AM TUITION CENTRE, NERKUNDRAM, CHENNAI-107

CHAPTER 16: INHERITANCE

UNIT- IV OBJECT ORIENTED PROGRAMMING WITH C++

STUDY MATERIAL

MOHAMED FAKRUDEEN,
PGT-COMPUTER SCIENCE,
SHAMROCK MATRIC. HR. SEC. SCHOOL,
MOGAPPAIR, CHENNAI

PART I

Choose the correct answers

1. **Which of the following is the process of creating new classes from an existing class?**

   (a) Polymorphism (b) Inheritance (c) Encapsulation (d) super class

2. **Which of the following derives a class student from the base class school?**

   (a) school: student (b) class student : public school

   (c) student : public school (d) class school : public student

3. **The type of inheritance that reflects the transitive nature is**

   (A) Single Inheritance (B) Multiple Inheritance

   (C) Multilevel Inheritance (D) Hybrid Inheritance

4. **Which visibility mode should be used when you want the features of the base class to be available to the derived class but not to the classes that are derived from the derived class?**

   (A) Private (B) Public (C) Protected (D) All of these

5. **Inheritance is process of creating new class from**

   (A) Base class (B) abstract (C) derived class (D) Function

6. **A class is derived from a class which is a derived class itself, then this is referred to as**

   (A) multiple inheritance (B) multilevel inheritance

   (C) single inheritance (D) double inheritance

7. **Which amongst the following is executed in the order of inheritance?**

   (A) Destructor (B) Member function (C) Constructor (D) Object

8. **Which of the following is true with respect to inheritance?**

   (A) Private members of base class are inherited to the derived class with private

(B) Private members of base class are not inherited to the derived class with private accessibility

(C) Public members of base class are inherited but not visible to the derived class

(D) Protected members of base class are inherited but not visible to the outsideclass

9. Based on the following class declaration answer the questions (from 9.1 o 9.4)

```
class vehicle
{
int wheels;
public:
void input_data(float,float);
void output_data();
protected:
int passenger;
};
class heavy_vehicle : protected vehicle
{
int diesel_petrol;
protected:
int load;
protected:
int load;
public:
voidread_data(float,float)
voidwrite_data();
};
class bus: private heavy_vehicle
```

```
{
charTicket[20];
public:
voidfetch_data(char);
voiddisplay_data(); };
};
```

9.1. Which is the base class of the class heavy_vehicle?

(a) Bus (b) heavy_vehicle (c) vehicle (d) both (a) and (c)

9.2. The data member that can be accessed from the function displaydata()

(a) passenger (b) load (c) Ticket (d) All of these

9.3. The member function that can be accessed by an objects of bus Class is

(a) input_data() , (b) read_data() ,output_data()write_data()

(c) fetch_data() (d) All of these display_data()

9.4. The member function that is inherited as public by Class Bus

(a) input_data() , (b) read_data() , output_data()write_data()

(c) fetch_data() (d) All of these

10. What is the order of constructor for object z1 to be invoked?

```
class x
{ int a;
public :
x()
{}
};
class y
{ x x1;
```

```
public :
y(){}
};
class z : public y,x
{
int b;
public:
z(){}
}z1;
```

(A)z , y,x,x (B) x,y,z,x (c) y,x,x,z (D) x,y,z

ANSWERS FOR OBJECTIVES				
1) B	2)D	3)C	4)A	5)A
6)B	7)C	8)B	9.1)C	9.2)D
9.3)C	9.4)D	10)D		

PART II

2 Marks:

1. **What is inheritance?**

   The mechanism of deriving new class from an existing class is called inheritance.
2. **What is a base class?**

   The class to be inherited is called base class or parent class
3. **Why derived class is called power packed class?**

   The derived class is a power packed class, as it can add additional attributes and methods and thus enhance its functionality.
4. **In what multilevel and multiple inheritance differ though both contains many base class?**

- In multiple inheritance the base classes do not have any relationship between them.
- In multilevel inheritance the level of inheritance can be extended to any number of level depending upon the relation.

5. What is the difference between public and private visibility mode?

Public:

- All other classes and methods can access.

Private:

- It is a default visibility mode
- No other classes and methods could access.

PART III

3 Marks:

1. What are the points to be noted while deriving a new class?

i. The keyword *class* has to be used
ii. The name of the derived class is to be given after the keyword *class*
iii. A single colon
iv. The type of derivation (the visibility mode), namely private, public or protected. If no visibility mode is specified, then by default the visibility mode is considered as private.
v. The names of all base classes (parent classes) separated by comma.

2. What is difference between the members present in the private visibility mode and the members present in the public visibility mode ?

The members present in the public visibility mode

- A public member is accessible from anywhere outside the class but within a program.
- The public member of a class can be inherited.

The members present in the private visibility mode

- A private member variable or function cannot be accessed, or even viewed from outside the class.
- The private member of a class cannot be inherited.

3. What is the difference between polymorphism and inheritance?

✓ Inheritance allows, code reusability and the polymorphism is, the occurrence of one function with different form.

- ✓ Inheritance allows the already existing code to be reused again in a program, and polymorphism provides a mechanism to dynamically decide what form of a function to be invoked.

4. What do you mean by overriding?

When a derived class member function has the same name as that of its base class member function ,the derived class member function shadows/hides the base class's inherited function .This situation is called function overriding

5. Write some facts about the execution of constructors and destructors in inheritance

- When an object of the derived class is created, the compiler first call the base class constructor and then the constructor of the derived class.
- This because the derived class is built up on the members of the base class.
- When the object of a derived class expires first the derived class destructor is invoked followed by the base class destructor.

PART IV

5 Marks:

1. Explain the different types of inheritance

There are different types of inheritance viz., Single Inheritance, Multiple inheritance, Multilevel inheritance, hybrid inheritance and hierarchical inheritance.

i. **Single Inheritance**

When a derived class inherits only from one base class, it is known as single inheritance

ii. **Multiple Inheritance**

When a derived class inherits from multiple base classes it is known as multiple inheritance

iii. **Hierarchical inheritance**

When more than one derived classes are created from a single base class , it is known as Hierarchical inheritance.

iv. Multilevel Inheritance

The transitive nature of inheritance is itself reflected by this form of inheritance. When a class is derived from a class which is a derived class – then it is referred to as multilevel inheritance.

v. Hybrid inheritance

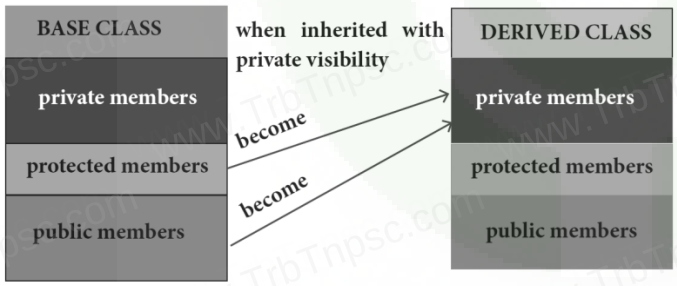
When there is a combination of more than one type of inheritance, it is known as hybrid inheritance.

2. Explain the different visibility mode through pictorial representation

- ✓ An important feature of Inheritance is to know which member of the base class will be acquired by the derived class. This is done by using visibility modes.
- ✓ The three visibility modes are private, protected and public.

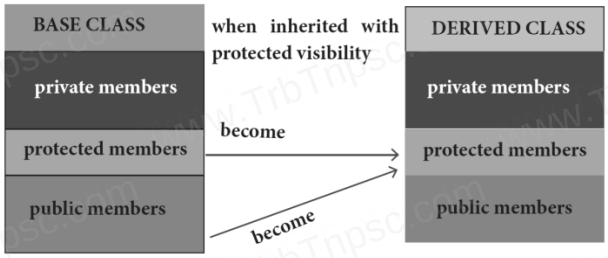
Private visibility mode

- ✓ When a base class is inherited with private visibility mode the public and protected members of the base class become 'private' members of the derived class



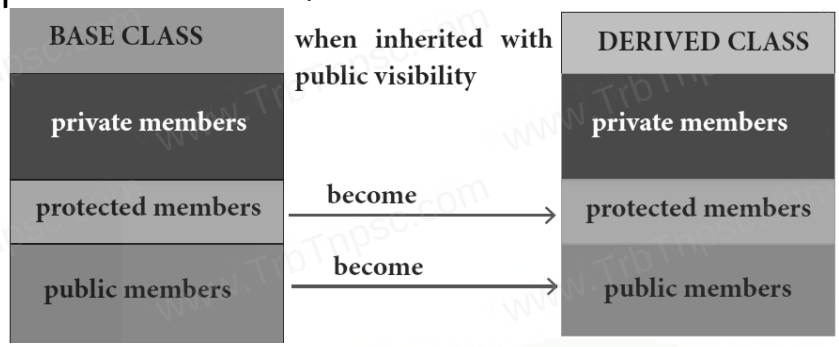
Protected visibility mode

When a base class is inherited with protected visibility mode the protected and public members of the base class become 'protected members ' of the derived class



Public visibility mode

When a base class is inherited with public visibility mode , the protected members of the base class will be inherited as protected members of the derived class and the public members of the base class will be inherited as public members of the derived class.



3. Consider the following c++ code and answer the questions

```
#include<iostream>
#include<string.h>
#include<stdio.h>
using name spacestd;
class publisher
{
char pname[15];
char hoffice[15];
char address[25];
double turnover;
protected:
char phone[3][1O];
      void register();

public:
publisher();
~publisher();
void enter data();
void disp data();
};
class branch
{
charbcity[15];
char baddress[25];
protected:
```

```
int no_of_emp;
public:
charbphone[2][10];
branch();
~branch();
void have data();
void give data();
};
class author: public branch, publisher
{
int aut_code;
char name[20];
float income;
public:
author();
~author();
voidgetdata();
voidputdata();
};
```

Answer the following questions based on the above given program:

3.1. Which type of Inheritance is shown in the program?

Multiple Inheritance

3.2. Specify the visibility mode of base classes.

public

3.3 Give the sequence of Constructor/Destructor Invocation when object of class author is created.

Sequence of Invocation of constructors: branch, publishers and author
Sequence of Invocation of destructors: author, publishers and branch

3.4. Name the base class(/es) and derived class (/es).

Base classes: branch, publishers
Derived class:author

3.5 Give number of bytes to be occupied by the object of the following class:

(a) publisher (b) branch (c) author

Number of bytes to be occupied by the object of publisher class: 93 bytes
Number of bytes to be occupied by the object of branch class: 64 bytes
Number of bytes to be occupied by the object of author class: 185 bytes

3.6. Write the names of data members accessible from the object of class author.

Data members accessible from the object of class author are:
phone, no_of_emp, aut_code, name, income

3.7. Write the names of all member functions accessible from the object of class author.

```
void register();

void enter data();
void disp data();
publisher();
branch();
void have data();
void give data();
voidgetdata();
voidputdata();
```

3.8 Write the names of all members accessible from member functions of class author.

```
char phone[3][10];
char bphone[2][10];
int no_of_emp;
```

4. Consider the following c++ code and answer the questions

```
class Personal
 {
 int Class,Rno;
 char Section;
 protected:
 char Name[20];
 public:
 personal();
```

```
void pentry();
voidPdisplay();
};
class Marks:private Personal
{
float M{5};
protected:
char Grade[5];
public:
Marks();
void M entry();
void M display();
};
class Result:public Marks
{
float Total,Agg;
public:
char FinalGrade, Commence[20];
Result();
void R calculate();
void R display();
};
```

4.1. Which type of Inheritance is shown in the program?

Multilevel Inheritance

4.2. Specify the visibility mode of base classes.

Personal-private
Marks-public

4.3 Give the sequence of Constructor/Destructor Invocation when object of class Result is created.

Sequence of Invocation of constructors: Personal, Marks, Result
Sequence of Invocation of destructors: Resulit, Marks and Personal

4.4. Name the base class(/es) and derived class (/es).

Base classes: Personal and Marks

Derived classes: Marks and Result

4.5 Give number of bytes to be occupied by the object of the following class: (a) Personal (b) Marks (c) Result

Number of bytes to be occupied by the object of Personal class: 29 bytes
Number of bytes to be occupied by the object of Marks class: 54 bytes
Number of bytes to be occupied by the object of Result class: 83bytes

4.6. Write the names of data members accessible from the object of class Result.

Grade, Total, Agg, FinalGrade, Commence

4.7. Write the names of all member functions accessible from the object of class Result.

Rcalculate(), Rdisplay(), Mentry(), MDisplay()

4.8 Write the names of all members accessible from member functions of class Result.

Data Members:

Grade, Total, Agg, FinalGrade, Commence

Member functions:

Mentry(), MDisplay()

5. Write the output of the following program

```
#include<iostream>
using namespace std;
class A
{
protected:
int x;
public:
void show()
{
cout<<"x = "<<x<<endl;
}
A()
{
cout<<endl<<" I am class A "<<endl;
}
```

```
~A()
{
cout<<endl<<" Bye ";
}
};
class B : public A
{
{
protected:
int y;
public:
B(int x, int y)
{
this->x = x; //this -> is used to denote the objects datamember
this->y = y; //this -> is used to denote the objects datamember
}
B()
{
cout<<endl<<" I am class B "<<endl;
}
~B()
{
cout<<endl<<" Bye ";
}
void show()
{
cout<<"x = "<<x<<endl;
cout<<"y = "<<y<<endl;
}
};
int main()
{
A objA;
B objB(30, 20);
objB.show();
return 0;
}
```

Output:

```
I am class A
I am class A
x=20
y=20
Bye
Bye
Bye
```

6. Debug the following program

Output

```
------------
15
14
13
Program :
------------
%include(iostream.h)
#include<conio.h>
Class A
{
public;
int a1,a2:a3;
Void getdata[]
{
a1=15;
a2=13;a3=13;
}
}
Class B:: public A()
{
PUBLIC
voidfunc()
{
int b1:b2:b3;
A::getdata[];
b1=a1;
b2=a2;
a3=a3;
cout<<b1<<'\t'<<b2<<'t\'<<b3;
}
void main()
{
clrscr()
B der;
```

der1:func();
getch();
}

Error NO.	Error Code	Corrected Code
1	%include	#include
2	(iostream.h)	<iostream.h>
3	Class A	class A
4	public;	public:
5	int a1,a2:a3;	int a1,a2,a3;
6	Void getdata[]	void getdata()
7	Class B:: public A()	class B: public A
8	PUBLIC	public:
9	voidfunc()	void func()
10	int b1:b2:b3;	int b1,b2,b3;
11	A::getdata[];	a.getdata();
12	a3=a3;	b3=a3;
13	clrscr()	clrscr();
14	der1:func();	der1.func();