



OOP

## Modern College of Engineering

Shivajinagar, Pune 5. Roll no: 21027

Name: Mahesh Jagtap

### Assignment no. 3

Title: Creating a class which uses the concept of inheritance, displays data & data members & uses the concept of exception handling.

#### Problem statement:

Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores a title & price of publication. From this class derive two classes: book which adds a page count & tape which adds playing time in minute. Write a program that instantiates the book & tape class, allows user to enter data & display data members. If an exception is caught, replace all the data members values with zero values.

Prerequisites: Object oriented programming, class & objects, inheritance, exception handling.

#### objective:

To learn the concept of inheritance & exception handling.



# Modern College of Engineering

Shivajinagar, Pune 5.

(2)

## Theory:

### Inheritance:

- Inheritance in oop can be described as a process of creating new classes from existing classes. New classes inherit some of the properties & behaviour of the existing classes. An existing class that is 'parent' of new class is called a base class. New class that inherits properties of base class is called derived class. Inheritance is a technique of code reuse. It also provides possibility to extend existing classes by creating derived classes.

The basic syntax of inheritance is :

class derived class : access Specifier Base class

There are 3 access specifiers :

- (1) public
- (2) private
- (3) protected



# Modern College of Engineering

Shivajinagar, Pune 5.

## ① public:

This inheritance mode is used mostly. In this, protected member of base class becomes protected members of derived class & public becomes public.

## ② protected:

In this, public & protected members of base class becomes protected members of derived class.

## ③ private:

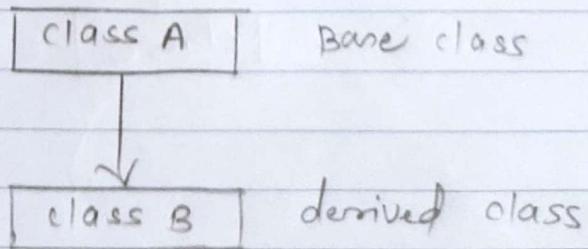
In this mode, public & protected members of base class becomes private members of derived class.

### Types of inheritance:

- ① single      ② multiple      ③ multilevel
- ④ hybrid      ⑤ Hierarchical.

## ① single inheritance:

In this type, one derived class inherits from only one base class. It is most simplest form of inheritance.





# Modern College of Engineering

Shivajinagar, Pune 5.

Syntax :

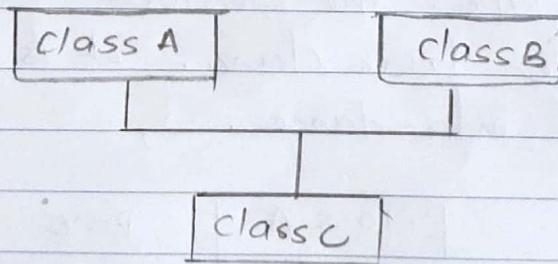
```
class derived-class-name :: Visibility base-class  
mode name  
{  
    // body of derived class  
}
```

## ② Multiple Inheritance :

In this type of inheritance a single derived class may inherit from two or more than two base classes.

Syntax :

```
class derived-class-name :: access spe. base-class1,  
access spe. base-class2  
{  
    // body of derived  
    class  
}
```





# Modern College of Engineering

Shivajinagar, Pune 5.

(5)

e.g.

class A

```
{  
} - - - -  
};
```

class B

```
{  
} - - -  
};
```

class C: public A, public B

```
{  
} - - - -  
};
```

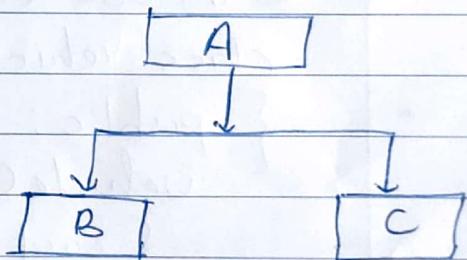
### 3] Hierarchical Inheritance

In this type, multiple derived classes inherits from a single base class.

Syntax:

class A

```
{  
} - - -  
};
```



class B : access A

```
{  
} - - -  
};
```

class C : access S A

```
{  
} - - -  
};
```



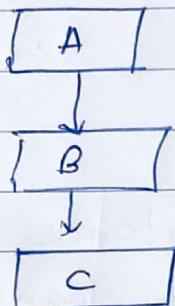
(6)

## Modern College of Engineering

Shivajinagar, Pune 5.

### a) multilevel

In this type of inheritance the derived class inherits from a class, which in turn inherits from some other class. The super class for one, is sub class for others.



// Multilevel inheritance:

```
# include <iostream>
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle" << endl;
    }
};
```

```
class fourwheeler : public vehicle
```

```
{
public:
    fourwheeler()
    {
        cout << " objects with 4 wheels are
                vehicles" << endl;
    }
};
```

①



# Modern College of Engineering

Shivajinagar, Pune 5.

class car : public - four Wheeler

{ public :  
car() }

{ cout << "Car has 4 wheels" << endl;

} ;

int main()

{ car obj ;  
return () ;  
}

Q/P :

This is a vehicle

Objects with 4 wheels are vehicles

Car has 4 wheels.

## 5) Hybrid inheritance

Hybrid inheritance is a combination of any 2 or more types of inheritance

Syntax :

Class A

{  
};

Class B : public A

{  
};

⑧



# Modern College of Engineering

Shivajinagar, Pune 5.

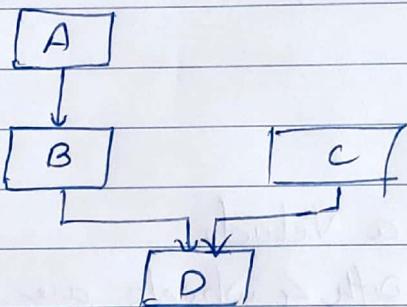
class C

```
{  
---  
};
```

class D : public B, public C

```
{  
---  
};
```

Diagram:



## ⑧ exception handling

- An exception is an unexpected problem that arises during the execution of a program.
- exception handling mechanism provides a way to transfer control from one part of a program to another.
- This makes it easy to separate the error handling code from the code written to handle the actual functionality of the program.
- C++ exception handling is built upon 3 keywords: try, catch, throw.



# Modern College of Engineering

Shivajinagar, Pune 5.

(9)

- try :

A block of code which may cause an exception is typically placed inside the try block. It's followed by one or more catch blocks. If an exception occurs , if is thrown from the try block .

- catch :

This block catches the exception thrown from the try block. Code to handle the exception is written inside this catch block .

- throw :

A program throws an exception when a problem shows up. This is done using throw keyword. every try catch should have a corresponding catch block .

A single try block can have multiple catch block .

Syntax : try - catch

try

{ // protected code

} catch (exceptionName e1)

{  
// catch block

} catch (e2)



## Modern College of Engineering

Shivajinagar, Pune 5.

(10)

```
{// catch block  
}  
} catch (eN)  
{ // catch block  
}
```

D While handling exceptions use the following functionalities:

cin.fail() - This function returns true when an input failure occurs. In this case it would be an input that is not an integer. If cin fails then the input buffer is kept in an error state.

cin.clear() - This is used to clear the error state of the buffer so that further processing of input can take place. This ensures that the input does not lead to an infinite loop of error message display.

cin.ignore() -

This function is used to ignore the rest of the line after the first instance of error that has occurred & it skips to or moves to next line.



# Modern College of Engineering

Shivajinagar, Pune 5.

// exception handling:

int main()

{ int x = -1 ;

cout << "Before try " ;

try {

cout << "Inside try " ;

if (x < 0)

{ throw x ;

cout << "After throw (Never executed) " ;

}

}

catch (int x)

{ cout << "exception caught " ;

}

cout << "After catch (will be executed) " ;

return 0 ;

}

O/P :

Before try

Inside try

Execution caught

After catch (will be executed)

Algorithm :

1. start

2. # create classes publication, book & tape

3. publication class having data member pages &



# Modern College of Engineering

Shivajinagar, Pune 5.

(12)

title & price .

member functions getdata() & putdata()

4. class book having data members pages &

5. class tape having data members minutes &  
member functions getdata() & putdata().

6. create an object bot class book & object  
& of class tape.

7. stop.

Conclusion :

In this assignment we have studied the  
concept of inheritance & exception handling  
in detail & implemented it in c++  
program.

## exception-handling.cpp X

▶ □ ...

D: > Second year study material > Object oriented programming > exception-handling.cpp > ...

```
1 #include<iostream>
2 using namespace std;
3 class publication //Base Class
4 {
5 protected:
6     string title;
7     float price;
8 public:
9     publication() //Default Constructor
10    {
11        title="Hello";
12        price=0;
13    }
14    publication(string t, float p) //Parameterized Constructor
15    {
16        t=title;
17        p=price;
18    }
19    void getdata()
20    {
21        cout<<"Enter the title : ";
22        cin.ignore();
23        getline(cin, title);
24        cout<<"Enter the price(rs.) : ";
25        cin>>price;
26    }
27    void displaydata()
28    {
29        cout<<"Title is : "<<title<<endl ;
30        cout<<"Price is : "<<price<<"rs"<<endl;
31    }
32    ~publication(){}
33}
34
35 class Book:public publication //Derived Class
36 {
37     int pagecount;
38 public:
39     Book() //Default Constructor
40    {
41        pagecount=0;
42    }
43    Book(string t , float p , int pgcnt) :publication(t,p) //ParameterizedConstructor and constructor of base class is being called
44    {
45        pgcnt=pagecount;
46    }
47
48    void getdata01() //Input book data
49    {
50        try
51        {
52            getdata();
53            cout<<"Enter page count:" ;
54            cin>>pagecount;
55            if(pagecount<=0)
56            {
57                throw pagecount;
58            }
59        catch (...)
60        {
61            cout<<"invalid page count";
62        }
63        pagecount=0;
64    }
65 }
```



⊗ 0 △ 0

Ln 34, Col 3 Spaces: 4 UTF-8 CRLF C++ Indents: 0 Win32 ⌂ ⌂

## exception-handling.cpp X

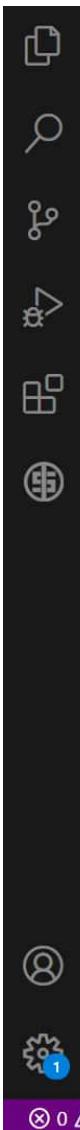
▶ ⏹ ⋮

```
D: > Second year study material > Object oriented programming > exception-handling.cpp > ...
64     |     pagecount=0;
65   }
66 }
67
68 void display01() //Displaybookdata
69 {
70     displaydata();
71     cout<<"No of Pages in book:"<<pagecount<<endl;
72 }
73 ~Book() {};
74
75 class Tape :public publication //Child Class CD from base class
76 {
77     float Playingtime;
78 public:
79     Tape():Playingtime(0) //Default constructor
80     {
81         Playingtime = 0;
82     }
83     Tape(string t, float p, float playtym) : publication(t,p) //ParameterizedConstructor and constructor of base class is being called
84     {
85         playtym=Playingtime;
86     }
87
88 void getdata02() //Input audiocassete data
89 {
90     try {
91         getdata();
92         cout<<"Enter number of minutes of playing:" ;
93         cin>>Playingtime;
94         if Playingtime<=0
95         {
96             throw Playingtime;
97         }
98     catch (...)
99     {
100         cout<<"Invalid playing time";
101         Playingtime=0;
102     }
103
104
105
106
107
108 void display02() //Display cassette data
109 {
110     displaydata();
111     cout<<"Playing time of Cassette:"<<Playingtime<<"min"<<endl;
112 }
113 ~Tape() {};
114
115
116 int main()
117 {
118     int choice,b_count=0,t_count=0;
119     Book bi[10];
120     Tape ti[10];
121     do
122     {
123         cout<<"*****Publication Info*****"<<endl;
124         cout<<"-----Menu-----"<<endl;
125         cout<<"1.Add information to book"<<endl;
126         cout<<"2.Add information to Tape"<<endl;
127         cout<<"3.Display Books information"<<endl;
128         cout<<"4.Display Tapes information"<<endl;
129         cout<<"5.Exit"<<endl;
```



⊗ 0 ▲ 0

Ln 34, Col 3 Spaces: 4 UTF-8 CRLF C++ Indents: 0 Win32 ⌂ ⌂



## exception-handling.cpp X

▶ □ ...

D: > Second year study material > Object oriented programming > exception-handling.cpp > ...

```
116 int main()
117 {
118     int choice,b_count=0,t_count=0;
119     Book b1[10];
120     Tape t1[10];
121     do
122     {
123         cout<<"*****Publication Info*****"<<endl;
124         cout<<"-----Menu-----"<<endl;
125         cout<<"1.Add information to book"<<endl;
126         cout<<"2.Add information to Tape"<<endl;
127         cout<<"3.Display Books information"<<endl;
128         cout<<"4.Display Tapes information"<<endl;
129         cout<<"5.Exit"<<endl;
130         cout<<"Enter choice:";
131         cin>>choice;
132         switch (choice)
133         {
134             case 1:
135                 b1[b_count].getdata01();
136                 b_count++;
137                 break;
138             case 2:
139                 t1[t_count].getdata02();
140                 t_count++;
141                 break;
142             case 3:
143                 cout<<"*****Publication info(Books)*****";
144                 cout<<"\n"; cout<<"\n"; cout<<"\n";
145                 for int i = 0 ; i<b_count;i++
146                 {
147                     b1[i].display01();
148                 }
149                 cout<<"\n"; cout<<"\n"; cout<<"\n";
150             case 4:
151                 cout<<" *****Publication info(Tapes)*****";
152                 cout<<"\n"; cout<<"\n"; cout<<"\n";
153                 for int i = 0 ; i<t_count;i++
154                 {
155                     t1[i].display02();
156                 }
157                 cout<<"\n"; cout<<"\n"; cout<<"\n";
158                 break;
159             case 5:
160                 cout<<"-----Thankyou!-----"<<endl;
161             }
162             cout<<"\n";
163             cout<<"\n";
164             cout<<"\n";
165         } while (choice!=5);
166         cout<<"\n";
167         cout<<"\n";
168     }
169     return 0;
170 }
```

⊗ 0 △ 0

Ln 34, Col 3 Spaces: 4 UTF-8 CRLF C++ Indents: 0 Win32 ⌂ ⌂

The screenshot shows a terminal window in Visual Studio Code displaying the output of a C++ program named exception-handling.cpp. The terminal tab is selected, and the output is as follows:

```
PS C:\Users\Jagta> cd "d:\Second year study material\Object oriented programming\" ; if ($?) { g++ exception-handling.cpp -o exception-handling } ;  
if ($?) { .\exception-handling }  
*****Publication Info*****  
-----Menu-----  
1.Add information to book  
2.Add information to Tape  
3.Display Books information  
4.Display Tapes information  
5.Exit  
Enter choice:1  
Enter the title :mj book  
Enter the price(rs.):250  
Enter page count:380  
*****Publication Info*****  
-----Menu-----  
1.Add information to book  
2.Add information to Tape  
3.Display Books information  
4.Display Tapes information  
5.Exit  
Enter choice:2  
Enter the title :mj tape  
Enter the price(rs.):420  
Enter number of minutes of playing:230  
*****Publication Info*****  
-----Menu-----  
1.Add information to book  
2.Add information to Tape  
3.Display Books information  
4.Display Tapes information  
5.Exit
```

The status bar at the bottom shows the following information: Ln 31, Col 6, Spaces: 4, UTF-8, CRLF, C++, Indents: 1, Win32, 22:20, ENG, 25-11-2020.

File Edit Selection View Go Run Terminal Help exception-handling.cpp - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Code + □ 🗑 ▾ ×

```
Enter the title :mj tape
Enter the price(rs.):420
Enter number of minutes of playing:230
*****Publication Info*****
-----Menu-----
1.Add information to book
2.Add information to Tape
3.Display Books information
4.Display Tapes information
5.Exit
Enter choice:3
*****Publication info(Books)*****

Title is:mj book
Price is:250rs
No of Pages in book:380

*****Publication Info*****
-----Menu-----
1.Add information to book
2.Add information to Tape
3.Display Books information
4.Display Tapes information
5.Exit
Enter choice:4
*****Publication info(Tapes)*****
```

Ln 31, Col 6 Spaces: 4 UTF-8 CRLF C++ Indents: 1 Win32 ⚙ 22:20 Type here to search 22:20 ENG 25-11-2020

File Edit Selection View Go Run Terminal Help exception-handling.cpp - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Code + □ 🗑 ▾ ×

```
No of Pages in book:380

*****Publication Info*****
-----Menu-----
1.Add information to book
2.Add information to Tape
3.Display Books information
4.Display Tapes information
5.Exit
Enter choice:4
*****Publication info(Tapes)*****

Title is:mj tape
Price is:420rs
Playing time of Cassette:230min

*****Publication Info*****
-----Menu-----
1.Add information to book
2.Add information to Tape
3.Display Books information
4.Display Tapes information
5.Exit
Enter choice:5
-----ThankYou!-----
PS D:\Second year study material\Object oriented programming>
```

Ln 31, Col 6 Spaces: 4 UTF-8 CRLF C++ Indents: 1 Win32 ⚙ 22:21 25-11-2020

Type here to search