

Chapter-9  
Inheritance

Examples:

**1.[Pg-197]**

```
# include<iostream.h>
# include<conio.h>
class add
{
int sum;
protected:
int num1,num2;
public:
add()
{
num1=num2=sum=0;
cout<<"\n Add constructor...";
}
accept()
{
cout<<"\n Enter two numbers...";
cin>>num1>>num2;
}
plus()
{
sum=num1+num2;
cout<<"\n The sum of two numbers
is..."
<<sum;
}
};
class subtract : public add
{
int sub;
public:
subtract()
{
sub=0;
cout<<"\n Subtract constructor...";
}
minus()
{
add::accept();
sub=num1-num2;
cout<<"\n The difference of two
numbers are..."<<sub;
}
};
void main()
{
clrscr();
```

```
subtract s;
int choice;
cout<<"\n Enter your choice...";
cout<<"\n1.Add...\n2.Subtract...";
cin>>' \n'<<choice;
switch(choice)
{
case 1: s.accept();
s.plus();
break;
case 2: s.minus();
break;
}
}
```

Output:

```
Add constructor...
Subtract constructor...
Enter your choice...
1.Add...
2.Subtract...
1
Enter two numbers...5 6
The sum of two numbers is...11
```

```
Add constructor...
Subtract constructor...
Enter your choice...
1.Add...
2.Subtract...
2
Enter two numbers...6 5
The difference of two numbers is...1
```

**2.[Pg-200]**

```
class add
{
private:
int sum;
protected:
int num1,num2;
public:
add();
accept();
plus();
};
class subtract : private add
{
int sub;
public:
subtract();
minus();
```

```
};
```

### **The data members and member functions**

**inherited by subtract are:**

Data Members:

int num1 & num2

Member functions:

accept() , plus()

**Constructors executed are:**

<u>Classes</u>	<u>Data members</u>	<u>Methods</u>
add	int sum , int num1 & num2	accept() , plus()
subtract	int num1 & num2, int sub	subtract(), minus()

### **3.[Pg-205]**

```
# include<iostream.h>
# include<conio.h>
class base
{
public:
base()
{
cout<<"\nConstructor of base
class...";
}
~base()
{
cout<<"\nDestructor of base
class...";
}
};
class derived:public base
{
public:
derived()
{
cout<<"\nConstructor of derived...";
}
~derived()
{
cout<<"\nDestructor of derived...";
}
};
class derived2:public base
{
public:
```

```
derived()
{
cout<<"\nConstructor of
derived2...";
}
~derived()
{
cout<<"\nDestructor of derived2...";
}
};
```

```
void main()
{
derived2 x;
}
```

### **Output:**

```
Constructor of base class...
Constructor of derived...
Constructor of derived2...
Destructor of derived2...
Destructor of derived...
Destructor of base class...
```

### **Exercise:**

#### **1.[Pg-206,207]**

```
class node
{
int x;
void init();
public:
void read();
protected:
void get();
};
class type:public node
{
int a;
public:
void func1();
protected:
int b;
void getb();
}
class statement:private type
{
int p;
public:
void func2();
protected:
```

```
void func3();
};
```

```
class B:public A
{
public:
void func()
{
int b1,b2,b3;
b1=a1;
b2=a2;
b3=a3;
};
};
void main()
{
B der;
der.a3=0;
der.func();
}
```

a1 cannot be assigned to b1 since a1 is the private data.

Cannot create an object because a3 is protected member.

Complete the following table:

Members of the class type	Accessibility of members/their classes		
	Private	Protected	Public
Members inherited by class type	x,void init()	void getb()	void read()
Defined in class type	a	b,void getb()	void func1()

Members of the class statement	Accessibility of members/their classes		
	Private	Protected	Public
Members inherited by class statement		void getb(),b	void func1()
Defined in class statement	p	void func3()	void func2()

Objects	Can access members	
	Data members	Member functions
Class type		void func1(), void read()
Class statement		void func1(), void func2()

3. Answer the question below: [Pg-209]

```
class vehicle
{
int wheels;
public:
void inputdata(int,int);
void outputdata();
protected:
int passenger;
};
class heavy_vehicle:protected vehicle
{
int diesel_petrol;
protected:
int load;
public:
void readdata(int,int);
void writedata();
};
class bus:private_heavy_vehicle
{
char marks[20];
public:
void fetchdata(char);
void displaydata()
};
```

a) Name the base class and derived class of the class heavy\_vehicle:

Base class = vehicle  
Derived class = bus

2. Find errors: [Pg-208]

```
# include<iostream.h>
class A
{
private:
int a1;
public:
int a2;
protected:
int a3;
};
```

- ***b) Name the data members that can be accessed from the function displaydata():***
  - load , passenger , marks[20]
- ***c) Name the data members that can be accessed by an object of bus class:***
  - marks[20]
- ***d) Is the member function output data accessible to the objects of heavy\_vehicle class:***
  - Yes. Since it is public

OSS St. Paul's