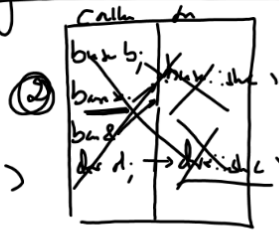


Polymorphism (Run Time Polymorphism)

class base

```
{
  public:
  virtual void show()
  {
    cout << "In show of base";
  }
}
```

```
};
class drv : public base
{
  public:
  void show()
  {
    cout << "In show of drv";
  }
}
```



Run Time Polymorphism

int main()

```
{
  base *p;
  base b;
  drv d;
```

✓ p = &b;

b->show();

✓ p = &d;

p->show();

return 0;

Fruit

Mango

Fruit *p;

Mango m;

✓ p = &m;

Orange o;

✓ p = &o;

$p = \&b;$ <div style="border: 1px solid black; padding: 2px; display: inline-block;"> $p \rightarrow \text{show}();$ </div> $p = \&d;$ <div style="border: 1px solid black; padding: 2px; display: inline-block;"> $p \rightarrow \text{show}();$ </div>	$b.\text{show}();$ $d.\text{show}();$ <hr/> $d.\text{show}();$ $d.\text{base}::\text{show}();$
--	---

```

class base
{
public:
    virtual void show()
    {
        cout<<"In show of base";
    }
};

class drv:public base
{
public:
    void show(int i)
    {
        cout<<"In show of drv";
    }
};

```

Err!

```

int main()
{
    base *p;
    base b;
    drv d;
    p=&b;
    p->show()
    p=&d;
    p->show(10);
    return 0;
}

```

Virtual Functions And Multilevel...

```
class base
{
    public:
        virtual void show()
        {
            cout<<"In show of base";
        }
};

class drv1:public base
{
    public:
        void show( )
        {
            cout<<"In show of drv1";
        }
};
```

```
class drv2:public drv1
{
    public:
        void show( )
        {
            cout<<"In show of drv2";
        }
};

int main()
{
    base *p,b;
    p=&b;
    p->show();
    drv1 d1;
    p=&d1;
    p->show();
    drv2 d2;
    p=&d2;
    p->show();
    return 0;
}
```