TASK - 01

1. **What is operator overloading?**

Operator overloading is used to overload or redefines most of the operators available in C++. It is used to perform the operation on the user-defined data type.

2. **Is it possible to overload the '+' operator for data type int?**

No, We cannot redefine a built-in operator.

3. **How do we differentiate between prefix and postfix increment operator while overloading them?**

If the overloaded operator has an int parameter, the operator is a postfix overload. If the overloaded operator has no parameter, the operator is a prefix overload

4. **What is the syntax of overloading '*' operator for a class Matrix. (You do not need to write the body of the function)**

```
Matrix operator*(Matrix x);
```

5. **Go through example 5.2 of the manual and write the output of the given code segment.**

```
counter c1(5), c2(10), c3;
c3=c1++;
c2=--c3;;
cout<<'\n'<<c1.get_count();
cout<<'\n'<<c2.get_count();
cout<<'\n'<<c3.get_count();
```

6

4

4

TASK – 02

Write a complete C++ program with the following features.

a. Declare a class Time with two fields hour and minute.

b. Provide appropriate constructors to initialize the data members.

c. Overload the pre and postfix increment operators to increment the minutes by one . Also make sure if minutes are 59 and you increment the Time it should update hours and minutes accordingly.

d. Provide a function display() to display the hours and minutes.

e. In main(), create two objects of Time, initialize them using constructors and call the display functions.

f. Test the following in your main:

a. T3= ++T1;

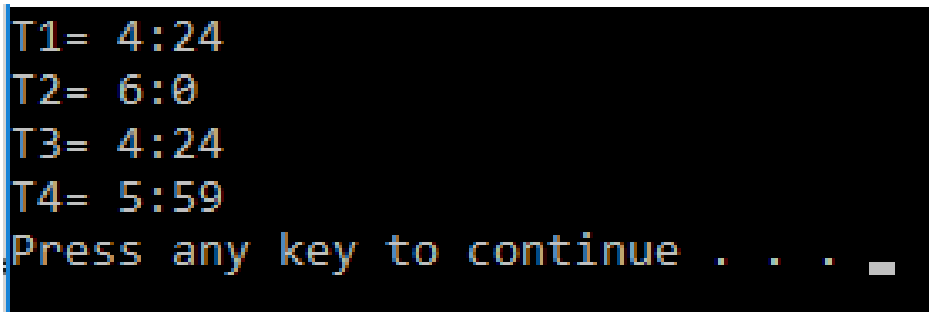
b. T4=T2++;

CODE:

```
// Lab5.cpp : Defines the entry point for the console application.
//

#include"stdafx.h"
#include<iostream>
usingnamespacestd;

classTime{
private:
	inthrs;
	int min;
public:
	Time(){
		hrs=0;
		min=0;
	}
```

```
	Time(inth,int m){
		hrs=h;
		min=m;
	}

	Time operator++(){
		++min;
		if(min>=60){
			hrs+=1;
			min-=60;}
		return Time(hrs,min);
	}

	Time  operator++(int){
		Time t(hrs,min);
		min++;
		if(min>=60){
			hrs+=1;
			min-=60;}
		return t;
	}

	voiddisplay(){

		cout<<hrs<<":"<<min<<endl;
	}
};

int _tmain(intargc, _TCHAR* argv[])
{
	Time t1(4,23),t2(5,59),t3,t4;
	t3=++t1;
	t4=t2++;
	cout<<"T1= ";
	t1.display();
	cout<<"T2= ";
	t2.display();
	cout<<"T3= ";
	t3.display();
	cout<<"T4= ";
	t4.display();
	system("pause");
	return 0;
}
```

OUTPUT:



TASK – 03

Implement above taks in sepearte header file and cpp file

Time.cpp:

```
#include"stdafx.h"
#include<iostream>
#include"Time.h"
usingnamespacestd;



Time::Time(){
            hrs=0;
            min=0;
      }

      Time::Time(inth,int m){
            hrs=h;
            min=m;
      }

      Time Time::operator++(){
            ++min;
            if(min>=60){//MIN>59
                  hrs+=1;
                  min-=60;}
            return Time(hrs,min);
      }

      Time Time::operator++(int){
```

```
            Time t(hrs,min);
            min++;
            if(min>=60){
                  hrs+=1;
                  min-=60;}
            return t;
      }

      voidTime::display(){

            cout<<hrs<<":"<<min<<endl;
      }
```

Time.h:

```
classTime{
private:
      inthrs;
      int min;
public:
      Time();
      Time(inth,int m);
      Time operator++();
      Time operator++(int);
      voiddisplay();
};
```

Main.cpp:

```
// Lab5.cpp : Defines the entry point for the console application.
//

#include"stdafx.h"
#include<iostream>
#include"Time.h"
usingnamespacestd;


int _tmain(intargc, _TCHAR* argv[])
{
      Time t1(4,23),t2(5,59),t3,t4;
      t3=++t1;
      t4=t2++;
```

```
	cout<<"T1= ";
	t1.display();
	cout<<"T2= ";
	t2.display();
	cout<<"T3= ";
	t3.display();
	cout<<"T4= ";
	t4.display();
	system("pause");
	return 0;
}
```

OUTPUT:

```
T1= 4:24
T2= 6:0
T3= 4:24
T4= 5:59
Press any key to continue . .
```