**Modern Education Society’s**

**College of Engineering, Pune**

|  |  |
| --- | --- |
| **NAME OF STUDENT:** Prathamesh Kalyan Sable | **CLASS:** SE Comp 1 |
| **SEMESTER/YEAR:** Sem-3 / 2022-23 | **ROLL NO:** 015 |
| **DATE OF PERFORMANCE:**  / /2022 | **DATE OF SUBMISSION:** / /2022 |
| **EXAMINED BY:** Prof. R. H. Shende | **EXPERIMENT NO: A-3** |

###### TITLE : BOOk & AUDIO CASSETTE

**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 a publication. 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 classes, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.

### **OBJECTIVES:**

1. To understand Inheritance and its types .

### **OUTCOMES:**

1. Develop programming application using object oriented programming language C++.
2. Design and apply OOP principles for effective programming.

**PRE-REQUISITES:**

* 1. Knowledge of class, object.
  2. Knowledge of string functions.

**APPARATUS:**

Working Computer system with g++ installed

**QUESTIONS:**

1. How do the properties of the following two derived classes differ?

a. class D1: private B{//....};

b. class D2: public B{//....};

2. When do we use the protected visibility specifier to a class member?

3. Describe the syntax of single inheritance in C++.

**Source Code:**

#include<iostream>

#include <vector>

using namespace std;

class publication{

    protected:

    string title;

    float price;

    public:

    publication(){

        title = "";

        price = 0;

    }

    publication(string title, float price){

    this-> title = title;

    this-> price = price;

    }

};

class book : public publication{

    private:

    int page\_count;

    public:

    book(){

        page\_count = 0;

    }

    book(string title,float price,int page\_count):publication(title, price){

        this->page\_count = page\_count;

    }

    void set\_data(){

        cout<<"Enter title of the book:";

        cin.ignore();

        getline(cin,title);

        cout<<"Enter price of the book:";

        cin>>price;

        cout<<"Enter page count of the book:";

        cin>>page\_count;

    }

    void display\_data(){

        cout<<"Title of the book is "<<title<<endl;

        cout<<"Price of the book is "<<price<<endl;

        cout<<"Number of pages of the book are "<<page\_count<<endl;

    }

};

class tape : public publication{

    private:

    float playing\_time;

    public:

    tape(){

        playing\_time = 0;

    }

    tape(string title,float price,float playing\_time):publication(title, price){

        this->playing\_time = playing\_time;

    }

    void set\_data(){

        cout<<"Enter title of the tape:";

        cin.ignore();

        getline(cin,title);

        cout<<"Enter price of the tape:";

        cin>>price;

        cout<<"Enter playing time of the tape:";

        cin>>playing\_time;

    }

    void display\_data(){

        cout<<"Title of the tape is "<<title<<endl;

        cout<<"Price of the tape is "<<price<<endl;

        cout<<"Playing time of the tape is "<<playing\_time<<endl;

    }

};

int main(){

    vector<book\*> book\_record;

    vector<tape\*> tape\_record;

    bool loop\_control = true;

    int choice;

    book \*book\_rec;

    tape \*tape\_rec;

    int i;

    while(loop\_control){

        cout<<"\n---MENU---"<<endl;

        cout<<"1. Add new Book"<<endl;

        cout<<"2. Add new Tape"<<endl;

        cout<<"3. Display all Book data"<<endl;

        cout<<"4. Display all Tape data"<<endl;

        cout<<"5. Display count of Books and Tape"<<endl;

        cout<<"6. Exit."<<endl;

        cout<<"Enter your Choice:";

        cin>>choice;

        switch (choice){

            case 1:

                book\_rec = new book;

                book\_rec->set\_data();

                book\_record.push\_back(book\_rec);

                cout<<"Book record added sucessfully :)"<<endl;

                break;

            case 2:

                tape\_rec = new tape;

                tape\_rec->set\_data();

                tape\_record.push\_back(tape\_rec);

                cout<<"Tape record added sucessfully :)"<<endl;

                break;

            case 3:

                for(i=0;i<book\_record.size();i++){

                    cout<<"\nRecord number :"<<i+1<<endl;

                    book\_rec = book\_record.at(i);

                    book\_rec->display\_data();

                }

                break;

            case 4:

                for(i=0;i<tape\_record.size();i++){

                    cout<<"\nRecord number :"<<i+1<<endl;

                    tape\_rec = tape\_record.at(i);

                    tape\_rec->display\_data();

                }

                break;

            case 5:

                cout<<"Total Book records :"<<book\_record.size()<<endl;

                cout<<"Total Tape records :"<<tape\_record.size()<<endl;

                break;

            case 6:

                loop\_control = false;

                cout<<"Thank You"<<endl;

                break;

            default:

                break;

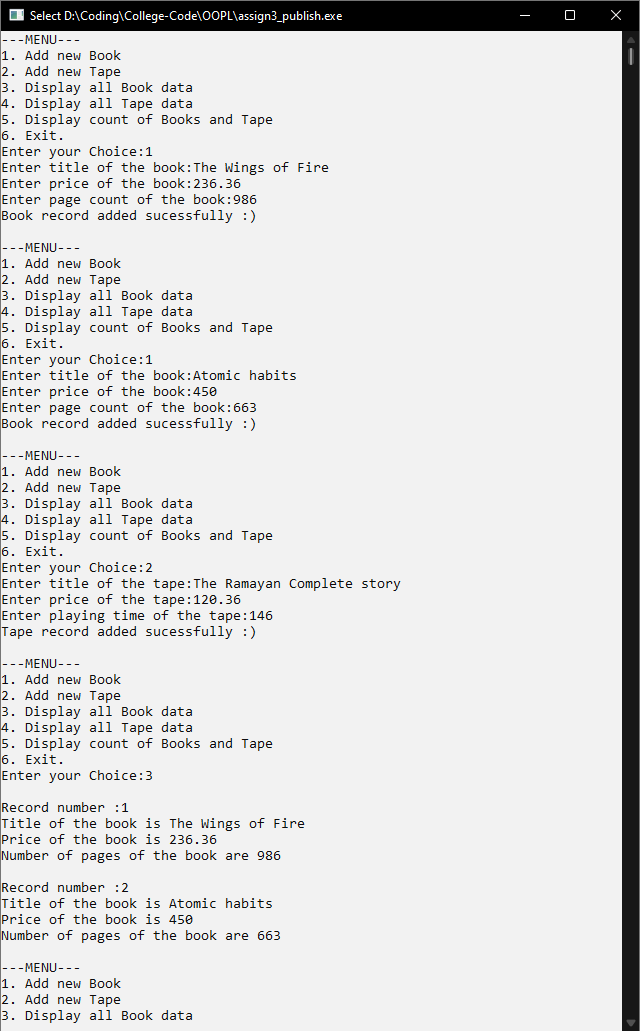
        }

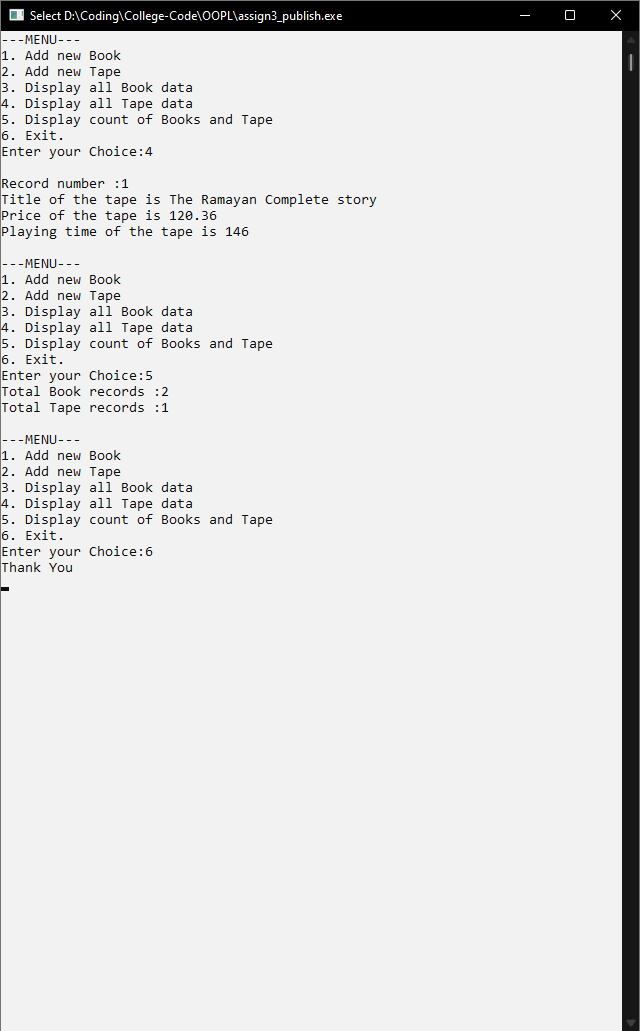
    }

    return 0;

}

**Output:**

****

****