Assignment No. - 16

Q.1 what are the types of inheritance according to the architecture ?

→ • Inheritance means reusability. one class can inherit another class.

• Types of Inheritance :
1. Single level
2. Multi - level
3. Multiple
4. Hierarchal

1. single Level - In single level inheritance Base class shared behaviours and characteristics of class to the derived class.

2. Multilevel : same as single level but it should have 3 or more classes.

3. Multiple - Two or more class share data / behaviours with one derived class.

4. Hierarchical - one parent class can share data with two or more derived class.

1.
$$\begin{tikzcd} A \arrow[d] \\ B \end{tikzcd}$$

2.
$$\begin{tikzcd} A \arrow[d] \\ B \arrow[d] \\ C \end{tikzcd}$$

3.
$$\begin{tikzcd} A \arrow[dr] & & B \arrow[dl] \\ & C & \end{tikzcd}$$

4.
$$\begin{tikzcd} & A \arrow[dl] \arrow[dr] & \\ B & & C \end{tikzcd}$$

Page No.
Date

Q.2 Explain the concept of access Specifier in detail ?

→ • Access specifier is concept used in class to limit the access of characteristics and member function.

• There are three Access Specifiers : (c++)

1. Public
2. Private
3. Protected

• Public . If we write the [code] using public [access] keyword. then we can access these data anywhere from [code] program.
(class members)

• Private - This is an default access specifier. Which specifies that we can access data within the class only. cannot accessed outside the class.

• Protected . To provide access to the child class we can use protected access Specifier.

Q.3 what is difference between private & protected access -Specifier.

→ • Private - If we declare class members using private keyword then it can be accessible for only these class only. we cannot access it out the class.

• Protected - If we declare class members using protected they it can be accessible only for its derived class.

• for- eg →

```
        class  Demo {
                        private:
                          int  a;
                        protected :
                          int b;
              Demo ( ) };

    class Hello  :  public  Demo {
                    public :
                     cout << " value of b :", b <<endl;
              };

   int  main () {
                  Demo  obj :
                  obj. a ;     // NA
                  obj. b ;     // A
```

Q.4 what is the default access specifier it it is not writting explicitely.

—• • private - is the default access specifier.

• IF we do not write ~~not~~ any access specifier then it is considered as private member of class.

• Example -

```
class Demo {
        int a;  // private
                   Access
    };
```

Q.5. what is the Inheritance according to access specifier?

— • According to the access Specifier there are 3 Inheritances as :

1. Public Inheritance :
2. Private ~~Protected~~ inheritance
3. Protected Inheritance

• 1. Public Inheritance :
- public members of Base class becomes public members of derived class.

• for eg -

```
class Base {
            public:
              int A;
    };
```

```
class Derived : public Base{
            // A is accessible
        }
```

- Private members of Base cannot be accessible to derived.
- protected members of Base can be accessible to derived.

2. Private - it makes the protected and public members of base class private [inaccessible) to derived class.

- Example -
```
class Demo
{ public :
      int a;
  protected :
      int b;
};
class Hello : private Demo
{
      // a & b Not asccessible
};
```

3. Protected - Public & protected members of Base become protected to Derived class.

```
class Base {
      public :
          int a;
};
~~class Base~~
class Derived : protected Base {
          // Accessible a   };
```

Page No.
Date

Q.6 Explain the constructor and destructor calling sequence in case of single level, multi-level and multiple inheritance.

→ 1. Single Level - In single level, constructor of Base class [constructor] will be called first then constructor of derived class gets called.

- Destructor of Derived called first then destructor of Base gets called.

2. Multilevel - constructor and Destructor calling sequence is same as Single level Inheritance.

3. Multiple - In case of multiple Inheritance, constructor of Base class calling depends on the Sequence of Base class & destructor as well.

Q.7 Draw object layout & class diagram of below code snippets and explain its internals working in detail. Explain the type of Inheritance in the below code snippet.

→

Page No.
Date

Q.7
```
class Base
{ public :
    int i ;
  float f ;
  double d;
  void fun()
    { }
  void gun()
    { }
};
```

```
class derived : public Base
{  public:
     int i ;
     double d ;
     void sun ()
       { }
};

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

- object Layout of above code snippet.

bobj

100		i
104		f
108		d
116		

Base

dobj :

100		i
104		f
108		d
116		i
120		d
128		

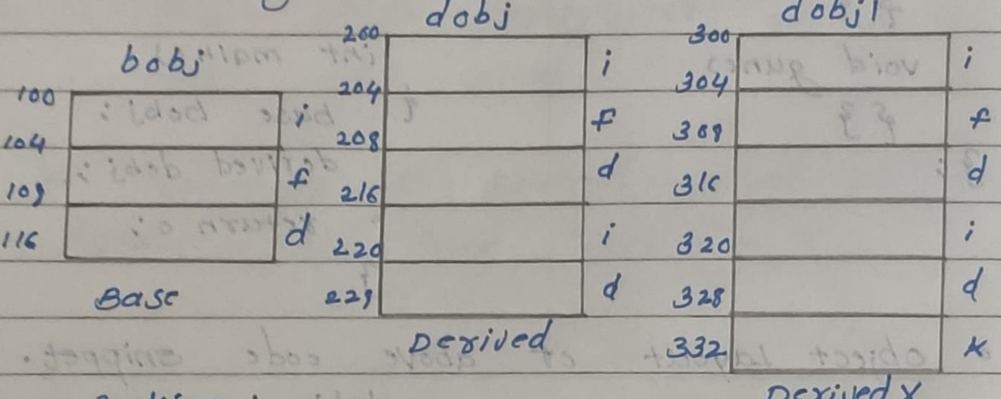
Derived

- In above code single level Inheritance is used.

- Explanation - - In above code, Base is parent class and derived is a child class. Derived inherit properties of Base class.
  - Base class have three data members & two behaviours and Derived have initially 2 data members & one method in it.
  - As Derived inherit Base class it total data members becomes five & total three methods.
  - Base class is of 16 byte and Derived 28 byte as it inherit Base. [16+12].

Q.9. Draw object layout & class diagram of below code snippet and explain its internal working in detail. Explain type of inheritance in the below code snippet.

--> <u>class Diagram & object Layout</u>



- <u>Working</u> :-
- In above code, Three classes are used/ defined as Base, Derived and DerivedX.
- Base consist of 3 characteristics and 2 methods
  Derived have 2 characteristics & 1 behaviour initially
  Derivedx have 1 characteristics & 1 behaviour initially.
- Derived class derived from Base class and DerivedX class derived from derived class.
- Derived class inherit properties of Base ∴ therefore no. of characteristics & behaviour becomes 5 & 3 respectively.
- DerivedX class inherit properties of Derived ∴ no. of Data mebers & methods becomes 6 & 4 respectively.
- Size of ∴ Base - 16 bytes
  Derived - 28 bytes
  DerivedX - 32 bytes

- <u>Multilevel Inheritance</u> is used in above code.

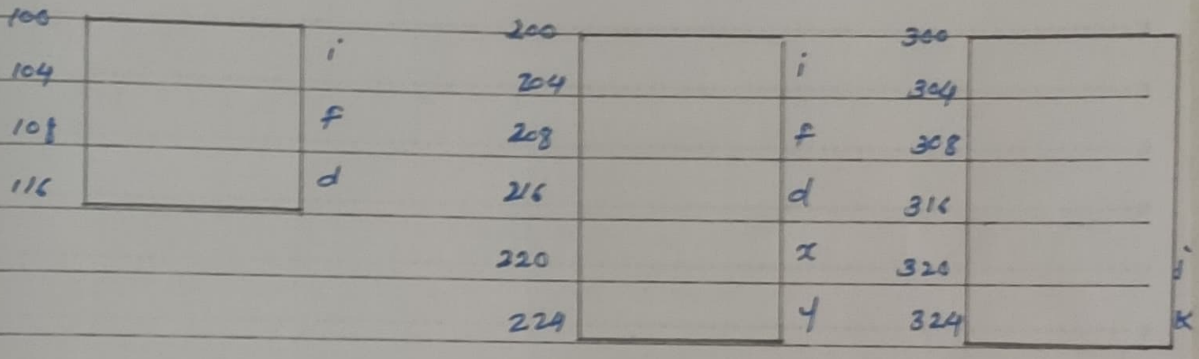
NOTES

Q.10 Draw object layout & class Diagram of below code snippet & explain.

→ Here, Hierarchical inheritance is used.

$$\begin{tikzcd}
 & \boxed{\text{Base}} \arrow[dl] \arrow[dr] & \\
\boxed{\text{derived1}} & & \boxed{\text{derived2}}
\end{tikzcd}$$

- class Base will share the properties with Derived1 & Derived2 class.



~~100~~
104
108
116
i
f
d
~~200~~
204
208
216
220
224
i
f
d
x
y
~~300~~
304
308
316
320
324
j
k