Program 1: BigInt class with overloaded relational operators

#include <iostream>

using namespace std;

class BigInt {

private:

int num;

public:

BigInt(int n) : num(n) {}

bool operator==(const BigInt& other) const {

return num == other.num;

}

bool operator!=(const BigInt& other) const {

return !(\*this == other);

}

bool operator<(const BigInt& other) const {

return num < other.num;

}

bool operator<=(const BigInt& other) const {

return (\*this < other) || (\*this == other);

}

bool operator>(const BigInt& other) const {

return !(\*this <= other);

}

bool operator>=(const BigInt& other) const {

return !(\*this < other);

}

};

int main() {

BigInt num1(10);

BigInt num2(20);

if (num1 == num2) {

cout << "num1 is equal to num2" << endl;

}

else if (num1 < num2) {

cout << "num1 is less than num2" << endl;

}

else {

cout << "num1 is greater than num2" << endl;

}

return 0;

}

**Output Snip:**

**Program 2: Date class with overloaded relational operators**

#include <iostream>

using namespace std;

class Date {

int day;

int month;

int year;

public:

Date() : day(0), month(0), year(0) {}

Date(int d, int m, int y) : day(d), month(m), year(y) {}

void Read() {

cout << "Enter day, month, and year: ";

cin >> day >> month >> year;

}

void Write() {

cout << day << "/" << month << "/" << year;

}

bool operator==(const Date& obj2) const {

return day == obj2.day && month == obj2.month && year == obj2.year;

}

bool operator!=(const Date& obj2) const {

return !(\*this == obj2);

}

bool operator<(const Date& obj2) const {

if (year < obj2.year)

return true;

else if (year == obj2.year) {

if (month < obj2.month)

return true;

else if (month == obj2.month) {

return day < obj2.day;

}

}

return false;

}

bool operator<=(const Date& obj2) const {

return (\*this < obj2) || (\*this == obj2);

}

bool operator>(const Date& obj2) const {

return !(\*this <= obj2);

}

bool operator>=(const Date& obj2) const {

return !(\*this < obj2);

}

};

int main() {

Date date1(5, 9, 2023);

Date date2;

date2.Read();

if (date1 == date2) {

cout << "Date 1 is equal to Date 2" << endl;

}

else if (date1 < date2) {

cout << "Date 1 is earlier than Date 2" << endl;

}

else {

cout << "Date 1 is later than Date 2" << endl;

}

return 0;

}

**Output Snip:**

**Program 3: String class with overloaded relational operators**

#include <iostream>

#include <cstring>

using namespace std;

class String {

private:

char\* str;

public:

String(const char\* s) {

str = new char[strlen(s) + 1];

strcpy(str, s);

}

~String() {

delete[] str;

}

const char\* c\_str() const {

return str;

}

bool operator==(const String& other) const {

return strcmp(str, other.str) == 0;

}

bool operator!=(const String& other) const {

return !(\*this == other);

}

bool operator<(const String& other) const {

return strcmp(str, other.str) < 0;

}

bool operator<=(const String& other) const {

return (\*this < other) || (\*this == other);

}

bool operator>(const String& other) const {

return !(\*this <= other);

}

bool operator>=(const String& other) const {

return !(\*this < other);

}

};

int main() {

String s1("Hello");

String s2("World");

String s3("Hello");

if (s1 == s2) {

cout << "s1 is equal to s2" << endl;

}

else if (s1 < s2) {

cout << "s1 is less than s2" << endl;

}

else {

cout << "s1 is greater than s2" << endl;

}

if (s1 == s3) {

cout << "s1 is equal to s3" << endl;

}

else {

cout << "s1 is not equal to s3" << endl;

}

return 0;

}

**Output Snip:**