***OOP LAB – 2***

***22108190***

**Program -1**

#include <iostream>

using namespace std ;

struct Date {

int date ;

int month ;

int year ;

};

int main () {

Date DOB ;

cout << "Enter date of the birth " ;

cin >> DOB.date ;

cout << "Enter month of the birth " ;

cin >> DOB.month ;

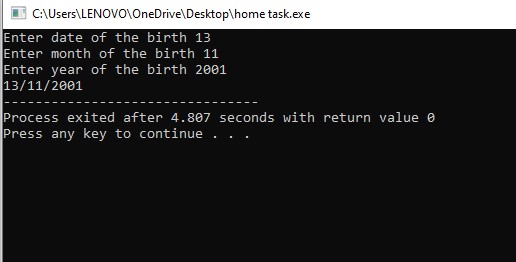
cout << "Enter year of the birth " ;

cin >> DOB.year ;

cout << DOB.date <<"/" << DOB.month << "/" << DOB.year ;

return 0 ;

}



**Program -2 :**

#include <iostream>

using namespace std;

struct Book {

int id;

double price;

int pages;

};

int main() {

Book books[3];

for (int i = 0; i < 3; i++) {

cout << "Enter the data for book " << i + 1 << endl ;

cout << "Enter ID " ;

cin >> books[i].id;

cout << "Enter price ";

cin >> books[i].price;

cout << "Enter pages " ;

cin >> books[i].pages;

}

int mostExpensiveIndex = 0;

for (int i = 1; i < 3; i++) {

if (books[i].price > books[mostExpensiveIndex].price) {

mostExpensiveIndex = i;

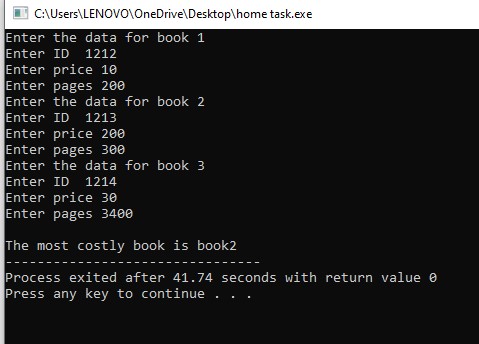
}

}

cout << "\nThe most costly book is book" << mostExpensiveIndex + 1;

return 0;

}



**Program – 3 (Home Task )**

#include <iostream>

using namespace std;

struct Employee {

int employeeCode;

char fullName[50];

int joinYear, joinMonth, joinDay;

};

int main() {

const int maxEmployees = 100;

Employee employees[maxEmployees];

int numEmployees;

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

cin >> numEmployees;

for (int i = 0; i < numEmployees; i++) {

cout << "Enter details for Employee #" << i + 1 << ":" << endl;

cout << "Employee ID: ";

cin >> employees[i].employeeCode;

cout << "Full Name:";

cin.ignore();

cin.getline(employees[i].fullName, 50);

cout << "Date of Joining (dd mm yyyy): ";

cin >> employees[i].joinDay >> employees[i].joinMonth >> employees[i].joinYear;

cout << endl;

}

int currentYear, currentMonth, currentDay;

cout << "Enter the current date (dd mm yyyy): ";

cin >> currentDay >> currentMonth >> currentYear

cout << "\nNames of employees with a tenure of 3 or more years:" << endl;

for (int i = 0; i < numEmployees; i++) {

int tenureYears = currentYear - employees[i].joinYear;

int tenureMonths = currentMonth - employees[i].joinMonth;

int tenureDays = currentDay - employees[i].joinDay;

if (tenureYears > 3 || (tenureYears == 3 && tenureMonths >= 0 && tenureDays >= 0)) {

cout << employees[i].fullName << endl;

}

}

return 0;

}

