**Content 35**

Single Inheritance Deep Dive: Examples + Code

#include <iostream>

using namespace std;

class base

{

    int data1;

public:

    int data2;

    void setdata();

    int getdata1();

    int getdata2();

};

void base::setdata(void)

{

    data1 = 4;

    data2 = 3;

}

int base::getdata1()

{

    return data1;

}

int base::getdata2()

{

    return data2;

}

class derived : public base

{

public:

    int data3;

    void process();

    void display();

};

void derived ::process()

{

    data3 = 10 \* data2;

}

void derived::display()

{

    cout << "The value of data 1:  " << getdata1() << endl;

    cout << "The value of data 2: " << data2 << endl;

    cout << "The value of data 3: " << data3 << endl;

}

int main()

{

    derived d1;

    d1.setdata();

    d1.process();

    d1.display();

    return 0;

}

**Output:**

The value of data 1: 4

The value of data 2: 3

The value of data 3: 30

**Code when mode of derived class is set as private:**

// when mode is set as private

#include <iostream>

using namespace std;

class base

{

    int data1;

public:

    int data2;

    void setdata();

    int getdata1();

    int getdata2();

};

void base::setdata(void)

{

    data1 = 4;

    data2 = 3;

}

int base::getdata1()

{

    return data1;

}

int base::getdata2()

{

    return data2;

}

class derived : private base

{

public:

    int data3;

    void process();

    void display();

};

void derived ::process()

{   setdata();          //here used it becoz I can't use it later the new class had set as private.

    data3 = 10 \* data2;

}

void derived::display()

{

    cout << "The value of data 1:  " << getdata1() << endl;

    cout << "The value of data 2: " << data2 << endl;

    cout << "The value of data 3: " << data3 << endl;

}

int main()

{

    derived d1;

    // d1.setdata();

    d1.process();

    d1.display();

    return 0;

}

**Output will same as above..**