//Devin Hardy

//CS372

#include <iostream>

#include <iomanip>

#include <cstdlib>

using namespace std;

class statistic

{

private:

float sum = 0.0;

int length = 0;

float largest = 0.0;

float smallest = 0.0;

public:

statistic(float val);

statistic(float vSum, int vSize, float vBig, float vSmall);

//Methods

void add(float val);

int getlength();

float getsum();

float getaverage();

float getlargest();

float getsmallest();

void emptystat();

//Overload

bool operator!=(statistic S1);

statistic operator+(statistic S1);

void operator=(statistic S1);

friend ostream& operator<<(ostream &out, const statistic& S1);

};

//Method Details

statistic::statistic(float val)

{

sum = val;

length = 1;

largest = val;

smallest = val;

}

statistic::statistic(float vSum, int vSize, float vBig, float vSmall)

{

sum = vSum;

length = vSize;

largest = vBig;

smallest = vSmall;

}

void statistic::add(float val)

{

sum = sum + val;

length++;

if(smallest > val)

smallest = val;

if(largest < val)

largest = val;

}

int statistic::getlength()

{

return length;

}

float statistic::getsum()

{

return sum;

}

float statistic::getaverage()

{

if(length > 0)

return (sum / length);

else

return 0.0;

}

float statistic::getlargest()

{

return largest;

}

float statistic::getsmallest()

{

return smallest;

}

void statistic::emptystat()

{

sum = 0.0;

length = 0;

largest = 0.0;

smallest = 0.0;

}

//Overload

bool statistic::operator!=(statistic S1)

{

return!(sum == S1.getsum() &&

length == S1.getlength() &&

largest == S1.getlargest() &&

smallest == S1.getsmallest());

}

statistic statistic::operator+(statistic S1)

{

float tSum = 0.0;

int tLength = 0;

float tLargest = 0.0;

float tSmallest = 0.0;

tSum = sum + S1.getsum();

tLength = length + S1.getlength();

if(largest > S1.getlargest())

tLargest = largest;

else

tLargest = S1.getlargest();

if(smallest < S1.getsmallest())

tSmallest = smallest;

else

tSmallest = S1.getsmallest();

statistic statSum(tSum, tLength, tLargest, tSmallest);

return statSum;

}

void statistic::operator=(statistic S1)

{

sum = S1.getsum();

length = S1.getlength();

largest = S1.getlargest();

smallest = S1.getsmallest();

}

ostream& operator<<(ostream &out, const statistic& S1)

{

out << "Sum = " << S1.sum << endl;

out << "Length = " << S1.length << endl;

out << "Average = " << (S1.sum / S1.length) << endl;

out << "Largest = " << S1.largest << endl;

out << "Smallest = " << S1.smallest << endl;

return out;

}

//function

bool operator==(statistic S1, statistic S2)

{

return( S1.getsum() == S2.getsum() &&

S1.getlength() == S2.getlength() &&

S1.getlargest() == S2.getlargest() &&

S1.getsmallest() == S2.getsmallest());

}

int main()

{

// Create stats

statistic statistician1(5.5);

statistician1.add(6.2);

statistician1.add(4.3);

statistician1.add(2.2);

statistic statistician2(5.5);

statistician2 = statistician1;

statistic statistician3(4.4);

statistician3.add(4.5);

statistician3.add(9.5);

statistician3.add(1.1);

// Perform tests

// Test ==

cout << "Equal to test" << endl;

if(statistician1 == statistician2)

cout << "Stat 1 is equal to Stat 2" << endl;

else

cout << "Stat 1 is not equal to Stat 2" << endl;

if(statistician1 == statistician3)

cout << "Stat 1 is equal to Stat 3" << endl;

else

cout << "Stat 1 is not equal to Stat 3" << endl;

cout << endl;

// Test !=

cout << "Not Equal to test" << endl;

if(statistician1 != statistician2)

cout << "Stat 1 is not equal to Stat 2" << endl;

else

cout << "Stat 1 is equal to Stat 2" << endl;

if(statistician1 != statistician3)

cout << "Stat 1 is not equal to Stat 3" << endl;

else

cout << "Stat 1 is equal to Stat 3" << endl;

cout << endl;

// Test <<

statistician2 = statistician3;

cout << "Stat 1\n" << statistician1 << endl;

cout << "Stat 2\n" << statistician2 << endl;

// Test +

statistician3 = statistician1 + statistician2;

cout << "Stat 3\n" << statistician3 << endl;

return 0;

}

