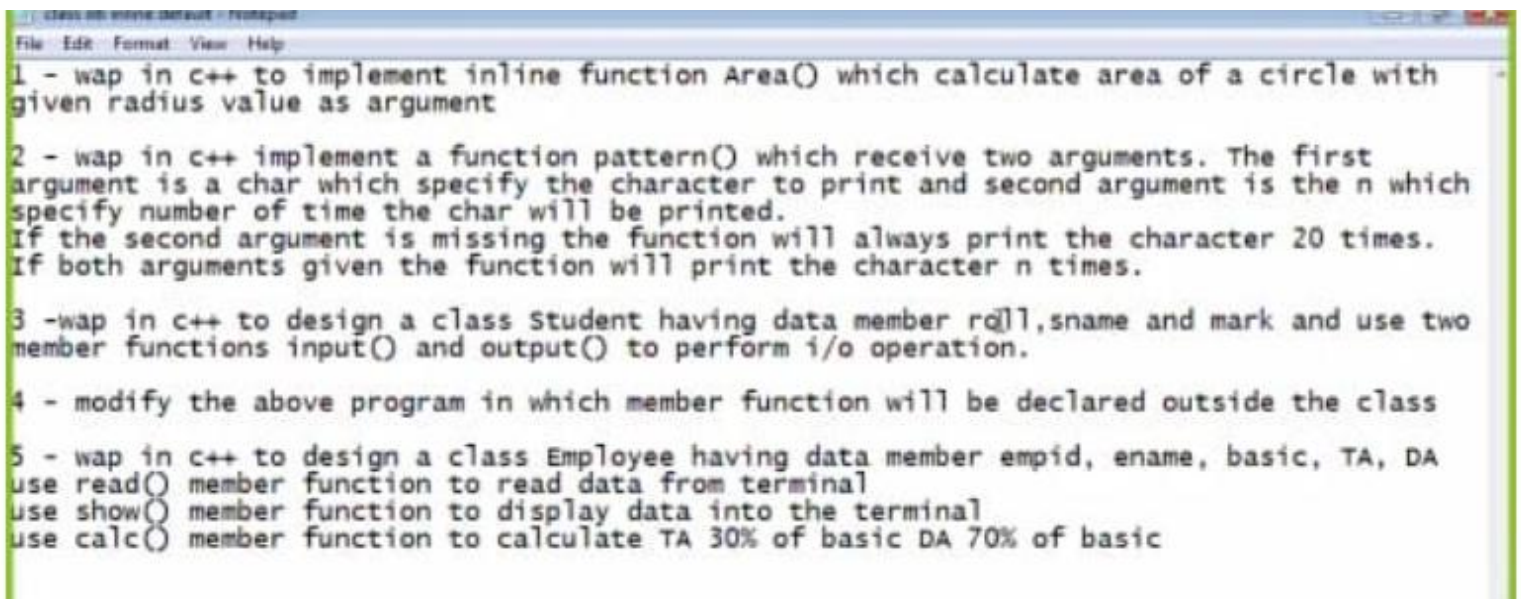


CHAUDHARY HAMDAN

1905387

OOP LAB-3

Date : 14-08-2020



```
class abc {
  ...
}
File Edit Format View Help
1 - wap in c++ to implement inline function Area() which calculate area of a circle with
given radius value as argument
2 - wap in c++ implement a function pattern() which receive two arguments. The first
argument is a char which specify the character to print and second argument is the n which
specify number of time the char will be printed.
If the second argument is missing the function will always print the character 20 times.
If both arguments given the function will print the character n times.
3 -wap in c++ to design a class Student having data member roll,sname and mark and use two
member functions input() and output() to perform i/o operation.
4 - modify the above program in which member function will be declared outside the class
5 - wap in c++ to design a class Employee having data member empid, ename, basic, TA, DA
use read() member function to read data from terminal
use show() member function to display data into the terminal
use calc() member function to calculate TA 30% of basic DA 70% of basic
```

1.

```
#include<iostream>

using namespace std;

inline float area(int r)
{
    return (3.14*r*r);
}

int main()
{
    int rad;

    cout<<"Enter radius of the circle\n";

    cin>>rad;

    cout<<"Area of the circle is "<<area(rad)<<endl;
}
```

```
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cmd /c .\"140820.exe"
Enter radius of the circle
7
Area of the circle is 153.86
PS C:\Users\KIIT\Desktop\projects\OOP LAB> |
```

2.

```
#include<iostream>

using namespace std;

void pattern(char c,int n=20)
{
    for(int i=0;i<n;i++)
    {
        cout<<c<<"\t";
    }
    cout<<endl;
}

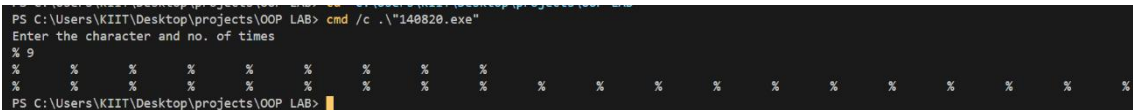
int main()
{
    int n;char c;

    cout<<"Enter the character and no. of times\n";

    cin>>c>>n;

    pattern(c,n);

    pattern(c);
}
```



```
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cmd /c .\"140820.exe"
Enter the character and no. of times
9
* * * * *
PS C:\Users\KIIT\Desktop\projects\OOP LAB>
```

3.

```
#include<iostream>

#include<string>

using namespace std;

class Student
{
    int roll;

    string name;

    int mark;

public:

    void input()
    {
        cout<<"Enter The Roll No.\n";

        cin>>roll;

        cout<<"Enter Name \n";

        cin>>name;

        cout<<"Enter marks in one subject\n";

        cin>>mark;

    }

    void display()
    {

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

        cout<<"Roll No.: "<<roll;

        cout<<"\tMarks: "<<mark<<endl;

    }

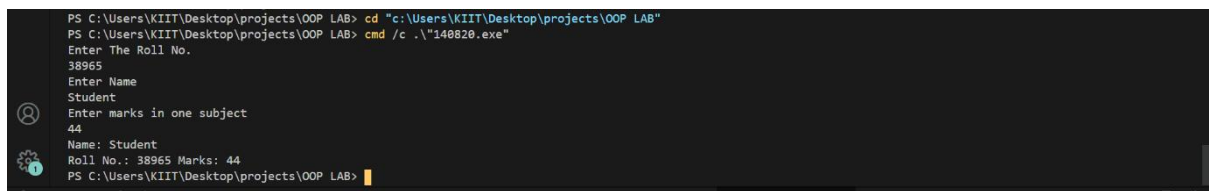
};
```

```
int main()
{
    Student obj;

    obj.input();

    obj.display();

    return 0;
}
```



```
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cd "c:\Users\KIIT\Desktop\projects\OOP LAB"
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cmd /c .\"140820.exe"
Enter The Roll No.
38965
Enter Name
Student
Enter marks in one subject
44
Name: Student
Roll No.: 38965 Marks: 44
PS C:\Users\KIIT\Desktop\projects\OOP LAB>
```

4.

```
#include<iostream>

#include<string>

using namespace std;

class Student
{
    int roll;

    string name;

    int mark;

public:
    void input();
    void display();
};

void Student::input()
{
    cout<<"Enter The Roll No.\n";

    cin>>roll;

    cout<<"Enter Name \n";

    cin>>name;

    cout<<"Enter marks in one subject\n";

    cin>>mark;
}

void Student::display()
{
    cout<<"Name: "<<name<<endl;

    cout<<"Roll No.: "<<roll;
```

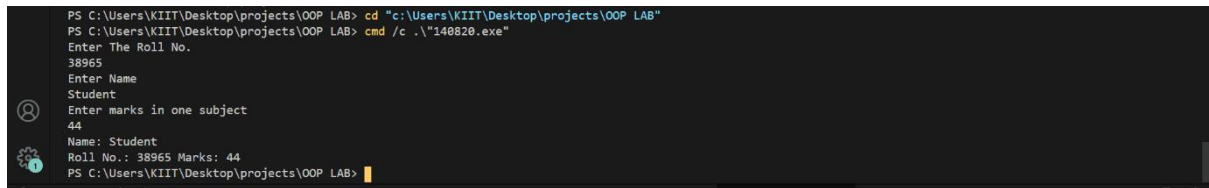
```
        cout<<"\tMarks: "<<mark<<endl;
    }

    int main()
    {
        Student obj;

        obj.input();

        obj.display();

        return 0;
    }
```



```
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cd "c:\Users\KIIT\Desktop\projects\OOP LAB"
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cmd /c .\"148820.exe"
Enter The Roll No.
38965
Enter Name
Student
Enter marks in one subject
44
Name: Student
Roll No.: 38965 Marks: 44
PS C:\Users\KIIT\Desktop\projects\OOP LAB>
```

5.

```
#include<iostream>  
#include<string>  
using namespace std;  
class Student  
{  
int empid;  
string name;  
int basic;  
float ta,da;  
public:  
void read();  
void show();  
void calc();  
};  
void Student::read()  
{  
    cout<<"Enter The Employee Id\n";  
    cin>>empid;  
    cout<<"Enter Name \n";  
    cin>>name;  
    cout<<"Enter Basic Salary\n";  
    cin>>basic;  
}  
void Student::show()  
{
```



```

    cout<<"Name: "<<name;

    cout<<"\tEmp Id: "<<empid<<endl;

    cout<<"Basic Salary: "<<basic<<endl;

    cout<<"TA: "<<ta;

    cout<<"\tDA: "<<da<<endl;

}

void Student::calc()

{

    ta = 0.3*basic;

    da=0.7*basic;

}

int main()

{

    Student obj;

    obj.read();

    obj.calc();

    obj.show();

    return 0;

}

```

```

PS C:\Users\KIIT\Desktop\projects\OOP LAB> cd "c:\Users\KIIT\Desktop\projects\OOP LAB"
PS C:\Users\KIIT\Desktop\projects\OOP LAB> cmd /c .\"140820.exe"
Enter The Employee Id
5686902
Enter Name
employee
Enter Basic Salary
10000
Name: employee Emp Id: 5686902
Basic Salary: 10000
TA: 3000 DA: 7000
PS C:\Users\KIIT\Desktop\projects\OOP LAB>

```

6. WAP to swap private data member of two classes. [The classes have no relation with each other].

```
#include <iostream>
using namespace std;

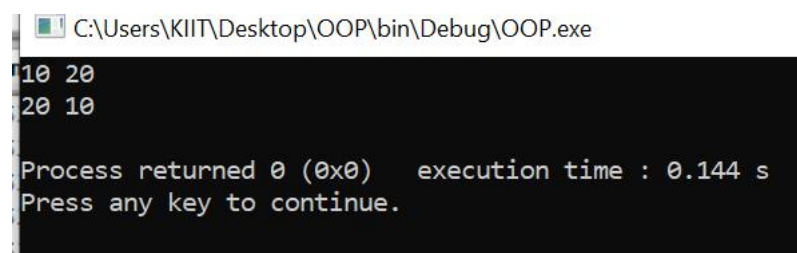
class B;
class A
{
    public:
    int x = 10;

    friend class B;
};

class B
{
    public :
    int y=20;
    friend class A;
    void swapp(A &ob)
    {
        ob.x = ob.x+y;
        y=ob.x-y;
        ob.x=ob.x-y;
    }
};

int main()
{
    A ob1;
    B ob;
    cout << ob1.x << " " << ob.y << endl;
    ob.swapp(ob1);
    cout << ob1.x << " " << ob.y << endl;

    return 0;
}
```



```
C:\Users\KIIT\Desktop\OOP\bin\Debug\OOP.exe
10 20
20 10

Process returned 0 (0x0)   execution time : 0.144 s
Press any key to continue.
```

7. Create two classes which stores distance in feet, inches and meter, centimeter format respectively. Write a function which compares distance in object of these classes and displays the larger one.

```
#include<iostream>
using namespace std;

class Met;
class inc{
    float feet, inches;
public:
    float total;
    void getdata(){
        cout<<"Enter the distance in feet and inches: "<<endl;
        cin>>feet;
        cin>>inches;
        total = (feet * 12) + inches;
        total = total * 2.54;
    }
    friend void calc(Met, inc);
};

class Met{
    float met, cent;
public:
    float total;
    void getdata(){
        cout<<"Enter the distance in metres and centimetres: "<<endl;
        cin>>met;
        cin>>cent;
        total = met * 100 + cent;
    }
    friend void calc(Met, inc);
};

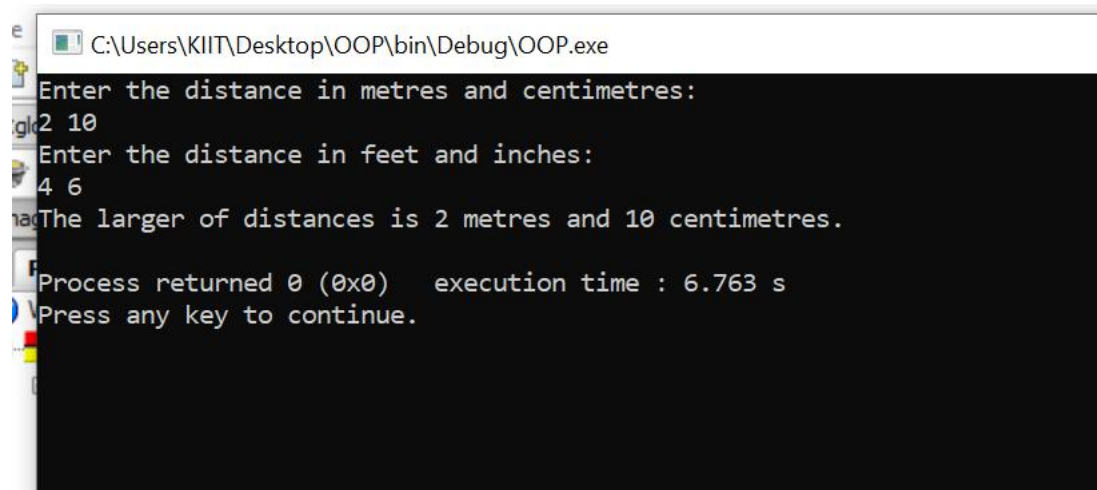
void calc(Met m, inc i){
    if(m.total > i.total){
        cout<<"The larger of distances is "<<m.met<<" metres and "<<m.cent<<"
centimetres."<<endl;
```

```

    }
    else{
        cout<<"The larger of distances is "<<i.feet<<" feet and "<<i.inches<<"
        inches."<<endl;
    }
}

int main(){
    Met m;
    inc i;
    m.getdata();
    i.getdata();
    calc(m, i);
    return 0;
}

```



```

C:\Users\KIIT\Desktop\OOP\bin\Debug\OOP.exe
Enter the distance in metres and centimetres:
2 10
Enter the distance in feet and inches:
4 6
The larger of distances is 2 metres and 10 centimetres.
Process returned 0 (0x0)   execution time : 6.763 s
Press any key to continue.

```

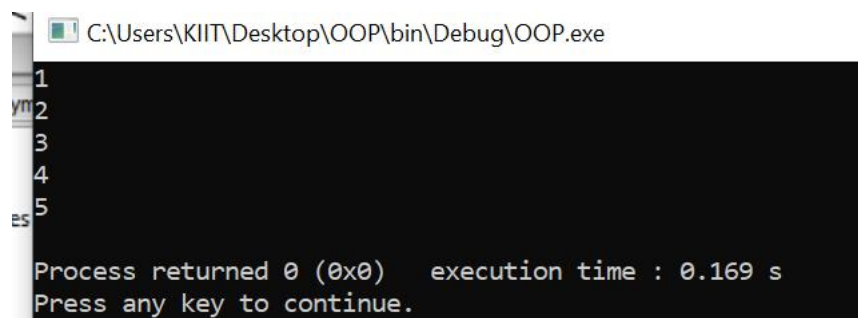
8. Create a class with an integer data member. Include functions for input and output in class. Count the number of times each function is called and display it.

```
#include <iostream>
using namespace std;
class A
{
    static int c;
    static void countt()
    {
        c++;
    }
public :
    static void display()
    {
        countt();
        cout << c << endl;
    }
};
```

```
int A :: c;
```

```
int main()
{
    A :: display();
    A :: display();
    A :: display();
    A :: display();
    A :: display();

    return 0;
}
```



```
C:\Users\KIIT\Desktop\OOP\bin\Debug\OOP.exe
1
2
3
4
5
Process returned 0 (0x0)   execution time : 0.169 s
Press any key to continue.
```

9. Create a class which stores name, roll number and total marks for a student. Input data for n students. Find the average marks scored by n students, store it as a data member of the class and display it using a function which may be called without object.

```
#include <iostream>
using namespace std;
```

```
class students
{
    string name;
    int roll;
    int tmarks;
    static int avgMarks;

public:
    void getdata()
    {
        cout<<"Enter Name: ";
        cin>>name;
        cout<<"Enter Roll: ";
        cin>>roll;
        cout<<"Enter Total Marks: ";
        cin>>tmarks;
    }
    static void display(students obj[],int n)
    {
        for(int i=0; i<n; i++)
        {
            avgMarks=avgMarks+obj[i].tmarks;
        }
        cout<<"\nAverage marks of the students : "<<(avgMarks/n);
    }
};

int students::avgMarks=0;
int main()
```

```

{
    int n;
    cout<<"Enter number of students: ";
    cin>>n;
    students ob[n];
    for(int i=0; i<n; i++)
    {
        ob[i].getdata();
    }
    cout << "\n\n";
    students::display(ob,n);

    return 0;
}

```

```

C:\Users\KIIIT\Desktop\OOP\bin\Debug\OOP.exe
Enter number of students: 2
Enter Name: hamdan
Enter Roll: 387
Enter Total Marks: 50
Enter Name: hamm
Enter Roll: 783
Enter Total Marks: 05

Average marks of the students : 27
Process returned 0 (0x0)   execution time : 35.161 s
Press any key to continue.

```

10. Create a class which stores name, author and price of a book. Store information for n number of books. Display information of all the books in a given price range using friend function.

```
#include<iostream>
#include<iomanip>
#include<string.h>
using namespace std;

class bookstore{
    string name;
    string author;
    float price;
public:
    void getbooks(){
        cout<<"Enter the name of the book: "<<endl;
        cin>>name;
        cout<<"Enter the author: "<<endl;
        cin>>author;
        cout<<"Enter the price: "<<endl;
        cin>>price;
    }
    friend void check(bookstore b[], int n);
};

void check(bookstore b[], int n){
    int i;
    int low, up;
    cout<<"Enter the lower index of the price range: "<<endl;
    cin>>low;
    cout<<"Enter the upper index of the price range: "<<endl;
    cin>>up;
    cout<<"\nThe books in the range "<<low<<" to "<<up<<" are: "<<endl;
```



```

        for( i = 0; i < n; i++){
            if(b[i].price >= low && b[i].price <= up){
                cout<<"\nName of the book"<<right<<setw(3)<<" :
"<<right<<setw(5)<<b[i].name<<endl;
                cout<<"Author of the book"<<right<<setw(3)<<" :
"<<right<<setw(5)<<b[i].author<<endl;
                cout<<"Price of the book"<<right<<setw(3)<<" :
"<<right<<setw(5)<<b[i].price<<endl;
            }
            else
            {
                cout<<"No books are in this price range."<<endl;
            }
        }
    }
}

```

```

int main(){
    int n, i;
    cout<<"Enter the number of books: "<<endl;
    cin>>n;
    bookstore b[n];
    for(i = 0; i < n; i++){
        b[i].getbooks();
    }
    check(b, n);
    return 0;
}

```

C:\Users\KIIT\Desktop\OOP\bin\Debug\OOP.exe

```
Enter the number of books:
2
Enter the name of the book:
harry
Enter the author:
allen
Enter the price:
97
Enter the name of the book:
potter
Enter the author:
walker
Enter the price:
98
Enter the lower index of the price range:
0
Enter the upper index of the price range:
100

The books in the range 0 to 100 are:

Name of the book : harry
Author of the book : allen
Price of the book :    97

Name of the book : potter
Author of the book : walker
Price of the book :    98

Process returned 0 (0x0)   execution time : 28.847 s
Press any key to continue.
```