```
//============================================================================
// Name        : clocktype.cpp
// Author      : Saranya Balakrishnan
// G.I.D       : 5
// Depart      : Computer Engineering
// SJSU ID     : 013786097
//============================================================================
```

1. Implement the class clock type described in the slides, use the same const modifier for all the methods FOR THE METHODS THAT HAVE THE CONST MODIFIER, the ones that don't DO NOT ADD it

```
#include <iostream>
using namespace std;
class clockType
{
public:
	void setTime(int hours, int minutes, int seconds);
	void getTime(int& hours,int& minutes, int& seconds) const;
	void printTime() const;
	void incrementSeconds();
	void incrementMinutes();
	void incrementHours();
	bool equalTime(const clockType& otherClock) const;
	clockType(int hours, int minutes, int seconds);
	clockType();
private:
	int hr;
	int sec;
	int min;
};

//Member function definitions
void clockType::setTime(int hours, int minutes, int seconds)
{
	if (0 <= hours && hours < 24)
		hr = hours;
	else
	hr = 0;
	if (0 <= minutes && minutes < 60)
		min = minutes;
	else
	min = 0;
	if (0 <= seconds && seconds < 60)
		sec = seconds;
	else
	sec = 0;
}

void clockType::getTime(int& hours, int& minutes, int& seconds) const
{
	hours = hr;
	minutes = min;
	seconds = sec;
```

```
}


void clockType::printTime() const
{
       if (hr < 10)
              cout << "0";
       cout << hr << ":";
       if (min < 10)
              cout << "0";
       cout << min << ":";
       if (sec < 10)
              cout << "0";
       cout << sec;
       cout<<endl;
}

void clockType::incrementHours()
{
       hr++;
           if (hr > 23)
       hr = 0;
}

void clockType::incrementMinutes()
{
       min++;
       if (min > 59) {
       min = 0;
       incrementHours(); //increment hours
       }
}

void clockType::incrementSeconds()
{
       sec++;
       if (sec > 59)
       {
         sec = 0;
         incrementMinutes(); //increment minutes
     }
}

bool clockType::equalTime(const clockType& otherClock) const
{
       return (hr == otherClock.hr
                && min == otherClock.min
                       && sec == otherClock.sec);
}

clockType::clockType() //default constructor
{
       hr = 0;
       min = 0;
```

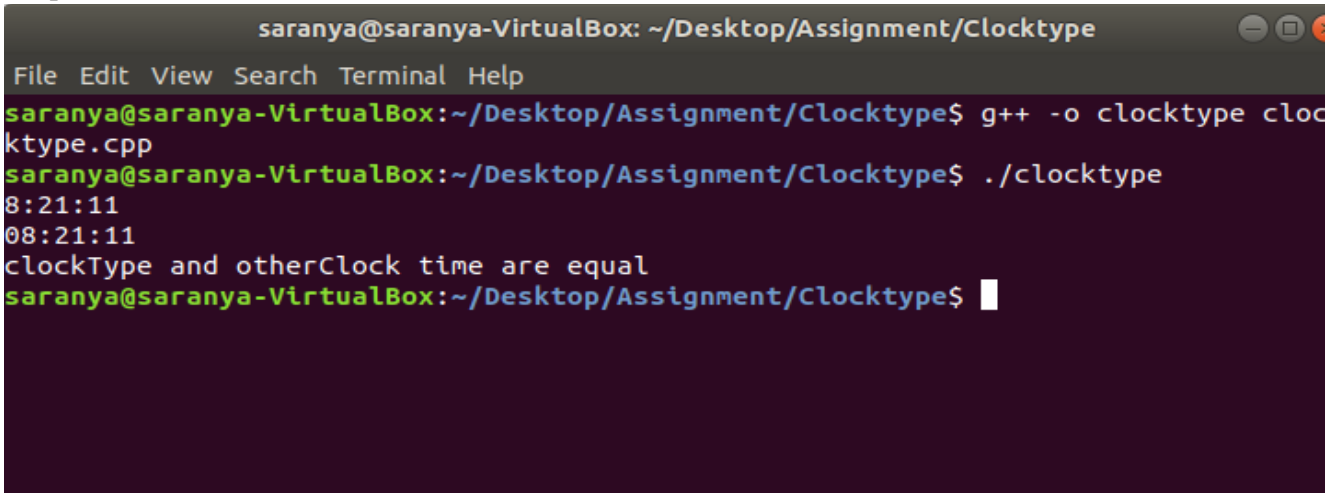
```
        sec = 0;
}


clockType::clockType(int hours, int minutes, int seconds)
{
        if (0 <= hours && hours < 24)
                hr = hours;
        else
        hr = 0;
        if (0 <= minutes && minutes < 60)
                min = minutes;
        else
        min = 0;
        if (0 <= seconds && seconds < 60)
                sec = seconds;
        else
        sec = 0;
}
```

2. Write a program that creates a clocktype object and calls all the methods. Report the results of compiling and running the program.

```
int main()
{
    //Calls constructor
    clockType clocktype1(8,17,10);
    //Creates otherClock as object for class classType
    clockType otherClock;
    //Creates clocktype as object for class classType
    clockType clocktype;
    clocktype.setTime(7,20,10);
    clocktype.incrementHours();
    clocktype.incrementMinutes();
    clocktype.incrementSeconds();
    int hours=8;
    int minutes=9;
    int seconds=10;
    clocktype.getTime(hours,minutes,seconds);
    cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
    clocktype.printTime();
    clocktype = otherClock;
    clocktype.equalTime(otherClock);
    if(clocktype.equalTime(otherClock)==1)
       cout<<"clockType and otherClock time are equal"<<endl;
    else
       cout<<"clockType and otherClock time are not equal"<<endl;
       return 0;
}
```

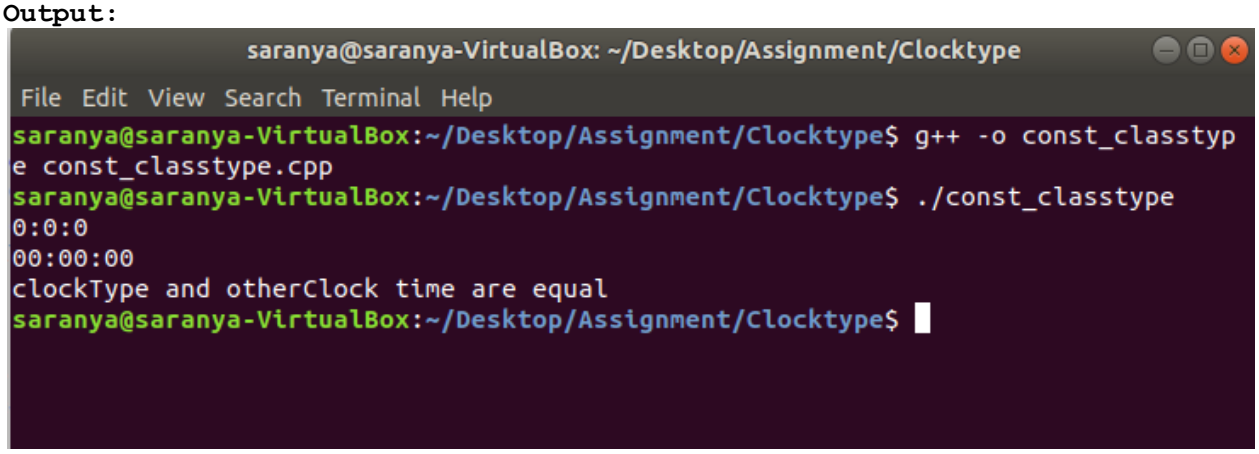
Output:



3. Write a new program that creates a const clocktype object and invoke all the const methods on it. Report the results of compilation and running the program.

```
int main()
{
    //Creates otherClock as object for class classType
    const clockType otherClock;
    // Creates clocktype as object for class classType
    const clockType clocktype;
    int hours=18;
    int minutes=19;
    int seconds=18;
    clocktype.getTime(hours,minutes,seconds);
    cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
    clocktype.printTime();
    clocktype.equalTime(otherClock);
    if(clocktype.equalTime(otherClock)==1)
       cout<<"clockType and otherClock time are equal"<<endl;
    else
       cout<<"clockType and otherClock time are not equal"<<endl;
       return 0;
}
```

Output:



4. Write another program that creates a const clocktype object and invoke all methods on it (const and non const) report the results of compilation.

```
int main()
{
    clockType otherClock;
    //Calls constructor
    clockType clocktype1(8,17,10);
    // Creates clocktype as object for class classType
    const clockType clocktype;
    clocktype.setTime(7,20,10);
    clocktype.incrementHours();
    clocktype.incrementMinutes();
    clocktype.incrementSeconds();
    int hours=8;
    int minutes=9;
    int seconds=10;
    clocktype.getTime(hours,minutes,seconds);
    cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
    clocktype.printTime();
    clocktype.equalTime(otherClock);
    if(clocktype.equalTime(otherClock)==1)
       cout<<"clockType and otherClock time are equal";
    else
       cout<<"clockType and otherClock time are not equal";
       return 0;
}
```

Output: Compilation error on setTime ,incrementMinutes, incrementHours, incrementSeconds because it is not declared as const.