

Assignment

16

Q1] Explain concept of Inheritance & what are types of inheritance w.r to the architecture?

Refer Assignment 13 Q6

Types w.r to Architecture:

- 1) Single level
- 2) Multi level
- 3) Multiple level
- 4) Hierarchical
- 5) Hybrid

Q2] Explain concept of Access specifier in detail

Refer Assignment 11 Q7

Q3] What is the difference between public & private Access specifier

Refer Assignment 11 Q7

Program:- access-cpp

Q4] What is default Access specifier

1. for classes:-

The default access specifier is private.

Eg: class MyClass{

```
    int x; // x is private by default
};
```

2. for struct:

The default access specifier is public

Eg: struct MyStruct{
 int x; // x is public by default
};

Q5] What are the types of inheritances w.r to access specifiers
Inheritance can be categorized into three types based on the access specifiers used to derive a class from its base class: public, protected & private inheritance.

1. Public Inheritance:

public members of the base class remain public in the derived class.

protected members remain protected.

private members of the base class are inaccessible

```
Ex: class Base {  
    public: int x;  
    protected: int y;  
    private: int z;  
};  
  
class Derived: public Base {  
    // x is public  
    // y is protected  
    // z is inaccessible  
};
```

2. Protected Inheritance:

Base class members:

Public & protected members of the class become protected in the derived class.

private members are inaccessible

```
Ex: class Derived: protected Base {  
    // x & y are protected  
    // z is inaccessible  
};
```

3. Private Inheritance:

Base class members:

public & protected members of the base class become private in the derived class

private members are inaccessible

Ex class Derived : private Base {

// x & y are private

// z is inaccessible

Q6] Explain Constructor & Destructor calling sequence in case of single level, multi level & multiple inheritance.

1. Single level & Multi level

Constructor

Destructor

Top to Bottom

Bottom to top

Demo → Hello
(Top) (Bottom)

Demo ← Hello
(Top) (Bottom)

Multiple Level:

Constructor & Destructor calling sequence depends on sequence of inheritance

Q7] Draw object layout & class diagram of below code snippets & explain its internal working in detail. Explain the type of inheritance in below code snippet.

100	bobj	
104	0	Base::i
108	0	Base::f
116	0	Base::d

Characteristics: i, f, d

Behaviours: fun(), gun()

200	dobj	
204	0	Base::i
208	0	Base::f
216	0	Base::d
220	0	Derived::i
228	0	Derived::d

Characteristics: i, f, d, i, d

Behaviours: fun(), gun(),
3unc()

It is a type of single level Inheritance

When creating an object of Derived (dobj):

- Constructor of Base class is called 1st to initialize inherited members.
- Constructor of Derived class initializes its own member

Q8] Same as Q7

The code displays Multiple Inheritance.

Derived class inherits base1 & base2

Object of Derived (dobj) is created: Calling Sequence

- 1st - constructor of base1
- 2nd - constructor of base2
- 3rd - Constructor of Derived

100	dobj	
104	0	base::i
108	0	base::f
112	0	base2::j
116	0	base2::d
120	0	Derived::i
124	0	Derived::d

characteristics:

i, f, j, g, i, d

Behaviours:

func(), gun(), sun()

Q9] Same as Q7

This example shows Multilevel Inheritance

100	0	base::i
104		base::f
108	0	

characteristics:

int i, float f

Behaviour:

func(), gun()

200	0	base::i
204		base::f
208	0	Derived::i
212	0	Derived::d
220	0	Derived::d

characteristics:

int i, float f, int i, doubled

behaviours:

func(), sun(), gun()

300	0	base::i
304		base::f
308	0	base::f
312	0	Derived::i
320	0	Derived::d
324	0	Derived::t

Characteristics:

int i, float f, int i, doubled, int k

Behaviours:

func(), gun(), sun(), run()

Here derived is inheriting base class & further Derivedx is inheriting Derived class.

All the characteristics & behaviours of base class are passed to Derived class & further all the characteristics & behaviours of Derived class are inherited to Derivedx class.

Q10] Same as Q7

100		
104	0	base::i
108	0	base::f
116	0	base::d

characteristics:

int i, float f, double

behaviours:

fun(), gun()

200		
204	0	base::i
208	0	base::f
216	0	base::d
220	0	Derived1::g
224	0	Derived1::y

characteristics:

i, f, d, z, y

Behaviours:

fun(), gun(), run()

300		
304	0	
308	0	
316	0	
320	0	Derived2::i
324	0	Derived2::k

Characteristics:

i, f, d, j, k

Behaviours:

fun(), gun(), son()