

OPERATOR OVERLOADING

```

class Person {
public:
    Person (string s) { name = s; }
    friend ostream& << (ostream& os, const Person&);
private:
    string name;
};

ostream& operator (ostream& os, const Person& p) {
    os << p.name << endl;
    return os;
}
    
```

```

class SmallInt {
private:
    int value;
    bool checkRange (int i);
public:
    static const int MAX = 255;
    static const int MIN = 0;
    explicit SmallInt (int v = 0);
    SmallInt (const SmallInt& orig);
    SmallInt& operator = (const SmallInt& orig);
    SmallInt& operator ++ ();
    const SmallInt operator ++ (int);
};
    
```

```

>> SmallInt s1; => s1(0)
>> SmallInt s2(5); => s2(5)
    
```

```

SmallInt::SmallInt (int v) {
    if (check(v)) {
        value = v;
    }
    else {
        error message;
    }
}
    
```

```

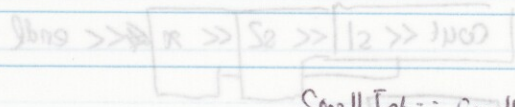
SmallInt::SmallInt (const SmallInt& orig) {
    value = orig.value; //echo called;
}
    
```

```

operator (int) const {
    return value;
}
    
```

```

>> cout << s1 + 5 << endl;
    
```



return type