

```

#include <iostream>
#include <string>
using namespace std;

// Base class Publication
class Publication {
protected:
    string title;
    float price;

public:
    // Default constructor
    Publication() : title(""), price(0.0) {}

    // Function to input data for Publication
    void inputData() {
        cout << "Enter title: ";
        cin.ignore(); // To ignore leftover newline from previous input
        getline(cin, title);
        cout << "Enter price: ";
        cin >> price;
    }

    // Function to display Publication data
    void displayData() {
        cout << "Title: " << title << endl;
        cout << "Price: " << price << endl;
    }
};

// Derived class Book
class Book : public Publication {
private:
    int pageCount;

public:
    // Default constructor
    Book() : pageCount(0) {}

    // Function to input data for Book
    void inputData() {
        Publication::inputData(); // Call the base class function
        cout << "Enter page count: ";
        cin >> pageCount;
    }

    // Function to display Book data
    void displayData() {
        Publication::displayData(); // Call the base class function
        cout << "Page count: " << pageCount << endl;
    }
};

// Derived class Tape
class Tape : public Publication {
private:
    float playingTime;

public:

```

```

// Default constructor
Tape() : playingTime(0.0) {}

// Function to input data for Tape
void inputData() {
    Publication::inputData(); // Call the base class function
    cout << "Enter playing time in minutes: ";
    cin >> playingTime;
}

// Function to display Tape data
void displayData() {
    Publication::displayData(); // Call the base class function
    cout << "Playing time: " << playingTime << " minutes" << endl;
}
};

int main() {
    try {
        Book book;
        Tape tape;

        // Input data for Book
        cout << "Enter data for Book:\n";
        book.inputData();

        // Input data for Tape
        cout << "\nEnter data for Tape:\n";
        tape.inputData();

        // Display data
        cout << "\nDisplaying data for Book:\n";
        book.displayData();

        cout << "\nDisplaying data for Tape:\n";
        tape.displayData();
    }
    catch (...) {
        cout << "Exception caught! Resetting data to zero values." << endl;
        // Reset all data to zero
        Book book;
        Tape tape;
        book.displayData();
        tape.displayData();
    }

    return 0;
}

```