**Inheritance**

// C++ program to demonstrate implementation

// of Inheritance

#include <bits/stdc++.h>

using namespace std**;**

//Base class

class Parent

**{**

public**:**

int id\_p**;**

**};**

// Sub class inheriting from Base Class(Parent)

class Child **:** public Parent

**{**

public**:**

int id\_c**;**

**};**

//main function

int main**()**

**{**

Child obj1**;**

// An object of class child has all data members

// and member functions of class parent

obj1**.**id\_c **=** 7**;**

obj1**.**id\_p **=** 91**;**

cout **<<** "Child id is " **<<** obj1**.**id\_c **<<** endl**;**

cout **<<** "Parent id is " **<<** obj1**.**id\_p **<<** endl**;**

**return** 0**;**

**}**

//Single inheritance

#include<iostream>

using namespace std**;**

class Base

**{**

**int** a**;**

protected**:**

**int** b**;**

public**:**

**int** c**;**

**void** get**(int** x**)**

**{**

a**=**x**;**

**}**

**void** show**()**

**{**

cout**<<**a**;**

**}**

**};**

class der**:**public Base

**{**

**int** aa**;**

protected**:**

**int** bb**;**

public**:**

**int** cc**;**

**void** take**(int** x**,int** y**,int** z**)**

**{**

aa**=**x**;**

bb=y**;**

b**=**z**;**

**}**

**void** disp**()**

**{**

cout**<<**aa**<<**bb**<<**b5**;**

**}**

**};**

**int** main**()**

**{**

der d**;**

d**.**get**(**10**);**

d**.**take**(**20**,**30**,**40**);**

d**.**c**=**50**;**

d**.**cc**=**60**;**

d**.**show**();**

d**.**disp**();**

cout**<<**d**.**c**<<**d**.**cc**;**

**}**

**//op=1020304050**

//Single inheritance

#include<iostream>

using namespace std**;**

class base

**{**

**int** a**;**

protected**:**

**int** b**;**

public**:**

**int** c**;**

**void** get**(int** x**)**

**{**

a**=**x**;**

**}**

**void** show**()**

**{**

cout**<<**a**;**

**}**

**};**

class der**:**private base

**{**

**int** d**;**

protected**:**

**int** e**;**

public**:**

**int** f**;**

**void** take**(int** x1**,int** x2**,int** x3**,int** x4**,int** x5**)**

**{**

get**(**x1**);**

d**=**x2**;**

b**=**x3**;**

e**=**x4**;**

c**=**x5**;**

**}**

**void** display**()**

**{**

show**();**

cout**<<**d**<<**b**<<**e**<<**c**;**

**}**

**};**

**int** main**()**

**{**

der obj**;**

obj**.**take**(**10**,**20**,**30**,**40**,**50**);**

obj**.**display**();**

obj**.**f**=**60**;**

cout**<<**obj**.**f**;**

**}**

***//Privately inheritance***

#include<iostream>

using namespace std**;**

class base **{**

private **:**

**int** a**;**

protected **:**

**int** b**;**

public **:**

**int** c**;**

**void** get**(int** x**){**

a**=**x**;**

**}**

***// protected:***

**void** show**(){**

cout**<<**a**;**

**}**

**};**

class der**:** private base**{**

private**:**

**int** aa**;**

protected**:**

**int** bb**;**

public **:**

**int** cc**;**

**void** take**(int** x**,int** y**,int** z**,int** w**,int** v**){**

get**(**x**);**

aa**=**y**;**

bb**=**z**;**

b**=**w**;**

c**=**v**;**

**}**

**void** disp**()**

**{**

show**();**

cout**<<**aa**<<**bb**<<**b**<<**c**;**

**}**

**};**

**int** main**()**

**{**

der d**;**

d**.**take**(**10**,**20**,**30**,**40**,**50**);**

d**.**cc**=**60**;**

***//d.show();***

d**.**disp**();**

cout**<<**d**.**cc**;**

**}**

***//Multiple inhertance proper***

#include<iostream>

using namespace std**;**

class a**{**

public**:**

**int** aa**;**

**};**

class b**{**

public**:**

**int** bb**;**

**};**

class great **:** public a **,** public b **{**

public**:**

**void** get**(int** x**,int** y**)**

**{**

aa**=**x**;**

bb**=**y**;**

**}**

**void** max**()**

**{**

**if(**aa**>**bb**)**

**{**

cout**<<**aa**<<**" is greater than "**<<**bb**;**

**}**

**if(**bb**>**aa**)**

**{**

cout**<<**bb**<<**" is greater than "**<<**aa**;**

**}**

**}**

**};**

**int** main**()**

**{**

great g**;**

g**.**get**(**20**,**60**);**

g**.**max**();**

**}**

***//Multiple inhertance proper***

#include<iostream>

using namespace std**;**

class a**{**

protected**:**

**int** aa**;**

public**:**

**void** get1**(int** x**)**

**{**

aa**=**x**;**

**}**

**};**

class b**{**

protected**:**

**int** bb**;**

public**:**

**void** get2**(int** x**)**

**{**

bb**=**x**;**

**}**

**};**

class great **:** public a**,** public b**{**

public**:**

**void** greatest**()**

**{**

**if(**aa**>**bb**)**

**{**

cout**<<**aa**<<** " is greater than "**<<**bb**;**

**}**

**if(**bb**>**aa**)**

**{**

cout**<<**bb**<<**" is greater than "**<<**aa**;**

**}**

**}**

**};**

**int** main**()**

**{**

great g**;**

g**.**get1**(**4**);**

g**.**get2**(**5**);**

g**.**greatest**();**

**}**

***//Hierarical inheritance***

#include<iostream>

using namespace std**;**

class base**{**

public**:** **int** a**,**b**;**

**void** get**(int** x**,int** y**)**

**{**

a**=**x**;**

b**=**y**;**

**}**

**};**

class add **:** public base**{**

public**:**

**void** addition**()**

**{**

cout**<<**"\nAddition is "**<<**a**+**b**;**

**}**

**};**

class sub **:** public base**{**

public**:**

**void** subtraction**()**

**{**

cout**<<**"\nSubtraction is "**<<**a**-**b**;**

**}**

**};**

class div **:** public base**{**

public**:**

**float** result**;**

**void** division**()**

**{**

result**=**a**/**b**;**

cout**<<**"\nDivision is "**<<**result**;**

**}**

**};**

class mul **:** public base**{**

public**:**

**void** multiplication**()**

**{**

cout**<<**"\nMultiplication is "**<<**a**\***b**;**

**}**

**};**

class rem **:** public base**{**

public**:**

**float** rem**;**

**void** remainder**()**

**{**

rem**=**a**%**b**;**

cout**<<**"\nReminder is "**<<**rem**;**

**}**

**};**

**int** main**(){**

add x1**;**

x1**.**get**(**10**,**5**);**

x1**.**addition**();**

sub x2**;**

x2**.**get**(**10**,**5**);**

x2**.**subtraction**();**

div x3**;**

x3**.**get**(**10**,**9**);**

x3**.**division**();**

mul x4**;**

x4**.**get**(**10**,**5**);**

x4**.**multiplication**();**

rem x5**;**

x5**.**get**(**10**,**9**);**

x5**.**remainder**();**

**}**

***//Hybrid inheritance***

#include<iostream>

using namespace std**;**

class principle**{**

protected**:**

**int** p**;**

public**:**

**void** take1**(int** x**)**

**{**

p**=**x**;**

**}**

**};**

class rate**:**public principle

**{**

protected**:**

**int** r**;**

public**:**

**void** take2**(int** x**)**

**{**

r**=**x**;**

**}**

**};**

class time**{**

protected**:**

**int** t**;**

public**:**

**void** take3**(int** x**)**

**{**

t**=**x**;**

**}**

**};**

class simple **:** public rate**,**public time

**{**

public**:**

**int** sim**;**

**void** interest**()**

**{**

sim**=**p**+**r**+**t**/**100**;**

cout**<<**"Simple interest is : "**<<**sim**;**

**}**

**};**

**int** main**()**

**{**

simple s**;**

s**.**take1**(**5**);**

s**.**take2**(**3**);**

s**.**take3**(**9**);**

s**.**interest**();**

**}**