

## Assignment - 20

1. Draw object layout of below code snippets and explain its internal working.

```

1. class base
{
    public:
        int i;
        float f;
        double d;
        void func()
        { }
        void gunc()
        { }
}

```

```

}

```

```

class derived: public base

```

```

{
    public:
        int i;
        double d;
        void gunc()
        { }
        void func()
        { }
}

```

```

}

```

```

int main()
{
    base bobj;
    derived dobj;
    return 0;
}

```

## Object Layout:

bobj		dobj	
100	i	200	i
104	f	204	f
108	d	208	d
116		216	i
		220	
		228	d

• All function stored on text section.

## Internal working:

- In above code, single level inheritance used.
- class derived inherits properties of its parent, base class.

```
2. class base1
{ public:
    int i;
    float f;
    void gun()
};
```

```
class base2
{ public:
    int j;
    float g;
    void fun()
    {}
};
```

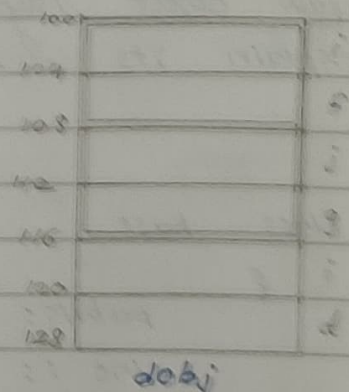
```
class derived : public base1, base2
```

```
{
    public:
        int i;
        double d;
        void sun()
        {}
        void fun()
        {}
};
```

```
int main()
```

```
{
    derived dobj;
    return 0;
}
```

o object layout :



Internal working :

- In above code snippet multiple inheritance is used.

- class base1 and base2 share their data with class derived.

8. class base

{

public:

int i;

float f;

void fun() //1000

{}

virtual void gun() //2000

{}

};

class derived : public base

{

public:

int i;

double d;

virtual void fun() //3000

{}

void gun() //4000

{}

virtual void sun() //5000

{}

int main()

{

derived dobj;

return 0;

}

## • Object Layout

Base

VTABLE

VPTR

500

500

2000

Base::  
gun

f

0

No call to Base.

Base

VTABLE

VPTR

600

600

fun  
1000 3000

0

gun  
2000 4000

f

0

sun  
5000

d

0

## • Internal working

The above code snippet demonstrates the concept of overriding.

## • internal working



4. class base

{ public:

int i;

float f;

virtual void func() //1000

{}

virtual void gunc() //2000

{}

virtual void sun() //3000

{}

virtual void runc() //4000

{}

};

class derived : public base

{

public:

int i;

double d;

virtual void func() //5000

{}

virtual void gunc() //6000

{}

virtual void sun() //7000

{}

virtual void runc() //8000

{}

};

int main() {

derived obj;

return 0;

}

• object layout :

obj

VPTR  $\frac{100}{104}$ 

500

108

Base::i

112

Base::f

116

derived::i

124

derived::d

VTABLE

func()

500

~~1000~~ 5000

504

gunc()

~~2000~~ 6000

508

sun()

~~3000~~ 7000

512

runc()

~~4000~~ 8000

520

5.

class base

{

public:

int i;

float f;

void gun() // 1000

{}

virtual void sun() // 2000

{}

};

class derived : public base

{

public:

int i;

double d;

virtual void func() // 8000

{}

void gun() // 4000

{}

virtual void sun() // 5000

{}

};

int main()

{

derived obj;

return 0;

}

• object layout :

96  
V PTR  
100

500

base :: i

base :: f

derived :: i

derived :: d

500

func()

derived :: fun

3000

304

gun()

derived :: gun

4000

308

base :: sun

sun()

derived :: sun

2000 3000

112