**Assignment 4 - INT and RINT classes with overloaded operators**

Necessary skills: *class design with operators, operator overloading*

**Description**

**Part A**. Define a class INT that behaves like an int (i.e. can be used in expressions with int values).  To do so, you will need to define an INT::operator int() conversion operator.  Also implement the postfix and prefix ++ and -- operators, and the += and -= assignment operators.

As an example, the following code would work if the class is defined correctly.   
  
   // note - you may need to change the definition of the main function to   
   // be consistent with what your C++ compiler expects.   
   int main() {  
      INT x;  
      INT y = 6;  
      int a = 5, b = 2;  
  
      x = a + b;  
      x += 43;  
      x -= 3;  
      cout << "x = " << x << "  a = " << a << endl;  
  
      a = x + --y;  
      b = x++;  
  
      cout << "x = " << x << "  y = " << y << endl;  
      cout << "a = " << a << "  b = " << b << endl;  
  
      cout << "The absolute value of x is " << abs(x) << endl;.  
   }

**Testing**

Use the attached **INTDemo.cpp** source file (below) to test your INT class.

**Part B**. Define a class RINT (restricted integer) that behaves like an int except that the only operations allowed are + and - (both unary and binary operators), = for assignment of an int to an RINT, and overloaded stream input and output operators for RINT values.  Do not define RINT::operator int().  After you have the test execution output, uncomment the last lines in main, compile again, and also include the compiler error messages.

As an example, the following code would work if the class is defined correctly.

// note - you may need to change the definition of the main function to

// be consistent with what your C++ compiler expects.

int main() {

RINT x, y = 4;

int a = 5, b = 2;

cout << x << endl;

cin >> x;

x = x + 1;

y = x - a;

//x++; // should generate error

//y += 3; // should generate error

}

**Testing**

Use the attached **RINTDemo.cpp** source file (below) to test your RINT class.