

## Assignment - 22

Q1] What is meant by Runtime polymorphism?

Ans Polymorphism is an object-oriented paradigm.

Polymorphism is single name & multiple behaviour.

There are two types of polymorphism

1] Compile-time polymorphism

2] Run-time polymorphism

1] To achieve Run-time polymorphism we use the concept of overriding. It performs dynamic/Late Binding.

2] Binding is connection function call to its body.

3] When which function to call gets checked at run-time, it is called as Run-time.

Q2] What is difference between Overriding & Overloading?

Ans

1] ~~Function~~ Overloading is used to achieve compile-time polymorphism

2] Overriding is used to achieve run-time polymorphism

2] There are ~~one~~ two types of each overloading :  
1] Function overloading  
2] Operator overloading

Overriding :  
1] virtual  
2] Pure virtual



3] In function Overloading we can define multiple function with same name & different prototype within the same class.

In Function Overriding ~~the~~ we can define a ~~sa~~ method in Base class & redefine that method in the derived class.

So when we use a Base class pointer & point to derived object, it calls the method of Base class. If the method is declared virtual in base class, it calls the method of derived class.

Q3] Explain internal implementation of a class which contains virtual function in it with VTABLE & VPTR.

Ans Example

```
Class Base {
```

```
    public:
```

```
        int i, j;
```

```
        virtual void fun()           // 1000
```

```
    {}
```

```
        void gun()                   // 2000
```

```
    {}
```

```
        virtual void sun             // 3000
```

```
    {}
```

```
};
```



class Derived : public Base

{

public:

int a, b;

// 4000

void fun()

{

// 5000

void sun()

{

// 6000

virtual void mun()

{

}

int main()

{

Base \*bp = new Derived;

bp → fun()

bp → gun()

bp → sun()

bp → mun()

return 0;

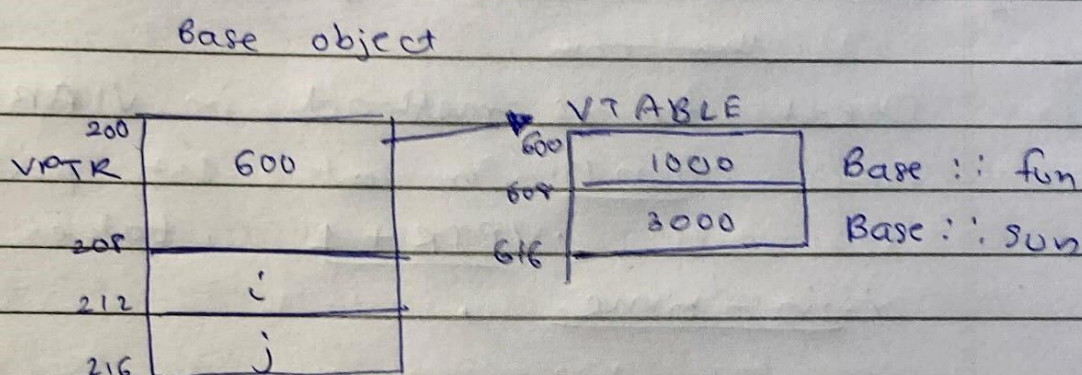
}

1] Whenever a class contains virtual function or if a class is derived from such a class which contains ~~pure~~ virtual function in it then the first 4/8 bytes of memory is reserved for the VPTR.

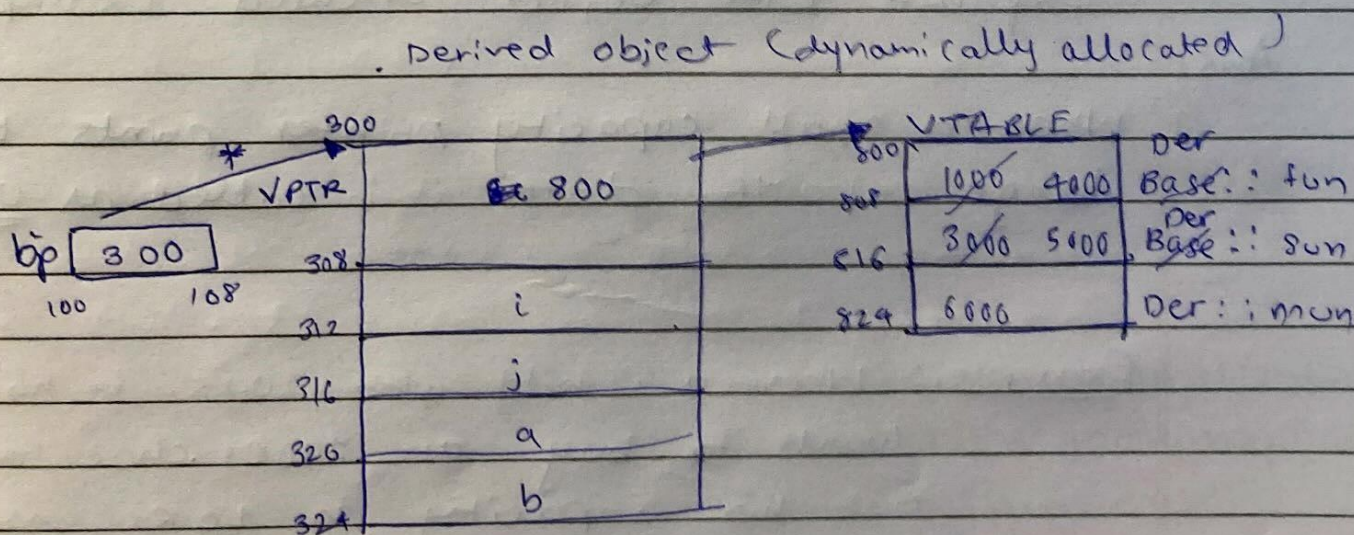
2] The VPTR holds address of the VTA.



- 3] The VTABLE contains all virtual function address in sequence in which they're declared.
- 4] Even though we do not declare Base object explicitly, it is implicitly inside Derived object



- 5] When we create a derived object, the base class object gets inherited inside the derived class object along with the VTABLE.



- 6] During compilation if the derived class contains redefinition of the virtual function from base class, then that method from the VTABLE gets overridden by the redefinition in the derived class.



Example : When a person with 100Rs wants to buy a 10Rs object, is called as down casting.

Q5] If a base class & derived class contains same named & prototyped method, which is considered as Overloading or Overriding or redefinition?

Ans  
It is considered as redefinition, since it is not given that base class contains virtual function & other requirements for Overriding are also not mentioned.

Q6] In which scenario of class first 4 bytes are ~~are~~ reserved as VPTR in object layout?

Ans  
If a class contains a virtual function or if a class is derived from such a class which contains virtual function in it, then the first 4/8 bytes of the object of that class are reserved for the VPTR.

Q7] What are the necessary things in our application which is used to achieve run time polymorphism?

Ans  
To achieve run-time polymorphism, our application should contain ..

- 1] Redefinition of a function (use of a virtual keyword to achieve overriding)
- 2] Concept of upcasting
- 3] At least Single level Inheritance.



7] All of this this overriding takes place at the time of compilation but which method to call gets decided at run-time, due to which it is called Run-time polymorphism.

8] When we call  $bp \rightarrow sun()$ , the compiler ~~the com~~ checks for a method in VTABLE. Since sun() is at 1st position in VTABLE the call VPTR + 1, takes place at run-time.

9] This is how VTABLE & VPTR's are implemented internally.

Q4] What is meant by upcasting & downcasting? Explain with object oriented Example.

Ans

1] When a small capacity pointer points to a large capacity data it is called as UPCASTING.

Example: When a child with 10 Rs. in his pocket wants to buy an 100 Rs. chocolate. But his capacity is 10 Rs only.

2] When a large capacity pointer points to a small data it is called as DOWNCASTING ('DOWNCASTING is not allowed')



Ans

When a large capacity pointer points to a data having less size ~~at~~; it is called as down casting.

It is not allowed because if a 20 bytes pointer points to a 12 bytes object, the remaining 8 bytes may contain other data out of address space.

Due to which it may result in run-time errors like segmentation fault.

Q10]

Can we define static virtual function in a class?

Ans

No, we cannot define a static virtual function since, virtual function concept is purely based on object. And static methods are irrespective of object.