

```

class Figure
{
protected:
    int dim1,dim2;
public:
    void get()
    {
        cout<<"Enter dimensions:";
        cin>>dim1>>dim2;
    }
    virtual void area()=0;
};

```

*Abstract class*

*Figure obj X*  
*Figure \*P;*

*This is a pure virtual fn*

```

class Rectangle: public Figure
{
public:
    void area()
    {
        cout<<"Area of rect="<<dim1*dimn2;
    }
};
class Triangle: public Figure
{
public:
    void area()
    {
        cout<<"Area of tri="<<.5*dim1*dimn2;
    }
};

```

*Concrete class*

```

int main()
{
    Figure * P=NULL;
    int choice;
    cout<<"Select a figure:"<<endl;
    cout<<"1. Rect"<<endl<<"2.Triangle"<<endl;
    cin>>choice;
    switch(choice)
    {
        case 1:
            P=new Rectangle;
            P->get();
            P->area();
            break;
        case 2:
            P=new Triangle;
            P->get();
            P->area();
            break;
    }
}

```

```

default:
    cout<<"Wrong choice";
}

if(P!=NULL)
    delete P;

return 0;
}

```

```

int main()
{
    Figure * P;
    int choice;
    cout<<"Select a figure:"<<endl;
    cout<<"1. Rect"<<endl<<"2.Triangle"<<endl;
    cin>>choice;
    switch(choice)
    {
        case 1:
            P=new Rectangle;
            P->get();
            P->area();
            break;
        case 2:
            P=new Triangle;
            P->get();
            P->area();
            break;
        default:
            cout<<"Wrong choice";
            return 0;
    }

    delete P;

    return 0;
}

```

Syntax:

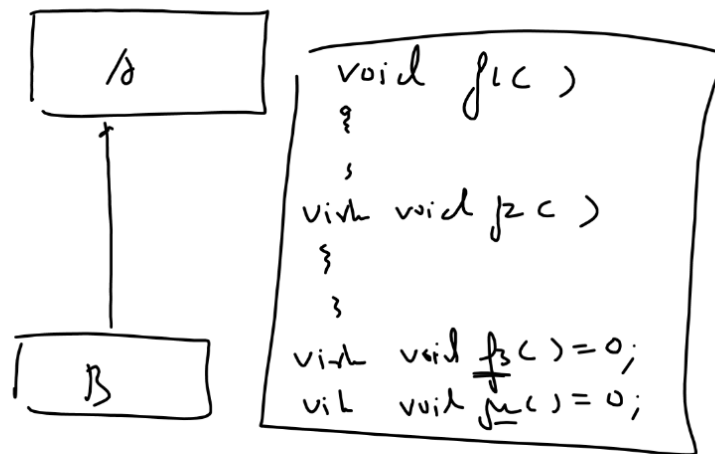
=====

virtual <ret\_type> <fn\_name>(arg)=0;

Example:

=====

virtual void area()=0;



After this session you will be able to answer:

1. why virtual functions run at slow pace ?
2. why constructor cannot be declared as virtual ?
3. why compiler always defines a default constructor in our class if we do not define any constructor ?

