

**DSA (Data Structure and Algorithms) Lab**

**LAB REPORT # 1**

**Semester**: 3rdSemester

**Section**: C

**Submitted To:**

**Abdullah Shahrose**

**Submitted By:**

**Name**: Abdul Ahad

**Roll No**: 22-CS-071

**Task 1:**

#include <iostream>

#include <string>

using namespace std;

struct Course

{

    int Course\_ID;

    string Course\_Name;

    int Marks;

    Course \*Head = this, \*Tail = this, \*Next = NULL;

};

struct Student

{

    int Student\_ID;

    string Student\_Name;

    string Department;

    int Semester;

    int Courses;

    Course course;

    int Marks;

};

void SetData(Student &S)

{

    cout << "\n\nCollecting Student Data" << endl

         << endl;

    cout << "Enter Student ID : ";

    cin >> S.Student\_ID;

    cout << "Enter Student Name : ";

    cin.ignore();

    getline(cin, S.Student\_Name);

    cout << "Enter Department : ";

    getline(cin, S.Student\_Name);

    cout << "Enter Student Semester : ";

    cin >> S.Semester;

    cout << "Enter the number of Courses : ";

    cin >> S.Courses;

    for (int i = 0; i < S.Courses; i++)

    {

        cout << endl

             << endl;

        cout << "Course : " << i + 1 << endl

             << endl;

        cout << "Enter Course ID : ";

        cin >> S.course.Tail->Course\_ID;

        cout << "Enter Course Name : ";

        cin.ignore();

        getline(cin, S.course.Tail->Course\_Name);

        cout << "Enter Course Marks : ";

        cin >> S.course.Tail->Marks;

        S.course.Tail->Next = new Course;

        S.course.Tail = S.course.Tail->Next;

    }

    cout << endl

         << endl;

    cout << "Enter Marks : ";

    cin >> S.Marks;

    cout << endl;

}

void GetData(Student &S)

{

    cout << "\n\nDisplaying Student Data" << endl

         << endl;

    cout << "Student ID : " << S.Student\_ID << endl;

    cout << "Student Name : " << S.Student\_Name << endl;

    cout << "Department : " << S.Student\_Name << endl;

    cout << "Student Semester : " << S.Semester << endl;

    cout << "Number of Courses : " << S.Courses << endl;

    for (int i = 0; i < S.Courses; i++)

    {

        cout << endl

             << endl;

        cout << "Course : " << i + 1 << endl

             << endl;

        cout << "Course ID : " << S.course.Head->Course\_ID << endl;

        cout << "Course Name : " << S.course.Head->Course\_Name << endl;

        cout << "Course Marks : " << S.course.Head->Marks << endl;

        S.course.Head = S.course.Head->Next;

    }

    cout << endl

         << endl;

    cout << "Marks : " << S.Marks << endl

         << endl;

}

int main()

{

    Student S;

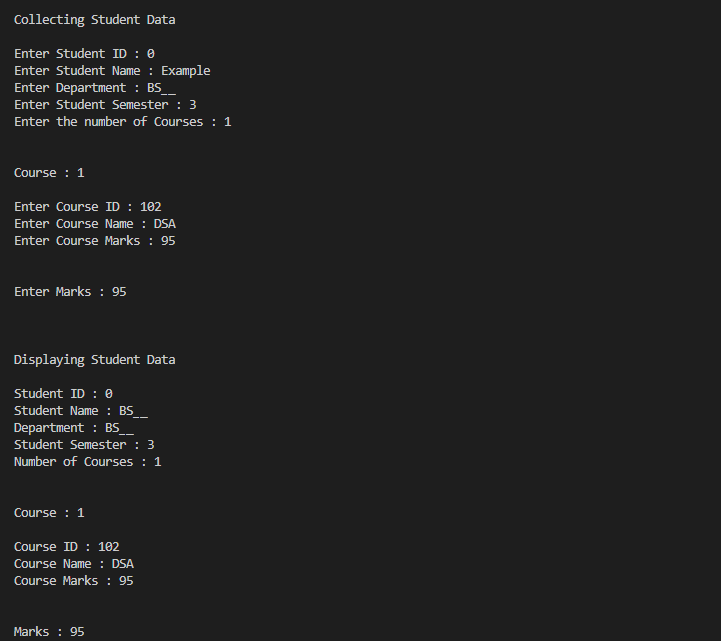
    SetData(S);

    GetData(S);

    return 0;

}

**Output:**

****

**Task 2:**

#include <iostream>

#include <string>

using namespace std;

class publication

{

protected:

    string title;

    float price;

public:

    virtual void getdata() = 0;

    virtual void putdata() = 0;

};

class book : public publication

{

private:

    int page\_count;

public:

    void getdata() override

    {

        cout << "Enter the book title : ";

        cin.ignore();

        getline(cin, this->title);

        cout << "Enter the price : ";

        cin >> this->price;

        cout << "Enter the number of pages : ";

        cin >> this->page\_count;

    }

    void putdata() override

    {

        cout << "\n\nBook title : " << this->title << endl;

        cout << "Price : " << this->price << endl;

        cout << "Number of pages : " << this->page\_count << endl;

        cout << "\n\n";

    }

};

class tape : public publication

{

private:

    float TimeInMinutes;

    void getdata() override

    {

        cout << "Enter the book title : ";

        cin.ignore();

        getline(cin, this->title);

        cout << "Enter the price : ";

        cin >> this->price;

        cout << "Enter the time in minutes : ";

        cin >> this->TimeInMinutes;

    }

    void putdata() override

    {

        cout << "\n\nBook title : " << this->title << endl;

        cout << "Price : " << this->price << endl;

        cout << "Time in minutes : " << this->TimeInMinutes << endl;

        cout << "\n\n";

    }

};

int main()

{

    publication \*p;

    cout << "\n\n\t\tChoose The menu for desired Task : \n";

    cout << "\t\t\tPress 1 to Select a Book\n";

    cout << "\t\t\tPress 2 to Select a Tape\n";

    cout << "\t\t\tPress 0 to exit\n";

choiceP:

    int choice = -1;

    cin >> choice;

    cout << "\n\n";

    switch (choice)

    {

    case 0:

        cout << "Thank you for your time..\nExitnig...\nQUIT\n\n";

        exit(0);

        break;

    case 1:

        p = new book();

        break;

    case 2:

        p = new tape();

        break;

    default:

        cout << "wrong input\n";

        goto choiceP;

    }

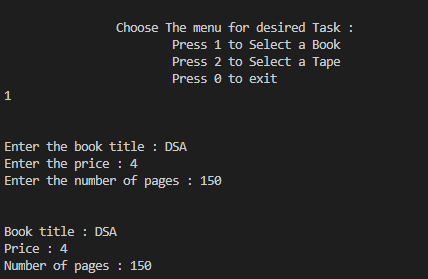
    p->getdata();

    p->putdata();

    return 0;

}

**Output:**

****

**Task 3:**

#include <iostream>

#include <string>

using namespace std;

class publication

{

protected:

    string title;

    float price;

public:

    virtual void getdata() = 0;

    virtual void putdata() = 0;

};

class book : public publication

{

private:

    int page\_count;

public:

    void getdata() override

    {

        cout << "Enter the book title: ";

        cin.ignore();

        getline(cin, this->title);

        cout << "Enter the price: ";

        cin >> this->price;

        cout << "Enter the number of pages: ";

        cin >> this->page\_count;

    }

    void putdata() override

    {

        cout << "\nBook title: " << this->title << endl;

        cout << "Price: " << this->price << endl;

        cout << "Number of pages: " << this->page\_count << endl;

        cout << "\n";

    }

};

class tape : public publication

{

private:

    float TimeInMinutes;

public:

    void getdata() override

    {

        cout << "Enter the tape title: ";

        cin.ignore();

        getline(cin, this->title);

        cout << "Enter the price: ";

        cin >> this->price;

        cout << "Enter the time in minutes: ";

        cin >> this->TimeInMinutes;

    }

    void putdata() override

    {

        cout << "\nTape title: " << this->title << endl;

        cout << "Price: " << this->price << endl;

        cout << "Time in minutes: " << this->TimeInMinutes << endl;

        cout << "\n";

    }

};

int main()

{

    const int MAX\_PUBLICATIONS = 10;

    publication \*pubArray[MAX\_PUBLICATIONS]; // Array of pointers to publication

    cout << "\n\t\tChoose the menu for the desired task:\n";

    cout << "\t\t1. Enter a Book\n";

    cout << "\t\t2. Enter a Tape\n";

    cout << "\t\t0. Exit\n";

    int choice;

    int numPublications = 0; // Keep track of the number of publications

    while (true)

    {

        cout << "Enter your choice: ";

        cin >> choice;

        if (choice == 0)

        {

            cout << "Exiting...\n";

            break; // Exit the loop if the user chooses 0

        }

        else if (choice == 1 || choice == 2)

        {

            if (numPublications < MAX\_PUBLICATIONS)

            {

                if (choice == 1)

                {

                    pubArray[numPublications] = new book();

                }

                else if (choice == 2)

                {

                    pubArray[numPublications] = new tape();

                }

                pubArray[numPublications]->getdata();

                numPublications++;

            }

            else

            {

                cout << "Maximum publications reached.\n";

            }

        }

        else

        {

            cout << "Invalid choice. Please enter 0, 1, or 2.\n";

        }

    }

    // Display the data for all the publications

    for (int i = 0; i < numPublications; i++)

    {

        pubArray[i]->putdata();

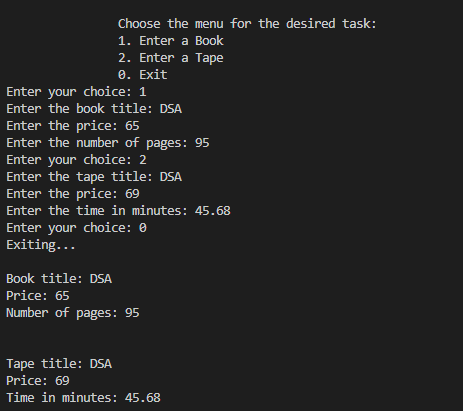
        delete pubArray[i]; // Delete the dynamically allocated objects

    }

    return 0;

}

**Output:**

****