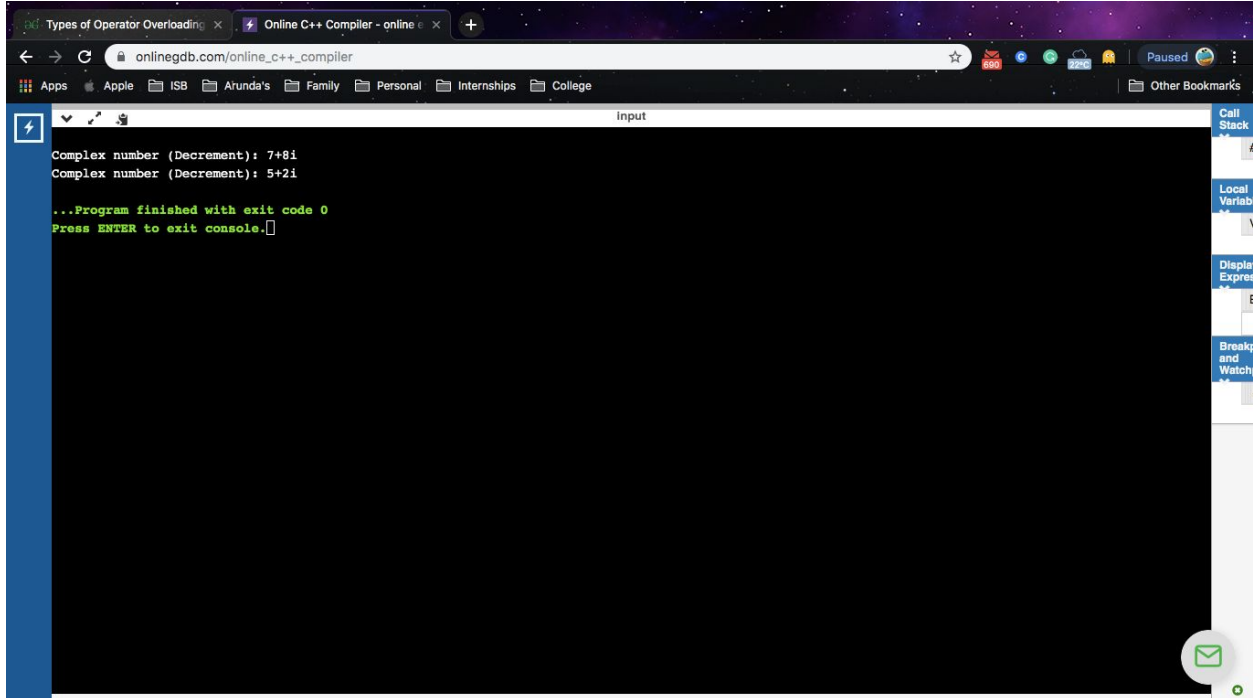


Unary Operator Overloading

```
#include <iostream>
using namespace std;
class Complex
{
    public:
    int real, imag;
    Complex(int r, int i)
    {
        this->real = r;
        this->imag = i;
    }
    void operator-()
    {
        real--;
        imag--;
        cout << "\nComplex number (Decrement): " << real << "+" << imag << "i";
    }
};
int main()
{
    Complex c1(8, 9);
    Complex c2(6,3);
    -c1;
    -c2;

    return 0;
}
```

Output

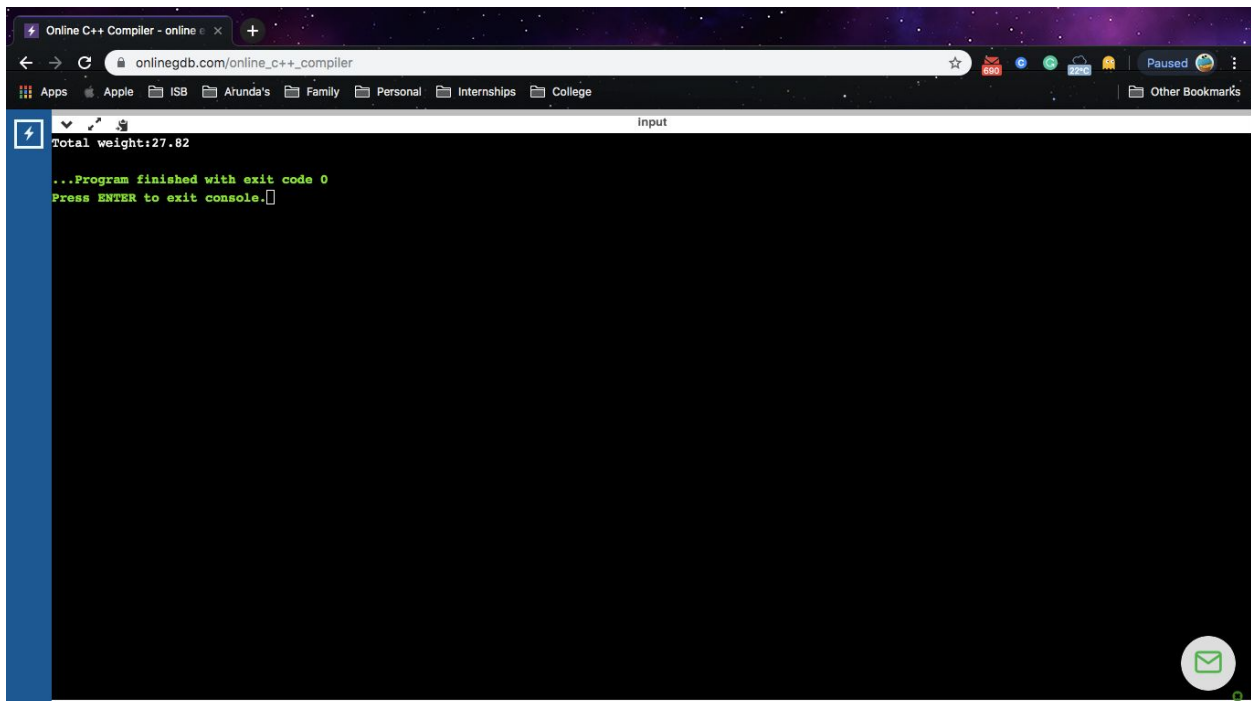
A screenshot of a web browser displaying an online C++ compiler. The browser's address bar shows 'onlinegdb.com/online_c++_compiler'. The compiler's interface includes a top menu bar with 'Types of Operator Overloading' and 'Online C++ Compiler - online'. Below the menu is a toolbar with icons for file operations and a 'Paused' button. The main area is a dark console window with the following text: 'Complex number (Decrement): 7+8i', 'Complex number (Decrement): 5+2i', '...Program finished with exit code 0', and 'Press ENTER to exit console.' On the right side of the console, there are several panels: 'Call Stack', 'Local Variables', 'Display Expressions', and 'Breakpoints and Watchpoints'. A green chat icon is visible in the bottom right corner of the console area.

Binary Operator Overloading

```
#include<iostream>
using namespace std;
class Weight
{
    public:
    int kilograms,grams;
    Weight()
    {
        this->kilograms=0;
        this->grams=0;
    }
    Weight(int kg,int g)
    {
        this->kilograms=kg;
        this->grams=g;
    }
    void printData()
```

```
        {
            cout<<"Kg:"<<kilograms<<"g:"<<grams<<endl;
        }
Weight operator+(Weight &w2)
{
    Weight w3;
    w3.kilograms=this->kilograms+w2.kilograms;
    w3.grams=this->grams+w2.grams;
    return w3;
}
};
int main()
{
    Weight w1(8,15);
    Weight w2(19,67);
    Weight w3;
    w3=w1+w2;
    cout<<"Total weight:"<<w3.kilograms<<". "<<w3.grams;
}
```

Output

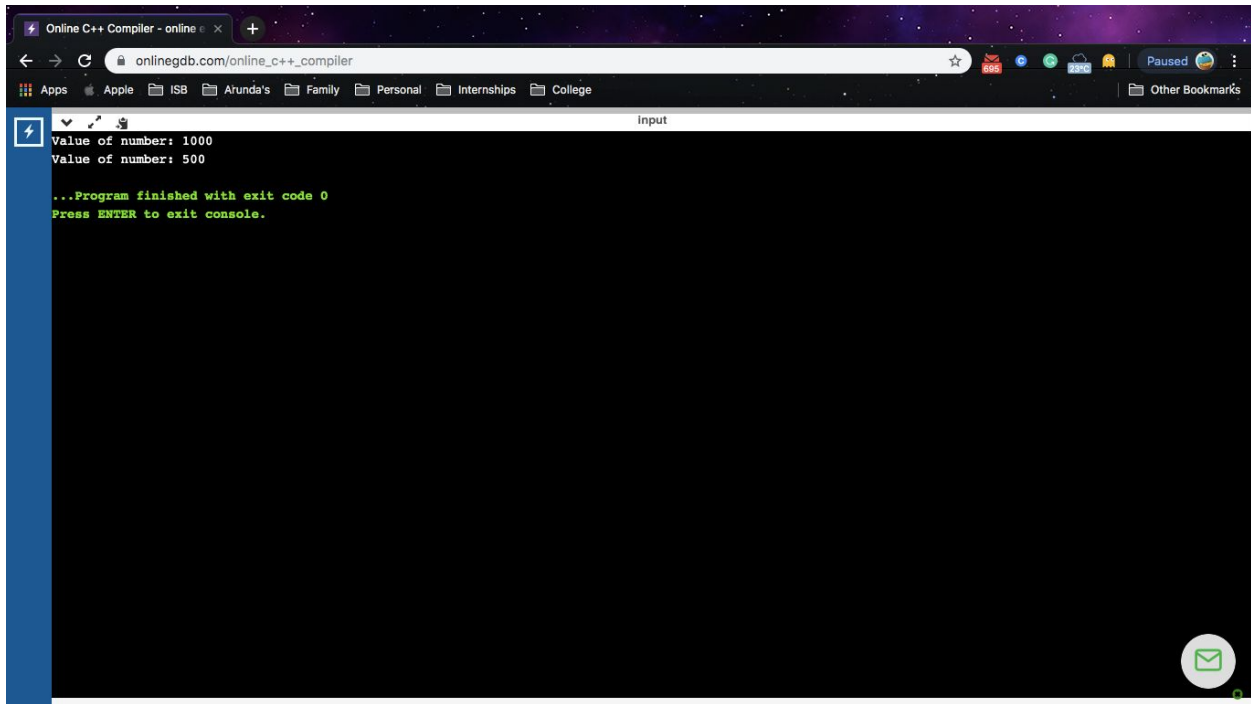
A screenshot of a web browser displaying an online C++ compiler. The browser's address bar shows 'onlinegdb.com/online_c++_compiler'. The page has a dark theme with a purple and black header. Below the header, there's a navigation bar with links to 'Apps', 'Apple', 'ISB', 'Arunda's', 'Family', 'Personal', 'Internships', and 'College'. The main area is a terminal window titled 'Input' with a blue sidebar on the left. The terminal output shows 'Total weight:27.82' on the first line, followed by '...Program finished with exit code 0' and 'Press ENTER to exit console.' on the next two lines. A green circular icon with a white envelope symbol is visible in the bottom right corner of the terminal area.

```
Online C++ Compiler - online × +
onlinegdb.com/online_c++_compiler
Apps Apple ISB Arunda's Family Personal Internships College Other Bookmarks
Input
Total weight:27.82
...Program finished with exit code 0
Press ENTER to exit console.
```

Friend Function Overriding

```
#include <iostream>
using namespace std;
class Number
{
    private:
        int a;
    public:
        void getNum(int x);
        friend void printNum(Number no);
};
void Number::getNum(int x)
{
    a=x;
}
void printNum(Number no)
{
    cout << "Value of number: " << no.a;
}
int main()
{
    Number nObj,nObj1;
    nObj.getNum(1000);
    printNum(nObj);
    cout<<"\n";
    nObj1.getNum(500);
    printNum(nObj1);
    return 0;
}
```

Output

A screenshot of a web browser displaying an online C++ compiler interface. The browser's address bar shows 'onlinegdb.com/online_c++_compiler'. The compiler's output window, titled 'Input', shows the following text: 'Value of number: 1000', 'Value of number: 500', '...Program finished with exit code 0', and 'Press ENTER to exit console.' The browser's taskbar at the bottom shows various icons, including a mail icon in the bottom right corner.

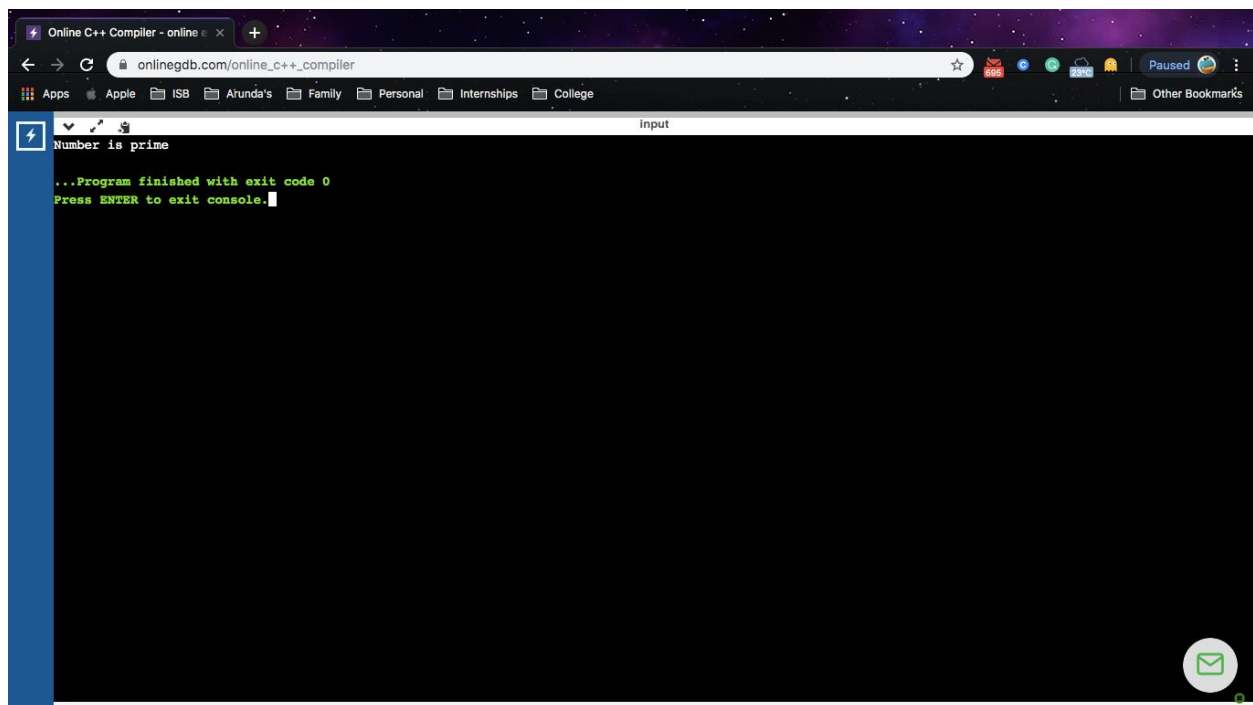
```
Value of number: 1000
Value of number: 500
...Program finished with exit code 0
Press ENTER to exit console.
```

Member Function Overriding

```
#include<iostream>
using namespace std;
class Number
{
    public:
    void show()
    {
        cout<<"Number";
    }
};
class Prime: public Number
{
    public:
    void show()
    {
        cout<<"Number is prime";
    }
}
```

```
};  
int main(void)  
{  
    Prime p = Prime();  
    p.show();  
    return 0;  
}
```

Output

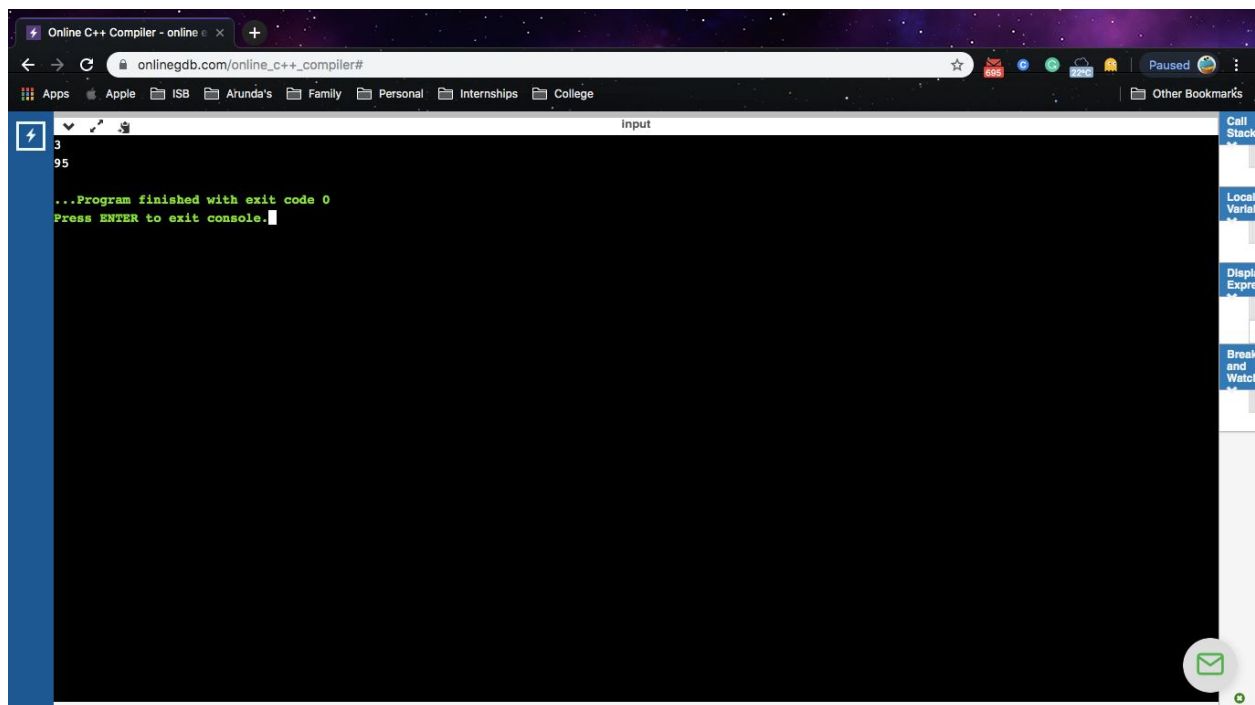


Member Function Overloading

```
#include<iostream>  
using namespace std;  
int sub(int x,int y)  
{  
    cout<<x-y;  
}  
int sub(int x,int y,int z)  
{  
    cout<<x+y+z;
```

```
}  
int main()  
{  
    sub(8,5);  
    cout<<"\n";  
    sub(10,15,70);  
}
```

Output



Friend function overriding

```
#include<iostream>  
using namespace std;  
class complex  
{  
    float real,imag;  
    public:  
    complex()  
    {  
        real=imag=0;  
    }  
}
```

```
}
complex(float r,float i)
{
    real = r;
    imag =i;
}
friend complex operator - (complex c)
{
    c.real=-c.real;
    c.imag=-c.imag;
    return c;
}
void display()
{
    cout<<"\nReal:"<<real;
    cout<<"\nImag:"<<imag;
}
};
int main()
{
    complex c1(1,2),c2;
    c1.display;
    c2=-c1;
    cout<<"\nAfter negation\n";
    c2.display();
}
```