

```

//
//  main.cpp
//  AbsoluteCpp_ch8_6
//

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

class IntPair
{
public:
    IntPair(int firstValue, int secondValue);
    IntPair operator++( ); //Prefix version
    IntPair operator++(int); //Postfix version
    void setFirst(int newValue);
    void setSecond(int newValue);
    int getFirst( ) const;
    int getSecond( ) const;
private:
    int first;
    int second;
};

int main( )
{
    IntPair a(1,2);
    cout << "Postfix a++: Start value of object a: ";
    cout << a.getFirst( ) << " " << a.getSecond( ) << endl;
    IntPair b = a++;
    cout << "Value returned: ";
    cout << b.getFirst( ) << " " << b.getSecond( ) << endl;
    cout << "Changed object: ";
    cout << a.getFirst( ) << " " << a.getSecond( ) << endl;

    a = IntPair(1, 2);
    cout << "Prefix ++a: Start value of object a: ";
    cout << a.getFirst( ) << " " << a.getSecond( ) << endl;
    IntPair c = ++a;
    cout << "Value returned: ";
    cout << c.getFirst( ) << " " << c.getSecond( ) << endl;
    cout << "Changed object: ";
    cout << a.getFirst( ) << " " << a.getSecond( ) << endl;
    return 0;
}

IntPair::IntPair(int firstValue, int secondValue)
    : first(firstValue), second(secondValue)
{ /*Body intentionally empty*/ }

IntPair IntPair::operator++(int ignoreMe) //postfix version
{
    int temp1 = first;
    int temp2 = second;
    first++;

```

```
        second++;
        return IntPair(temp1, temp2);
    }

    IntPair IntPair::operator++( ) //prefix version
    {
        first++;
        second++;
        return IntPair(first, second);
    }

    void IntPair::setFirst(int newValue)
    {
        first = newValue;
    }

    void IntPair::setSecond(int newValue)
    {
        second = newValue;
    }

    int IntPair::getFirst( ) const
    {
        return first;
    }

    int IntPair::getSecond( ) const
    {
        return second;
    }
}
```