



**Course code: CSC-284**  
**Lab Report: 07**

Submitted To

Tanzina Tasnim Bithi

Department of Computer, Science and Engineering

Submitted By

Name: Md.Pranto Ali

ID: 23303101

Program: BCSE

Section: A

Semester: Fall 2024

Submission date:

---

Teacher signature

## 1. Sum of number using function overloading:

main.cpp



Share

Run

Output

Clear

```
1 #include <iostream>
2 using namespace std;
3 class addition_numbers {
4     public:
5     void sum(int a, int b) {
6         cout << a + b << endl;
7     };
8     void sum(double a, double b, double c) {
9         cout << a + b + c << endl;
10    };
11    void sum(float a, float b, float c) {
12        cout << a + b + c << endl;
13    };
14 };
15 int main() {
16     cout << "Name: Md.Pranto Ali" << "Id: 23303101" << endl;
17     addition_numbers obj;
18     cout << "Call First Function:" << endl;
19     obj.sum(5, 6);
20     cout << "Call Secound Function:" << endl;
21     obj.sum(5.5, 6.6, 7.7);
22     cout << "Call Third Function:" << endl;
23     obj.sum(8.5, 2.6, 88.7);
24     return 0;
25 }
```

Name: Md.Pranto AliId: 23303101

Call First Function:

11

Call Secound Function:




19.8

Call Third Function:

99.8

=== Code Execution Successful ===




## 2. Finding Area of Circle , Rectangle and square Using Polymorphism:

main.cpp	   Share	Run	Output	Clear
<pre>1 #include&lt;iostream&gt; 2 using namespace std; 3 class Area { 4     public: 5     void find_area(double r) { 6         cout &lt;&lt; "Area of Circle: " &lt;&lt; 3.14 * r * r &lt;&lt; endl; 7     }; 8     void find_area(double l, double b) { 9         cout &lt;&lt; "Area of Rectangle: " &lt;&lt; l * b &lt;&lt; endl; 10    }; 11    void find_area(double s, double s1, double s2) { 12        cout &lt;&lt; "Area of Square: " &lt;&lt; s * s1 * s2 &lt;&lt; endl; 13    }; 14 }; 15 int main() { 16     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; "Id: 23303101" &lt;&lt; endl; 17     Area obj; 18     cout &lt;&lt; "Call First Function:" &lt;&lt; endl; 19     obj.find_area(5); 20     cout &lt;&lt; "Call Secound Function:" &lt;&lt; endl; 21     obj.find_area(5, 6); 22     cout &lt;&lt; "Call Third Function:" &lt;&lt; endl; 23     obj.find_area(5, 5, 5); 24     return 0; 25 };</pre>			<pre>Name: Md.Pranto AliId: 23303101 Call First Function: Area of Circle: 78.5 Call Secound Function: Area of Rectangle: 30 Call Third Function: Area of Square: 125  === Code Execution Successful ===</pre>	




### 3. Operator Overloading Unary Minus:

main.cpp	Run	Output
<pre>1 #include &lt;iostream&gt; 2 using namespace std; 3 class Complex{ 4     private: 5         int a, b; 6     public: 7         Complex(int x, int y){ 8             a = x; 9             b = y; 10        } 11        void display(){ 12            cout &lt;&lt; "Complex number: " &lt;&lt; a &lt;&lt; " + " &lt;&lt; b &lt;&lt; "i" 13                &lt;&lt; endl; 14        } 15        Complex operator -(){ 16            return Complex(-a, -b); 17        } 18    int main() { 19        cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; "Id: 23303101" &lt;&lt; endl; 20        Complex c1(2, 3); 21        Complex c2 = -c1; 22        c2.display(); 23        return 0; 24    }</pre>	<div>Run</div>	<p>Name: Md.Pranto AliId: 23303101</p> <p>Complex number: -2 + -3i</p> <p>=== Code Execution Successful ===</p>






## 4. Operator Overloading Increment:

main.cpp	   Share	Run	Output	Clear
<pre>1 #include &lt;iostream&gt; 2 using namespace std; 3 class Complex{ 4     private: 5         int a, b; 6     public: 7         Complex(int x, int y){ 8             a = x; 9             b = y; 10        } 11        void display(){ 12            cout &lt;&lt; "Complex number: " &lt;&lt; a &lt;&lt; " + " &lt;&lt; b &lt;&lt; "i" 13                &lt;&lt; endl; 14        } 15        Complex operator ++(){ 16            a++; 17            b++; 18            return Complex(a, b); 19        }; 20 int main() { 21     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; "Id: 23303101" &lt;&lt; endl; 22     Complex c1(2, 3); 23     Complex c2 = ++c1; // c2 = c1.operator++; 24     c2.display(); 25     return 0; 26 };</pre>			<p>Name: Md.Pranto AliId: 23303101</p> <p>Complex number: 3 + 4i</p> <p>=== Code Execution Successful ===</p>	

## 5. Operator Overloading Decrement:

main.cpp	   Share	Run	Output	Clear
<pre>1 #include &lt;iostream&gt; 2 using namespace std; 3 class Complex{ 4     private: 5         int a, b; 6     public: 7         Complex(int x, int y){ 8             a = x; 9             b = y; 10        } 11        void display(){ 12            cout &lt;&lt; "Complex number: " &lt;&lt; a &lt;&lt; " + " &lt;&lt; b &lt;&lt; "i" 13                &lt;&lt; endl; 14        } 15        Complex operator --(){ 16            a--; 17            b--; 18            return Complex(a, b); 19        }; 20 int main() { 21     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; "Id: 23303101" &lt;&lt; endl; 22     Complex c1(2, 3); 23     Complex c2 = --c1; // c2 = c1.operator--(); 24     c2.display(); 25     return 0; 26 }; 27</pre>		<pre>Name: Md.Pranto AliId: 23303101 Complex number: 1 + 2i  === Code Execution Successful ===</pre>		

## 6. Overloading Binary Operators:






main.cpp	   Share 	Output 
<pre>1 #include &lt;iostream&gt; 2 using namespace std; 3 class Complex { 4     private: 5         int a, b; 6     public: 7     Complex(int x, int y) { 8         a = x; 9         b = y; 10    } 11    void display() { 12        cout &lt;&lt; "Complex number: " &lt;&lt; a &lt;&lt; " + " &lt;&lt; b &lt;&lt; "i" &lt;&lt;             endl; 13    } 14    Complex operator +(Complex c) { 15        return Complex(a + c.a, b + c.b); 16    } 17 }; 18 int main() { 19     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; "Id: 23303101" &lt;&lt; endl; 20     Complex c1(2, 3); 21     Complex c2(3, 4); 22     Complex c3 = c1 + c2; // c3 = c1.operator+(c2); 23     c3.display(); 24     return 0; 25 }</pre>	<pre>Name: Md.Pranto AliId: 23303101 Complex number: 5 + 7i  === Code Execution Successful ===</pre>	

## 7. Overloading Operators addition using Friend:

main.cpp	Run	Output
<pre>1 2 #include &lt;iostream&gt; 3 using namespace std; 4 class Complex{ 5     private: 6         int a, b; 7     public: 8     Complex(int x, int y){ 9         a = x; 10        b = y; 11    } 12    void display(){ 13        cout &lt;&lt; "Complex number: " &lt;&lt; a &lt;&lt; " + " &lt;&lt; b &lt;&lt; "i" 14            &lt;&lt; endl; 15    } 16    friend Complex operator +(Complex c1, Complex c2); 17 } 18 Complex operator +(Complex c1, Complex c2){ 19     return Complex(c1.a + c2.a, c1.b + c2.b); 20 } 21 int main() { 22     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; endl &lt;&lt; "Id: 23303101" &lt;&lt; 23         endl; 24     Complex c1(3, 2); 25     Complex c2(4, 5); 26     Complex c3 = c1 + c2; // c3 = operator+(c1, c2); 27     c3.display(); 28     return 0; 29 }</pre>	<div>Run</div>	<div>Name: Md.Pranto Ali Id: 23303101 Complex number: 7 + 7i</div> <div>=== Code Execution Successful ===</div>



## 8. Overloading Operator Using Friend (Add,sub):

main.cpp	   Share 	Output 
<pre>1 #include&lt;iostream&gt; 2 using namespace std; 3 class A{ 4     private: 5     int a; 6     public: 7     A(){ 8         a=0; 9     } 10    void display(){ 11        cout&lt;&lt;"Value of a: "&lt;&lt;a&lt;&lt;endl; 12    } 13    friend void operator -(A &amp;obj); 14 }; 15 void operator -(A &amp;obj){ 16     obj.a=-obj.a; 17 } 18 int main(){ 19     cout &lt;&lt; "Name: Md.Pranto Ali" &lt;&lt; endl &lt;&lt; "Id: 23303101" &lt;&lt;         endl; 20     A obj; 21     obj.display(); 22     -obj; 23     obj.display(); 24     return 0; 25 } 26</pre>	<pre>Name: Md.Pranto Ali Id: 23303101 Value of a: 0 Value of a: 0  === Code Execution Successful ===</pre>	