Assignment 5

Class - SE IV ROLL NO - 21430
Batch - F4
D.O.P - 18/9/2020 D.O.S - 30/9/2020

Title - Creating a class which uses the concept of inheritance, displays data and data members and uses the concept of exception handaling.

Problem Statement - Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publication from this class derive two classes : book which adds a page count (type int) & type which adds a plying time in minute (type float). Write a program that instantiates the book and tape class allows user to enter data and displays data member. If an exception is caught replace all the data member values with zero values.

Objective - To learn concept of inheritance and exception handaling.

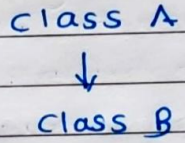
S/W requirements :-
1. Windows / Linux OS.
2. MinGW compiler for windows.
3. Eclipse, Qt creator or any equivalent IDE.

H/W requirements -:
1. Monitor for displaying
2. Windows 10 pc.

Theory -:

1) Inheritance -
Inheritance is a mechanism of acquiring the features and behaviour of a class by another class. The class where members are inherited is called the base class and the class that inherits some members is called derive

Types of Inheritance. -:
i) Single Inheritance -
In this only one class is inherited by base class.



Eg.
```
class A
{

}
class B : public A
{

}
```

2) Multileve Inheritance -:
In this derived class is created from another derived class.

class A:

↓

class B

↓

class C

E.g.

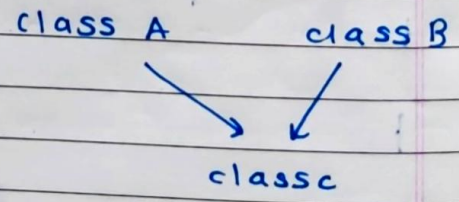
```
class A
{
}
class B :public A
{
}
class C :public B
{
}
```

3. Multiple Inheritance.

In this derived class inherit more than one class.

E.g.

```
class A
{
}
class B
{
}
class C :public A , public B
{
}
```



In above case A and B are both the parent class and C is derived class and it will inherit properties of both A & B.

4. Hierarchical Inheritance.
In this type one class is inherited by more than one class

Eg. class A
{

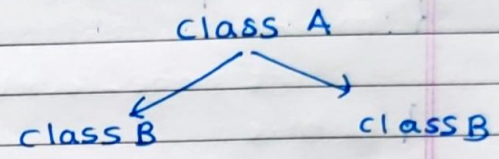
}

class B : public A
{

}

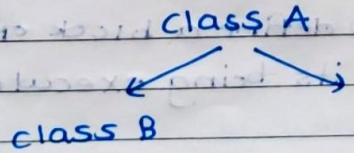
class C : public A
{

}



In this case class A is inherited by class B & C.

5. Hybrid ~~class~~ Inheritance.
This is combination of more than one inheritance it can be combination of Multilevel & multiple inheritance, Hierarchical and multilevel or hierarchical, multilevel and multiple inheritance.



Eg. class A
{

}
class D
{

}
class B : public A
{

}
class C : public A, public D
{

}

In the above example class A, B and C are showing hirarchical inheritance and class D and C are showing single inheritance.

* Exception Handling

Exception handling is a part of c++ object oriented programming it allows to handle the unwanted situation during program execution. In exception handling we use try, catch and throw this keywords.

i) Try statement allows to define a block of code to be tested for error while its being executed.

ii) throw keywords throws an exception when a problem is detected which allows to create custom error.

iii) The catch statement allows to define block of code to be executed if an error occurs in try block.

Syntax:

```
try
{
    // Testing for error or exception
    // throws exception
}
catch()
{
    // Execute that code when exception occurs
}
```

* ADT of class.

```
class publication
{
  protected:
    string name;
    Float price;
  public:
    publication() { // Default constructor }

    publication (string str, Float p)
    {
        // parameterised constructor
    }
```

```
    void getdata()
    {                }
    void display()
    {                }
    ~publication()
    {                }
};
class Book::public publication
{
  private:
    int page_count;
  public:
    Book(){//Default constructor}
    Book(int pg) { // parameterised constructor }
    void getdata(){}
    Void display()
    {              }
    ~Book()
    {        }
};
class Tape:public publication
{
  private:
    float minutes;
  public:
    Tape(){ // Default constructor }
    Tape(float min){ // parameterized constructor}
    void getdata()
    {        }
    void display()
    {        }
    ~Tape(){ // Destructor }
};
```

* Psuedocode

1. Start
2. initialize object of Book and Tape class.
3. call Book object.getdata()
4. call Tape-object.getdata()
5. Display book information
6. Display Tape information.
7. END.

* Test cases

	Description	I/P	Expected O/P	Actual O/P	Result
1.	Taking data of Tape.	Title-END price-150 Time-20	Title-END Price-150 Time=20 min	Title-END Price-150 Time=20 min	Pass
2.	Taking data of Book	Title-Fire price-270 pages-147	Title-Fire price-270 pages-147	Title-Fire price-270 pages-147	Pass

Conclusion-1

Hence, we have successfully studied Concept of inheritance and exception handling.

```cpp
#include<iostream>
#include<string.h>
using namespace std;
int cb=0,ct=0;
class publication
{
    protected:
        char title[100];
        float price;
    public:
    publication()
    {
        strcpy(title," ");
        price=0;
    }
    void getdata()
    {
        cout<<"Title of Publication : ";
        cin.getline(title,98);
        cout<<"Price : ";
        cin>>price;
        cin.ignore(1);
    }
    ~publication();
};
```

```
25 };
26
27 class tape: protected publication
28 {
29
30     float play_time;
31     public:
32     tape()
33     {
34         play_time=0;
35     }
36     void getdata()
37     {
38         publication::getdata();
39         cout<<"Enter the play time(minutes) : ";
40         cin>>play_time;
41         cout<<endl;
42
43      try
44         {
45             if(price < 0)
46             {
47                 price=0;
48                 throw "Price cannot be neagative";
49
```

```
50          }
51          else if(play_time<1)
52          {
53              play_time=0;
54              throw "play_time cannot be neagative";
55          }
56      }
57      catch (const char msg[] )
58      {
59          cout <<"\nException Caught \n";
60          cout<<"\n"<<msg;
61          strcpy(title," ");
62          ct--;
63      }
64  }
65  void showdata()
66  {
67      cout<<"                    The Publication name is "<<title<<endl;
68      cout<<"                    The price of the tape is "<<price<<endl;
69      cout<<"                    Play time of the tape is "<<play_time<<" min"<<endl;
70      cout<<"\n\n";
71  }
72  ~tape();
73 };
74
```

```
76  class book: protected publication
77  {
78
79      int page;
80      public:
81      book()
82      {
83          page=0;
84      }
85      void getdata()
86      {
87          publication::getdata();
88          cout<<"No.of Pages in the book : ";
89          cin>>page;
90          try
91          {
92              if (price < 0)
93              {
94                  price=0;
95                  throw "Price cannot be neagative";
96
97              }
98              else if(page<1)
99              {
100                 page=0;
```

```
100             page=0;
101             throw "Number of pages cannot be neagative";
102         }
103     }
104     catch (const char msg[] )
105     {
106         cout <<"\nException Caught \n";
107         cout<<"\n"<<msg;
108         strcpy(title," ");
109         cb--;
110     }
111
112 }
113 void showdata()
114 {
115
116     cout<<"                    The Publication name is "<<title<<endl;
117     cout<<"                    The price of the book is "<<price<<endl;
118     cout<<"                    No. of pages is "<<page<<endl;
119     cout<<"\n\n";
120 }
121 ~book();
122 };
123 int main()
124 {
```

```
{
    int bk,tp;
    cout<<"How many Book Do you want to add : ";
    cin>>bk;
    cout<<"\nHow many Tape Do you want to add : ";
    cin>>tp;
    book *b1=new book[bk];
    tape *t1=new tape[tp];
    while(1)
    {
        cout<<"\n-----------Enter choice-----------";
        cout<<"\nEnter 1 to add book data";
        cout<<"\nEnter 2 to add tape data";
        cout<<"\nEnter 3 to disaply all book data";
        cout<<"\nEnter 4 to disaply all tape data";
        cout<<"\nENter 0 to exit";
        cout<<"\n\nchoice : ";
        int ch;
        cin>>ch;
        cin.ignore(1);
        switch(ch)
        {
            case 1:
                {
                    if(cb<bk)
```

```
149                 {
150                     (b1+cb)->getdata();
151                     cb++;
152                 }
153                 else
154                 {
155                     cout<<"\n\n     ! ! ! ! You have Entered all Books data ! ! ! !\n\n";
156                 }
157                 break;
158             }
159         case 2:
160         {
161                 if(ct<tp)
162                 {
163                     (t1+ct)->getdata();
164                     ct++;
165                 }
166                 else
167                 {
168                     cout<<"\n       ! ! ! ! You have Entered all Tapes data ! ! ! ! ";
169                 }
170                 break;
171         }
172         case 3:
173         {
```

```
173     {
174         if(cb==0)
175         {
176             cout<<"\n       No Book data found, enter data first\n";
177         }
178         else
179         {
180             int i;
181             cout<<"\n                          -------- Book Details --------\n";
182             for(i=0;i<cb;i++)
183             {
184                 (b1+i)->showdata();
185             }
186         }
187
188         break;
189     }
190     case 4:
191     {
192         if(ct==0)
193         {
194             cout<<"\n       No Tape data found, enter data first\n";
195         }
196         else
197         {
```

```
197             {
198                 int i;
199                 cout<<"\n                    -------- Tape Details --------\n";
200                 for(i=0;i<ct;i++)
201                 {
202                     (t1+i)->showdata();
203                 }
204 
205             }
206             break;
207 
208         }
209         case 0:
210         {
211             delete b1;
212             delete t1;
213             exit(1);
214             break;
215         }
216         default:
217         {
218             cout<<"\nIn valid Input enter again";
219         }
220     }
221 
```

```
How many Book Do you want to add : 2

How many Tape Do you want to add : 2

----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit

choice : 1
Title of Publication : END
Price : 150
No.of Pages in the book : 147

----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit

choice : 2
Title of Publication : FIRE
Price : 250
Enter the play time(minutes) : 45


----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit

choice : 3

-------- Book Details --------
The Publication name is END
The price of the book is 150
No. of pages is 147


----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit

choice : 4

-------- Tape Details --------
The Publication name is FIRE
The price of the tape is 250
```


F:\Ganesh\DEV CPP\program\oop 4.exe
No.of Pages in the book : 147
----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit
choice : 2
Title of Publication : FIRE
Price : 250
Enter the play time(minutes) : 45
----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit
choice : 3
-------- Book Details --------
The Publication name is END
The price of the book is 150
No. of pages is 147
----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit
choice : 4
-------- Tape Details --------
The Publication name is FIRE
The price of the tape is 250
Play time of the tape is 45 min
----------Enter choice----------
Enter 1 to add book data
Enter 2 to add tape data
Enter 3 to disaply all book data
Enter 4 to disaply all tape data
ENter 0 to exit
choice : 0
