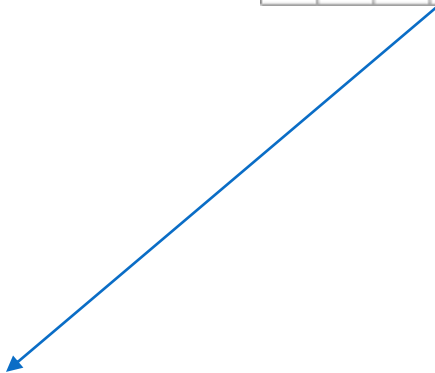
- *Sometimes implementation of function is not required in a base class such a class is called abstract class.*
- *In such cases we have to make function as abstract function by using virtual keyword that is also called pure virtual function.*
- *A pure virtual function is declared by assigning 0 in declaration.*

Example



Area

getArea()



getArea()

Area = $w \times h$
 w = width
 h = height



getArea()

Area = $\pi \times r^2$

Important points:

- *A class is abstract if it has at least one pure virtual function.*
- *If we do not override the pure virtual function in derived class, then derived class also becomes abstract class.*
- *An abstract class can have constructors.*
- *We cannot create objects of abstract classes.*

What will be the output of the following C++ code?

```
#include<iostream>
using namespace std;
class Mobile
{
long int contactno;
public:
    virtual void getStatus() = 0;
    Mobile()
    {
        contactno=9898989890;
        cout<<contactno;
    }
};
```

```
class Person: public Mobile
{
public:
    void getDetails() { cout << "Blocked"; }
};
int main()
{
    Person p;
    p.getDetails();
    return 0;
}
```

- A. 9898989890
- B. Blocked
- C.9898989890Blocked
- D. Error



What will be the output of the following C++ code?

```
#include<iostream>
using namespace std;
class Mobile
{
long int contactno;
public:
    virtual void getStatus() = 0;
    Mobile()
    {
        contactno=9898989890;
        cout<<contactno;
    }
};
```

```
class Person: public Mobile
{
public:
    void getStatus(){ cout <<"Blocked"; }
    void getDetails(){cout<<"Kumar";}
};

int main()
{
    Person p;
    p.getDetails();
    return 0;
}
```

- A. 9898989890
- B. kumar
- C. 9898989890kumar
- D. Error

Pointer to object

- A variable that holds an address value is called a pointer variable
- Object can also have an address , so there is also a pointer that can point to the address of an object.

```
class Date
```

```
{  
};
```

```
Date d1;
```

```
Date *d2;
```

```
d2=&d1;
```

```
d2->functions()
```



object pointer

this pointer

- It can be used **to refer current class instance variable.**

Syntax:

`this-> instance variable=value`

Example

this pointer

This keyword represent address of current instance of the class.

Example

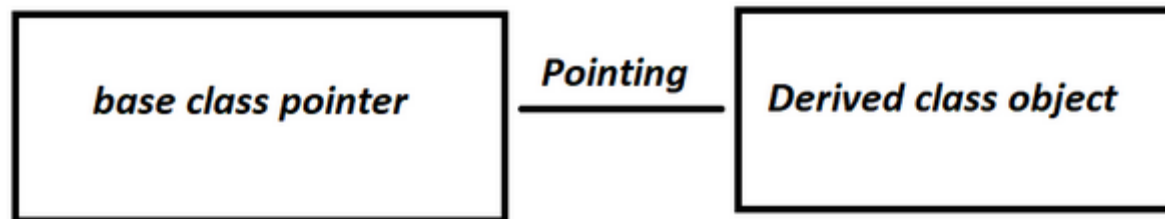
this pointer

To return reference to the calling object.

Example

pointer to derived class

- A pointer of one class can point to other class, but classes must be a base and derived class, then it is possible.
- To access the variable of the base class, base class pointer will be used.



Pointer to derived class

```
class base{};  
class derive: public base{};  
base b1,*b2;  
derive d1;  
b2=&d1;
```

What will be the output of the following C++ code?



```
#include <iostream>
using namespace std;
class mobile
{
public:
    void playRingTone()
    {
        cout<<"mobile ring tone"<<endl;
    }
};
class samsung:public mobile
{
    void playRingTone()
    {
        cout<<"samsung ring tone"<<endl;
    }
};
```

```
class realme:public mobile
{
    void playRingTone()
    {
        cout<<"real me ringtone"<<endl;
    }
};
int main()
{
    mobile *mptr;
    samsung s1;
    mptr=&s1;
    mptr->playRingTone();
    return 0;
}
```

- A. mobile ring tone
- B. samsung ring tone
- C. real me rington
- D. Error



Any Query?