Yusuf Fawzy Elnady

Big Decimal Integer

Summary:

Different variations of types **int** and **float** exist in C++ and other languages. They are usually limited by **minimum** and **maximum** values. Sometimes it is desired to have versions of these types with unlimited bounds. Java solves this problem by providing **BigInteger** and **BigDecimal** classes. So, this program develops a new C++ type (class) that can hold **unlimited long decimal integer values** and performs arithmetic operations on them.

Example:

```
BigDecimalInt num1("12345678901234567890123456789O");
BigDecimalInt num2("11345678901134567890113456789O");
BigDecimalInt num3 = num2 + num1;
cout << "num1 = " << num1 << endl;
cout << "num2 = " << num2 << endl;
//23691357802369135780236913578O
cout << "num2 + num1 = " << num3 << endl;
```

Class Methods:

```
BigDecimalInt (string decStr); // Initialize from string and rejects bad input
BigDecimalInt (int decInt); // Initialize from integer
BigDecimalInt operator+ (BigDecimalInt anotherDec); //overloads the + operator
BigDecimalInt operator- (BigDecimalInt anotherDec); //overloads the - operator
int size();
void operator=(BigDecimalInt anotherDec);
void operator=(string a);
void operator=(int n);
friend istream &operator>>(istream &in, BigDecimalInt& b); //overloads the >> operator
friend ostream& operator << (ostream& out, BigDecimalInt b); //overloads the << operator
friend void operator <<(BigDecimalInt &b, string a);
```

Using data encapsulation, you can store the digits of the big decimal integer in whatever container you like, I chose to store as strings.

The project is built using **C++, Code::Blocks**.