**Assignment 3 (OOPs)**

**Name: Yatin Kanwar   
Branch: ECE  
SID: 23105057**

**Q1. C++ Program To enter required number of book records and display them using array declaration of class’s objects.**

#include <iostream>

#include <string>

using namespace std;

class Book {

private:

    string title;

    string author;

    int year;

public:

    void input() {

        cout << "Enter book title: ";

        getline(cin, title);

        cout << "Enter book author: ";

        getline(cin, author);

        cout << "Enter publication year: ";

        cin >> year;

        cin.ignore();

    }

    void display() const {

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

        cout << "Author: " << author << endl;

        cout << "Year: " << year << endl;

    }

};

int main() {

    int n;

    cout << "Enter number of books: ";

    cin >> n;

    cin.ignore();

    Book\* books = new Book[n];

    for (int i = 0; i < n; ++i) {

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

        books[i].input();

    }

    cout << "\nBook Records:" << endl;

    for (int i = 0; i < n; ++i) {

        cout << "Details of book " << i + 1 << ":" << endl;

        books[i].display();

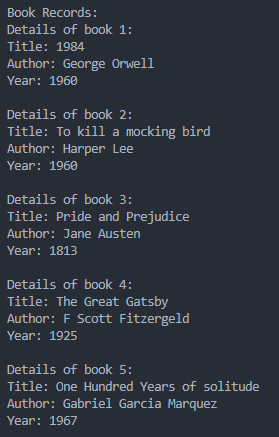
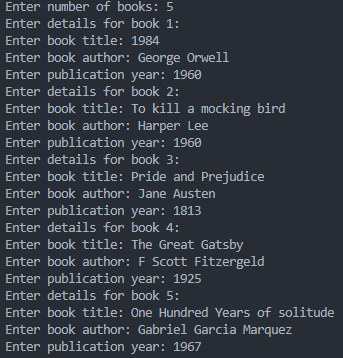
        cout << endl;

    }

    delete[] books;

    return 0;

}

**Output:  
**

**Q2. Program To enter the data of a students and display it using indirection operator(using pointer to class objects) and structure operator.**

#include <iostream>

#include <string>

using namespace std;

class Student {

private:

    string name;

    int rollNo;

public:

    void input() {

        cout << "Enter student name: ";

        getline(cin, name);

        cout << "Enter roll number: ";

        cin >> rollNo;

        cin.ignore();

    }

    void display() const {

        cout << "Name: " << name << endl;

        cout << "Roll Number: " << rollNo << endl;

    }

};

int main() {

    Student student;

    Student\* ptr = &student;

    cout << "Enter student data:" << endl;

    ptr->input();

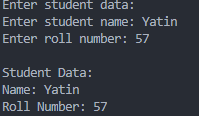
    cout << "\nStudent Data:" << endl;

    ptr->display();

    return 0;

}

**Output:**

****

**Q3. Program To find the highest of two numbers using the nesting of member functions.**

#include <iostream>

using namespace std;

class Number {

private:

    int num1, num2;

    int max() const {

        return (num1 > num2) ? num1 : num2;

    }

public:

    void input() {

        cout << "Enter two numbers: ";

        cin >> num1 >> num2;

    }

    void display() const {

        cout << "The highest number is: " << max() << endl;

    }

};

int main() {

    Number number;

    number.input();

    number.display();

    return 0;

}

**Output:**

****

**Q4. Program using of inline function sum of three numbers.**

#include <iostream>

using namespace std;

class Calculator {

public:

    inline int sum(int *a*, int *b*, int *c*) {

        return a + b + c;

    }

};

int main() {

    Calculator calc;

    int x, y, z;

    cout << "Enter three numbers: ";

    cin >> x >> y >> z;

    cout << "Sum of the three numbers is: " << calc.sum(x, y, z) << endl;

    return 0;

}

**Output:**

****

**Q5. Program to simulate an arithmetic calculator for integer. The program should be able to produce the result calculated and the number of arithmetic operators performed so far. Any wrong operation is to be reported.**

#include <iostream>

using namespace std;

class Calculator {

private:

    int numOperations;

public:

    Calculator() : numOperations(0) {}

    void add(int *a*, int *b*) {

        cout << "Result: " << a + b << endl;

        numOperations++;

    }

    void subtract(int *a*, int *b*) {

        cout << "Result: " << a - b << endl;

        numOperations++;

    }

    void multiply(int *a*, int *b*) {

        cout << "Result: " << a \* b << endl;

        numOperations++;

    }

    void divide(int *a*, int *b*) {

        if (b != 0) {

            cout << "Result: " << a / b << endl;

            numOperations++;

        } else {

            cout << "Error: Division by zero" << endl;

        }

    }

    void showOperations() const {

        cout << "Total operations performed: " << numOperations << endl;

    }

};

int main() {

    Calculator calc;

    int a, b;

    char op;

    while (true) {

        cout << "Enter operation (+, -, \*, /) or q to quit: ";

        cin >> op;

        if (op == 'q') break;

        cout << "Enter two integers: ";

        cin >> a >> b;

        switch (op) {

            case '+': calc.add(a, b); break;

            case '-': calc.subtract(a, b); break;

            case '\*': calc.multiply(a, b); break;

            case '/': calc.divide(a, b); break;

            default: cout << "Invalid operation" << endl;

        }

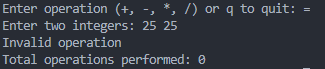
        calc.showOperations();

    }

    return 0;

}

**Output:**

****

**Q6. Program Printing simple and Inverted Pyramid pattern using for loop.**

#include <iostream>

using namespace std;

void printPyramid(int *n*) {

    cout << "Simple Pyramid:" << endl;

    for (int i = 1; i <= n; ++i) {

        for (int j = i; j < n; ++j) cout << " ";

        for (int k = 1; k <= (2 \* i - 1); ++k) cout << "\*";

        cout << endl;

    }

    cout << "\nInverted Pyramid:" << endl;

    for (int i = n; i >= 1; --i) {

        for (int j = n; j > i; --j) cout << " ";

        for (int k = 1; k <= (2 \* i - 1); ++k) cout << "\*";

        cout << endl;

    }

}

int main() {

    int n;

    cout << "Enter the number of rows for the pyramid: ";

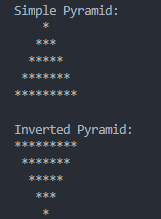
    cin >> n;

    printPyramid(n);

    return 0;

}

**Output:**

****

**Q7. Program Operator Precedence and Associativity in C++.**

#include <iostream>

using namespace std;

int main() {

    int a = 5, b = 10, c = 15;

    int result;

    result = a + b \* c;

    cout << "a + b \* c = " << result << endl;

    result = (a + b) \* c;

    cout << "(a + b) \* c = " << result << endl;

    result = a - b + c;

    cout << "a - b + c = " << result << endl;

    result = a - (b + c);

    cout << "a - (b + c) = " << result << endl;

    result = a \* b / c;

    cout << "a \* b / c = " << result << endl;

    result = a / b \* c;

    cout << "a / b \* c = " << result << endl;

    return 0;

}

**Output:**