

## PRACTICAL 3

Name: Aryan Mane

Roll no.: S212043

Div: B

Batch: B2

### 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 publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float).

Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero value

```
#include <iostream>
using namespace std;
class publication
{
    string title;
    float price;
public:
    void setdata()
    {
        cout<<"Enter the title: " <<endl;
        getline (cin, title);
        if ( "\0" == title)
            throw string("Error: Title is an empty string.");
        cout<<"Enter the price: " <<endl;
        cin>> price;
        if (price <= 0)
            throw string ("Error: Incorrect Price.");
    }
    void display()
    {
        cout<<"Title is: " << title <<endl;
        cout<<"Price is: " << price <<endl;
    }
    void clear()
    {
        title = "\0";
        price = 0.0;
    }
};
class book : public publication
{
    int pgcount ;
public:
    void setdata()
```

```

{
publication::setdata();
cout<<"Enter the pgcount: " <<endl;
cin>> pgcount;
if (pgcount <= 0)
throw string ("Error: Incorrect Pagecount.");
}
void display()
{
publication::display() ;
cout<< "Number of pages are: " << pgcount <<endl;
}
void clear()
{
publication::clear() ;
pgcount = 0;
}
};class Tape : public publication
{
int ptime;
public:
void setdata()
{
publication::setdata();
cout<<"Enter the playing time: " <<endl;
cin>> ptime;
if(ptime <= 0)
throw string ("Error: Incorrect Playing Time.");
}
void display()
{
publication::display();
cout<< "Playing Time: " << ptime <<endl ;
}
void clear()
{
publication::clear();
ptime = 0;
}
};
int main()
{
book bk;
Tape tp;
string mystr;
try
{
cout<<"Enter Book Details: " <<endl;
bk.setdata();
cout<<"\nBook Details are: " <<endl;

```

```

bk.display();
}
catch(string errmsg)
{
cout<<errmsg<<endl<<endl;
bk.clear();
bk.display();
}
getline (cin, mystr);
try
{
cout<<"\nEnter Tape Details: "<<endl;
tp.setdata();
cout<<"\nTape Details are: "<<endl;
tp.display();
}
catch(string errmsg)
{
cout<<errmsg<<endl<<endl;
tp.clear();
tp.display();
}
return 0;
}

```

**Input /Output: Without expectation**

```
aryan@ubuntu:~$ g++ 1.cpp -o run
```

```
aryan@ubuntu:~$ ./run
```

```
Enter Book Details:
```

```
Enter the title:
```

```
OOP'S
```

```
Enter the price:
```

```
799
```

```
Enter the pgcount:
```

```
452
```

```
Book Details are:
```

```
Title is: OOP'S
```

```
Price is: 799
```

```
Number of pages are: 452
```

```
Enter Tape Details:
```

```
Enter the title:
```

```
Polymorphism
```

```
Enter the price:
```

```
699
```

```
Enter the playing time:
```

```
30.69
```

```
Tape Details are:
```

```
Title is: Polymorphism
```

```
Price is: 699
```

```
Playing Time: 30
```

```
aryan@ubuntu:~$
```