

Chapter – 16

Inheritance

PART – I

I. Choose the correct answer

Question 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

Answer:

- (b) Inheritance

Question 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

Answer:

- (b) class student: public – School

Question 3.

The type of inheritance that reflects the transitive nature is

- (a) Single Inheritance
- (b) Multiple Inheritance
- (c) Multilevel Inheritance
- (d) Hybrid Inheritance

Answer:

- (c) Multilevel Inheritance

Question 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

Answer:

- (a) Private

Question 5.

Inheritance is process of creating new class from

- (a) Base class
- (b) abstract
- (c) derived class
- (d) Function

Answer:

- (a) Base class

Question 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

Answer:

- (b) multilevel inheritance

Question 7.

Which amongst the following is executed in the order of inheritance?

- (a) Destructor
- (b) Member function
- (c) Constructor

(d) Object

Answer:

(c) Constructor

Question 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 outside class

Answer:

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

Question 9.

Based on the following class declaration answer the questions (from 9.1 to 9.4)

```
class vehicle
{ int wheels;
public:
void input_data(float,float);
void output_data();
protected:
int passenger;
};
class heavyvehicle : protected vehicle {
int diesel_petrol;
protected:
int load;
protected:
int load;
public:
void read_data(float,float)
```

```
void write_data(); };  
class bus: private heavy_vehicle {  
    char Ticket[20];  
public:  
    void fetch_data(char); void display_data(); };  
};
```

Question 9.1.

Which is the base class of the class heavy_vehicle?

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

Answer:

- (c) vehicle

Question 9.2.

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

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

Answer:

- (d) All of these

Question 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()

Answer:

- (d) All of these display_data()

Question 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 display_data()

Answer:

- (d) All of these display_data()

Question 10.

```
class x
{int a;
public :
x()
}
};
class y
{ x x1;
public :
y()
}
};
class z : public y, x
{ intb;
public:
z(){}
} z1;
```

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

- (a) z, y, x, x
- (b) x, y, z, x
- (c) y, x, x, z
- (d) x, y, z

Answer:

- (c) y, x, x, z

PART – II

II. Answers to all the questions

Question 1.

What is inheritance?

Answer:

It is a process of creating new classes called derived classes, from the existing or base classes.

Question 2.

What is a base class?

Answer:

A class that is used as the basis for inheritance is called a superclass or base class.

Question 3.

Why derived class is called a power-packed class?

Answer:

The derived class is a power-packed class, as it can add additional attributes and methods and thus enhance its functionality.

Question 4.

In what multilevel and multiple inheritances differ though both contain many base class?

Answer:

The derived class is a power-packed class, as it can add additional attributes and methods and thus enhance its functionality.

Question 5.

What is the difference between public and private visibility mode?

Answer:

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.

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.

PART – III

III. Answers to all the questions

Question 1.

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

Answer:

The following points should be observed for defining the derived class.

- The keyword class has to be used.
- The name of the derived class is to be given after the keyword class.
- A single colon.
- 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 private.
- The names of all base classes(parent classes) separated by a comma.

Question 2.

What are the differences between the members present in the private visibility mode and the members present in the public visibility mode?

Answer:

Members present in the private visibility mode:

- Can be accessed only by the class members.

- By default the members will be in private visibility mode.
- When classes are inherited, private members are not inherited.

Members present in the public visibility mode:

- Can be accessed by outside members also.
- Members, to be in public visibility mode has to be specified explicitly.
- When classes are inherited, the public members are inherited as private, protected and public members of the derived class.

Question 3.

What is the difference between polymorphism and inheritance though both are used for the reusability of code?

Answer:

Inheritance:

- Promotes code sharing and reduces development cost.
- Inheritance is implemented in classes.
- Inheritance type may be a single inheritance, multiple inheritances, multilevel inheritance, hierarchical inheritance, and hybrid inheritance.

Polymorphism:

- Reduces the complexity of the program.
- Polymorphism is a common interface for multiple forms.
- Polymorphism is basically implemented on function.
- Polymorphism may be a compile Time polymorphism (overloading) or run-time polymorphism (overriding).

Question 4.

What do you mean by overriding?

Answer:

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.

Question 5.

Write some facts about the execution of constructors and destructors in inheritance. Some Facts About the execution of constructor in inheritance

Answer:

Some facts about the execution of constructors in inheritance.

- Base class constructors are executed first, before the derived class constructors execution.
- A derived class cannot inherit the base class constructor but it can call the base class constructor by using.
- If there are multiple base classes, then it starts executing from the leftmost base class.
- In multilevel inheritance, the constructors will be executed in the order of inheritance.

Some facts about the execution of the destructor in inheritance.

- Own (Derived) class constructors are executed first, before the base class destructors execution.
- A derived class cannot inherit the destructor.
- In multilevel inheritance, the destructors will be executed in the reverse order of inheritance.

PART – IV

IV. Answers to all the questions

Question 1.

Explain the different types of inheritance.

Answer:

Types of Inheritance: There are different types of inheritance viz., Single

inheritance, Multiple inheritance, Multilevel inheritance, hybrid inheritance, and hierarchical inheritance.

1. **Single Inheritance:** When a derived class inherits only from one base class, it is known as single inheritance.
2. **Multiple Inheritance:** When a derived class inherits from multiple base classes it is known as multiple inheritances.
3. **Hierarchical inheritance:** When more than one derived class is created from a single base class, it is known as Hierarchical inheritance.
4. **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.
5. **Hybrid inheritance:** When there is a combination of more than one type of inheritance, it is known as hybrid inheritance.
6. Hence, it may be a combination of Multilevel and Multiple inheritances or Hierarchical and Multilevel inheritance or Hierarchical, Multilevel and Multiple inheritances.

Question 2.

Explain the different visibility modes through pictorial representation.

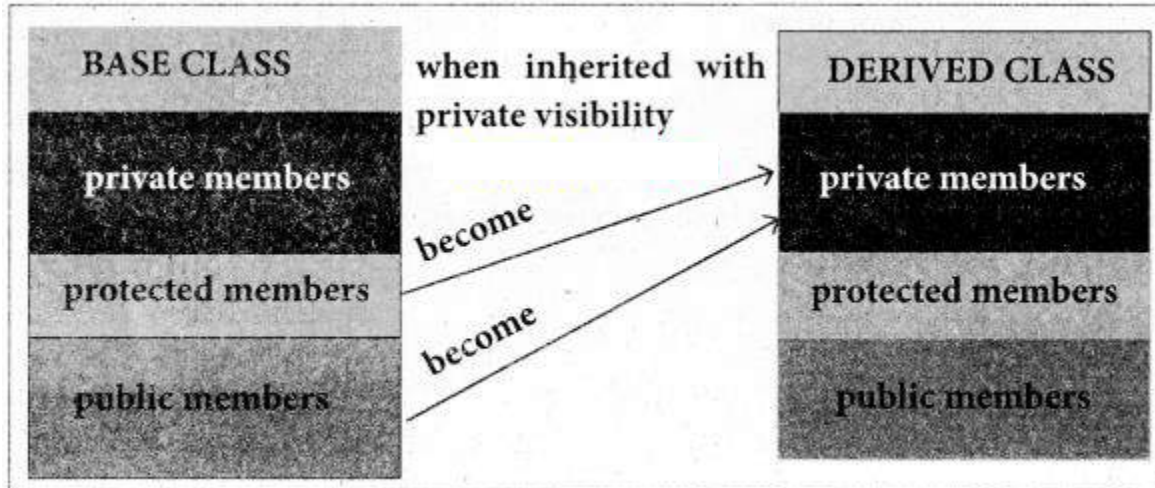
Answer:

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 accessibility of base class by the derived class is controlled by visibility modes. The three visibility modes are private, protected, and public.

The default visibility mode is private. Though visibility modes and access specifiers look similar, the main difference between them is Access specifiers control the accessibility of the members within the class whereas visibility modes control the access of inherited members within the class.

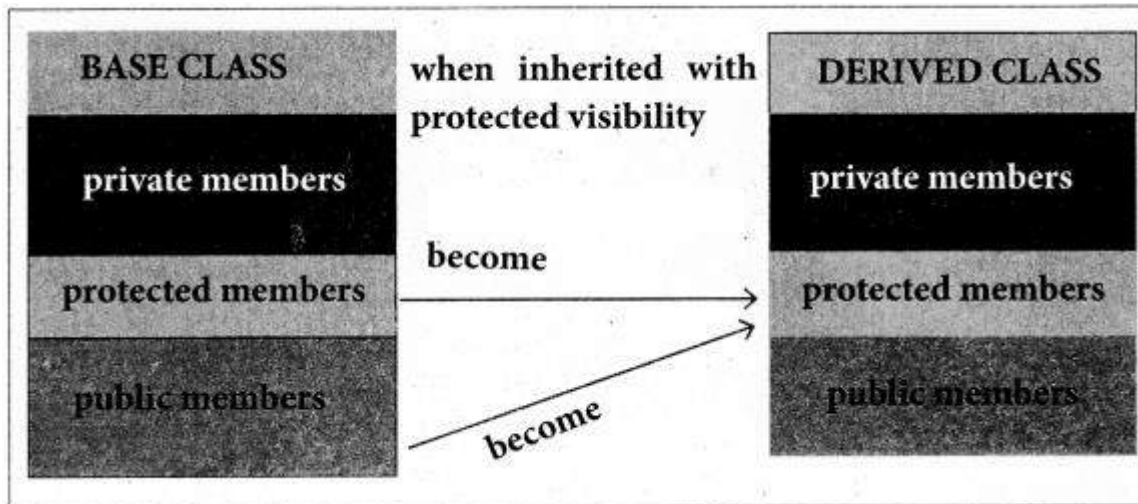
Private visibility mode:

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

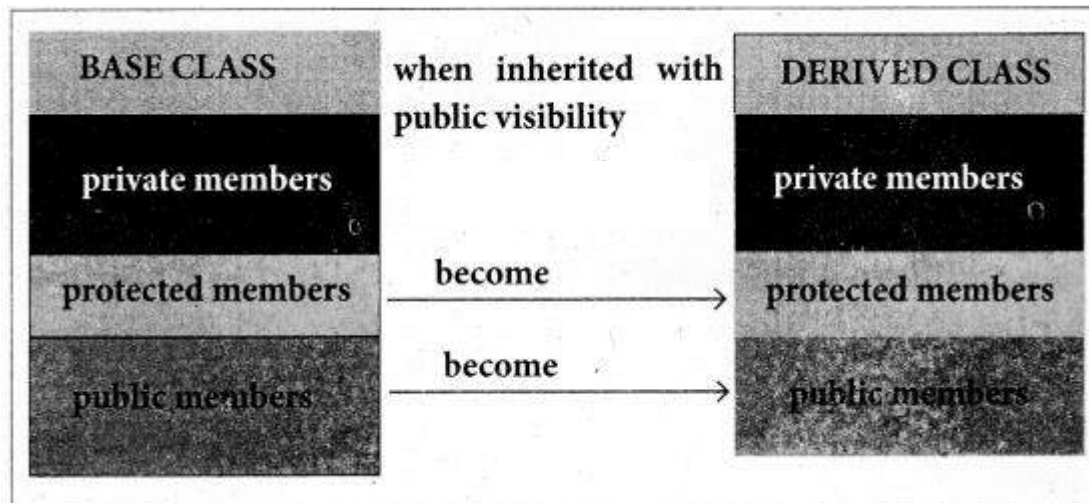


Protected visibility mode:

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.



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.



Question 3.

```
#include
#include
#include
using namespace std;
class publisher
{
    char pname[15];
    char hoffice[15];
    char address[25];
    double turnover;
protected:
    char phone[3][10];
    void register();
public:
    publisher();
    ~ publisher();
    void enter data();
    void disp data();
};
class branch
{
    char bcity[15];
    char baddress[25];
protected:
    int noofemp;
```

```

public:
charbphone[2][10];
branch();
~ branch();
void have data();
void give data();
};
class author: public branch, publisher
{
intaut_code;
charaname[20];
float income;
public:
author();
~author();
voidgetdata();
voidputdata();
};

```

Answer the following questions based on the above-given program:

Question 3.1.

Which type of Inheritance is shown in the program?

Answer:

Multiple inheritances.

Question 3.2.

Specify the visibility mode of base classes.

Answer:

Private for the publisher.

Public for a branch.

Question 3.3.

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

Answer:

```

branch(); // constructor of branch class
publisher (); // constructor of publisher class
author (); // constructor of author class
~author (); // destructor of author class
~publisher (); // destructor of publisher class
~branch (); // destructor of branch class

```

Question 3.4.

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

Answer:

Base class(/es): branch and publisher

Derived class (/es): author.

Question 3.5.

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

(a) publisher

(b) branch

(c) author

Publisher		Branch		Author	
<u>Data member</u>	<u>Bytes</u>	<u>Data member</u>	<u>Bytes</u>	<u>Data member</u>	<u>Bytes</u>
char pname [15]	15	char bcity[15]	15	char pname [15]	15
char hoffice [15]	15	char address[25]	25	char hoffice [15]	15
char address [25]	25	int no_of_emp	4	char address [25]	25
double turnover	8	char bphone[2][10]	20	double turnover	8
char phone[3][10]	30			char phone[3][10]	30
				char bcity[15]	15
				char address[25]	25
				int no_of_emp	4
				char bphone[2][10]	20
Total	93	Total	64	int aut_code	4
				char aname[20]	20
				float income	4
				Total	185

Question 3.6.

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

Answer:

The data members that can be accessed is bphone[2][10].

Question 3.7.

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

Answer:

Member functions that can be accessed are:

1. public member functions of branch have_data();
give_data();

2. public member functions of author:
getdata();
putdata();

Question 3.8.

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

Answer:

Data members accessible	Member functions accessible
char phone[3][10] int no_of_emp char bphone[2][10] int aut_code char aname[20] float income	void register(); void enter_data(); void disp_data(); void have_data(); void give_data(); void getdata(); void putdata();

Question 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();
void Pdisplay();
};
class Marks:private Personal
{
float M{5};
protected:
char Grade[5];
public:
Marks();
void M entry();
void M display 0;
};
class Result:public Marks
{
float Total, Agg;
public:
char FinalGrade, Commence[20];
Result();
void R calculate();
void R display();
};

```

Question 4.1.

Which type of Inheritance is shown in the program?

Answer:

Multilevel inheritance.

Question 4.2.

Specify the visibility mode of base classes.

Answer:

Private for personal public for marks.

Question 4.3.

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

Answer:

Personal (); // constructor of class personal

Marks (); // constructor of class marks

Result (); // destructor of class Result

Marks (); // destructor of class marks

Personal (); // destructor of class personal

Question 4.4.

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

Answer:

Base class of marks: → Personal

Base class of Result: → Marks

Derived classes: → Marks and Results

Question 4.5.

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

(a) Personal

(b) Marks

(c) Result

Answer:

Personal		Marks		Result	
<u>Data member</u>	<u>Bytes</u>	<u>Data member</u>	<u>Bytes</u>	<u>Data member</u>	<u>Bytes</u>
int class:	4	int class:	4	int class:	4
int Rno:	4	int Rno:	4	int Rno:	4
char section	1	char section	1	char section	1
char name[20];	20	char name[20];	20	char name[20];	20
		float M[5];	20	float M[5];	20
		char Grade[5];	5	char Grade[5];	5
				float total	4
Total	29	Total	54		
				float Agg	4
				char Final Grade	1
				commence[20];	20
				Total	83

Question 4.6.

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

Answer:

Data members accessible are:

1. The private, public, and protected data members of personal cannot be accessed.
2. The public data members of marks: Nil (i/7) Public data members of Result char Final Grade; char commence [20];

Question 4.7.

Write the names of all member functions accessible from the object of class Result. Member functions accessible are:

Answer:

1. The private, public and protected member functions of personal cannot be accessed.

2. The public member functions of marks

void mentry();

void

mdisplay();

3. Public member functions of Result

void Rcalculate();

void Rdisplay();

Question 4.8.

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

Answer:

Data members accessible

char Grade[5];

float Total;

float Age;

char Final Grade;

char Commence[20];

Member functions accessible

void mentry();

void mdisplay();

void Rcalculate();

void Rdisplay();

Question 5.

Write the output of the following program

Answer:

```
#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 B
X = 30 Y = 20
Bye Bye
Bye

Question 6.
Debug the following program.

Answer:
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
    void func()
    {
        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();
}
#include<iostream.h>    // % replaced by #, () replaced with <>
#include<conio.h>        // optional
using namespace std;    // missing statement
class A                  // 'C' replaced by 'c'
{
public:                  // ';' replaced by ':'
int a1, a2, a3;          // ';' replaced by ','
void getdata()           // [] replaced by ()
{
    a1=15;
    a2=14; a3=13;        // a2 value is 14
};                        // ';' added
class B; public A         // 'C' replaced by 'c' and () removed
{
public:
void func()              // space added
{
    int b1, b2, b3;      // ';' replaced by ','
    A::getdata();        // [] replaced by ()
    b1=a1;
    b2=a2;
    a3=a3;
    cout<<b1<< '\n'<<b2<< '\n'<<b3;    // line breaks
}
};                        // missing };
int main()               // void replaced by int
{
    system ("cls");       // command to clear screen
    B der;

    der.func();
    cin.get();            // to pause the output
    return 0;             // main () returns 0
}

```