classmate
Date
Page

EXPERIMENT 7

Write a program to demonstrate the compile time polymorphism (function overloading and operator overloading)

① WAP using function overloading to calculate the area of a labaratory (which is rectangular) & area of classroom (which is square)

② WAP using function overloading to calculate the sum of 5 float values and sum of 10 integers values

③ WAP to implement unary ++ operator when used with the object so that the numeric data member of the class is negated.

④ WAP to implement the unary ++ operator (for pre increment & post increment) when used with the object so that the numeric data member of the class is incremented.

①
```
#include <iostream>
using namespace std;
float area (float length, float breadth)
{
    return length * breadth;
}
float area (float side)
{
```

```
return length side * Side;
}
int main()
{
float length, breadth, side;
cout<< "Enter length & breadth of the laboratory
 :";
cin>> length >> breadth;
cout << "Area of laboratory: "<< area (length,
   breadth ) << endl;
~~cout << "Area of laboratory :"~~
cout << "Enter side of classroom: ";
 cin >> side;
 cout << " area of classroom : "<< area (size)
                              << endl;
}
```

Output -

```
Enter length and breadth of the laborator
                              : 5  2
      Area of laboratory : 10
      Enter side of classroom : 5
        Area of classroom : 25
```

②

```
#include<iostream>
using namespace std;
class Sum
{
public:
float calculate (float a, float b, float c, float d,
              float e)
   {
```

classmate
Date
Page

```
return a+b+c+d+e ;
}
int calculate (int a1,int a2,int a3,int a4,int a5
            ,int a6 ,int a7,int a8,int a9,inta
                10)
 {
 return a1+a2+a3+a4+a5+a6+a7+a8+a9+
            a10 ;
  }
 };
int main()
{
 Sum S;
cout <<" Sum of 5 float values: "<<s.calculate
     (1.1,2.2,3.3, 4.4, 5.5) << endl;
cout <<" Sum of 10 integers: "<< s.calculate
     (1,2, 3, 4, 5,6,7,8,9,10) <<endl;
 return 0 ;
}
```

③

```
#include<iostream>
using namespace std;
class num
{
 int i;
 public:
 void accept()
{
cout << "Enter the value:" ;
  cin>>i ;
 }
void display()
{
cout << "Value is:" << i;
 }
void operator ++ ()
{
 i = ++i;
}
};
int main()
{
 num n;
 n.accept ();
 ++n;
 n.display();
 return 0;
}
```

```
Output - Enter the value : 8
         Value is: 9
```

④
```
#include<iostream>
using namespace std;
class Num
 {
 int i;
 public:
 void accept ()
 {
 cout << "Enter the value :";
 cin >> i;
}
void display()
 {
 cout << " value is : " << i << endl;
 }
void operator ++ ()
{
  ++ i ;
}
void operator --()
{
  -- i;
}
void operator ++ (int)
{
 i++ ;
}
void operator -- (int)
 {
   i--  ;
}
};
int main()
 {
 Num n, n1;
  n.accept();
 ++n ;
cout <<"After prefix increment";
 n.display ();
 n1. accept();
  n1 -- ;
```

```
cout <<"After postfix decrement ";
  n1.display();
}
```

Output -
```
Enter the value :10
After prefix increment value is : 11
Enter the value :10
After postfix increment value is : 9
```

x ———— x ———— x

16/10