



Assignment 3

OOPL-LAB, B.Tech II Semester, 2022, IIIT Pune

GROUP 19:

Team Members:

Mis No	Students Name
112115156	SOHAM SANGHAVI
112115157	SOM SINGH LODHI
112115158	SOURISH MITTAL
112115159	SRIJAN KHANDELWAR
112115160	SRUSHTI SURESH WAGHMARE

GitHub Repo Link:

Click on this link to see code

<https://github.com/SomSingh23/Object-Oriented-Lab3>

Input:

```
84 int main(){
85     cout<<endl;
86     cout<<fixed<<setprecision(2);
87     Polar one(2,pi) ;
88     one.display() ; You, 1 second ago • Uncommitted chan
89     Polar two(5 , (3*pi)/2) ;
90     two.display();
91     // Polar three = two - one; //three.display() ;
92     Polar four = two*one;
93     four.display();
94     Polar five = one - two ;
95     Polar six = two - one ;
96     five.display();
97     six.display() ;
98
99     return 0 ;
100 }
```

Output:

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

PS C:\Users\somsi\oops_lab3____> cd "c:\Users\somsi\oops_lab3____\" ; if ($?) { g++ lab3.cpp -o lab3 } ; if ($?) { .\lab3 }

Polar representation (in Radian):      2.00( cos(3.14)+ i*sin(3.14) )
Polar representation (in Degree):      2.00( cos(180.00)+ i*sin(180.00) )
Coordinate Representation :            -2.00 + i* 0.00

Polar representation (in Radian):      5.00( cos(4.71)+ i*sin(4.71) )
Polar representation (in Degree):      5.00( cos(270.00)+ i*sin(270.00) )
Coordinate Representation :            0.00 + i* -5.00

Polar representation (in Radian):      10.00( cos(1.57)+ i*sin(1.57) )
Polar representation (in Degree):      10.00( cos(90.00)+ i*sin(90.00) )
Coordinate Representation :            0.00 + i* 10.00

Polar representation (in Radian):      5.39( cos(1.95)+ i*sin(1.95) )
Polar representation (in Degree):      5.39( cos(111.80)+ i*sin(111.80) )
Coordinate Representation :            -2.00 + i* 5.00

Polar representation (in Radian):      5.39( cos(5.09)+ i*sin(5.09) )
Polar representation (in Degree):      5.39( cos(291.80)+ i*sin(291.80) )
Coordinate Representation :            2.00 + i* -5.00

PS C:\Users\somsi\oops_lab3____> 
```

C++ code:

```
#include <bits/stdc++.h>

using namespace std ;

const double pi = acos(-1); // use pi instead of 3.14

class Polar{ // class declare here

private:

double x ; // real

double y ; // imaginary

double r ;

double theta ;

public:

// only one constructor

Polar(double radius , double radian){

r = radius ;

if(radian == pi/2)

{ x = 0;

y = radius;

theta = radian; }

else if(radian == 0){ x= radius ; y=0;theta = radian;}

else if(radian == (3*pi)/2){

y= -1*radius ;

x = 0 ; theta = (-1 * pi)/2;

}

else if(radian ==pi){ y=0 ; x=-1*radius ;theta = radian;}

else if(radian==2*pi){ y=0; x = radius ;theta = radian; }

else {

x = radius*(cos(radian));theta = radian;

y = radius*(sin(radian)); //cout<<"else called out"<<endl;

}
```

```

}

// display funtion to display polar representation
void display(){

    cout<<"Polar representation (in Radian):"<<"\t"; cout<<r<<"( cos"<<"("<<theta<<")"<<"+"
i*sin("<<theta<<") )" <<endl;

    cout<<"Polar representation (in Degree):"<<"\t"; cout<<r<<"( cos"<<"("<<(theta*180)/pi<<")"<<"+"
i*sin("<<(theta*180)/pi<<") )" <<endl;

    cout<<"Coordinate Representation : "<<"\t"<<"\t"; cout<<x<<" + i * " <<y<<endl;

    cout<<endl;
}

// operator *

Polar operator*(Polar &obj1); // prototype declared here its not a friend function

friend Polar operator-(Polar &obj1 , Polar &obj2) ; // prototype of friend function to implement - overator for two
polar number

~Polar(){/* destructor is declared here */ }

};

// outside class representation of operator overloading
Polar Polar::operator*(Polar &obj1){

double xx = x*obj1.x ;

double yy = y*obj1.y ;

double fx = -yy+xx ;

double ixx = x*obj1.y;

double iyy = y*obj1.x ;

double fy = ixx+iyy ;

double radius2 = (fx*fx + fy*fy) ;

double radius = pow(radius2 , 0.5) ;

double rad = atan(abs(fy/fx)) ;

// cout<<radius<<" " <<rad<<endl;

if(fy>=0 and fx>=0){rad = rad; }

else if(fy<=0 and fx<=0){rad = rad - pi; }

else if(fy>=0 and fx<=0){rad = pi - rad; }

else if(fy<=0 and fx>=0){rad = -1 * rad; }

```

```

Polar crazyxyz(radius , rad) ;

return crazyxyz ;

}

// friend function body

Polar operator -(Polar &obj1 , Polar &obj2){

    // cout<<obj1.x<<" "<<obj1.y<<endl;

    // cout<<obj2.x<<" "<<obj2.y<<endl;

    double xx = obj1.x + (-1*obj2.x) ;

    double yy = obj1.y + ( -1*obj2.y );

    // cout<<xx<<" "<<yy<<endl;


    double radius2 = (xx*xx + yy*yy) ;

    double radius = pow(radius2 , 0.5) ;

    double rad = atan(abs(yy/xx)) ;

    // cout<<radius<<" "<<rad<<endl;

    if(yy>=0 and xx>=0){rad = rad; }

    else if(yy<=0 and xx<=0){rad = rad - pi; }

    else if(yy>=0 and xx<=0){rad = pi - rad; }

    else if(yy<=0 and xx>=0){rad = -1 * rad; }

    Polar xyz(radius , rad) ;

    return xyz;

}

int main(){

    cout<<endl;

    cout<<fixed<<setprecision(2);

    Polar one(2,pi) ;

    one.display() ;

    Polar two(5 , (3*pi)/2) ;

    two.display();

    // Polar three = two - one; //three.display() ;

    Polar four = two*one;

```

four.display();

Polar five = one - two ;

Polar six = two - one ;

five.display();

six.display() ;

return 0 ;

}