4]Title: Creating a class which uses the concept of inheritance, displays data and data members and 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 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 values.

#include <iostream>

#include <string>

using namespace std;

// Base class

class Publication {

protected:

string title;

float price;

public:

Publication() : title(""), price(0.0) {}

// Method to set publication data

void setData() {

cout << "Enter title: ";

getline(cin >> ws, title); // Discards leading whitespace

cout << "Enter price: ";

cin >> price;

}

// Method to display publication data

void display() {

cout << "Title: " << title << endl;

cout << "Price: " << price << endl;

}

};

// Derived class for books

class Book : public Publication {

private:

int pageCount;

public:

Book() : pageCount(0) {}

// Method to set book-specific data

void setData() {

Publication::setData();

cout << "Enter page count: ";

cin >> pageCount;

}

// Method to display book-specific data

void display() {

Publication::display();

cout << "Page Count: " << pageCount << endl;

}

};

// Derived class for tapes

class Tape : public Publication {

private:

float playingTime;

public:

Tape() : playingTime(0.0) {}

// Method to set tape-specific data

void setData() {

Publication::setData();

cout << "Enter playing time (in minutes): ";

cin >> playingTime;

}

// Method to display tape-specific data

void display() {

Publication::display();

cout << "Playing Time: " << playingTime << " minutes" << endl;

}

};

int main() {

Book book;

Tape tape;

// Input data for book

cout << "Enter details for Book:" << endl;

book.setData();

// Input data for tape

cout << "\nEnter details for Tape:" << endl;

tape.setData();

// Display data for book

cout << "\nDetails of Book:" << endl;

book.display();

// Display data for tape

cout << "\nDetails of Tape:" << endl;

tape.display();

return 0;

}