src/main.cpp

```cpp
// Title:  main.cpp
// Desc:   Testing
// Name:   An Tran

#include "OrderedPair.h"
#include "functions.h"
#include <iostream>

int main(){
    OrderedPair obj1;  // going to be set to (2.2, 3.0)
    OrderedPair obj2(10, 5.5);

    // initalization
    std::cout << "Initalization \n";
    std::cout << "obj1 \nExpected: (0.000000, 0.000000) \nActual:   ";
    obj1.displayPair();
    std::cout << "obj2 \nExpected: (10.000000, 5.500000) \nActual:   ";
    obj2.displayPair();

    std::cout << std::endl;
    // setting
    std::cout << "Setting \n";
    double newXVal = 2.2;
    double newYVal = 3.0;
    obj1.setPair(newXVal, newYVal);
    std::cout << "obj1 \nExpected: (2.200000, 3.000000) \nActual:   ";
    obj1.displayPair();

    std::cout << std::endl;
    // element-wise arithmetic functioning
    std::cout << "element-wise arithmetic functioning \n";
    // add pair
    std::cout << "obj1 \nExpected: 5.2 \nActual:   " << obj1.addPair() << std::endl;

    // subtract pair
    std::cout << "obj1 \nExpected: -0.8 \nActual:   " << obj1.subtractPair() <<
std::endl;

    // multiply pair
    std::cout << "obj1 \nExpected: 6.6 \nActual:   " << obj1.multiplyPair() <<
std::endl;

    // divide pair
    obj1.setY(0.0);
    std::cout << "obj1 \nExpected: yVal==0-1 \nActual:   " << obj1.dividePair() <<
std::endl;
    obj1.setY(3.0);
    std::cout << "obj1 \nExpected: 0.733333 \nActual:   " << obj1.dividePair() <<
std::endl;

    // power pair
    std::cout << "obj1 \nExpected: 10.648 \nActual:   " << obj1.powerPair() <<
std::endl;

```

```
    // pair-wise arithmetic functioning
    OrderedPair combinedObj;
    std::cout << "pair-wise arithmetic functioning \n";

    // add pair
    combinedObj.addPair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (12.200000, 8.500000) \nActual:   ";
    combinedObj.displayPair();

    // subtract pair
    combinedObj.subtractPair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (-7.800000, -2.500000) \nActual:   ";
    combinedObj.displayPair();

    // multiply pair
    combinedObj.multiplyPair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (22.000000, 16.500000) \nActual:   ";
    combinedObj.displayPair();

    // divide pair
    obj2.setY(0.0);
    combinedObj.dividePair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (0.220000, 16.500000) \nActual:   "; //will
keep the previous result becasue it is not being set to anything
    // There should be a display of DIVIDE BY ZERO
    combinedObj.displayPair();
    obj2.setY(5.5);
    combinedObj.dividePair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (0.220000, 0.545455) \nActual:   ";
    combinedObj.displayPair();

    // power pair
    combinedObj.powerPair(obj1, obj2);
    std::cout << "combinedObj \nExpected: (2655.992279, 420.888346) \nActual:   ";
    combinedObj.displayPair();

    // Testing functions
    std::array<float, MAX_SIZE> grades;
    int arrayLength{0};

    std::cout << "Enter grades (up to 50), terminate with -1: \n";
    arrayLength = getInput(grades);
    if (arrayLength > 0) {
        display(grades, arrayLength);
    } else {
        std::cout << "No grades were entered \n";
    };

    grades.fill(0);
    arrayLength = 0;

    std::cout << "Enter the number of random grades to generate (1 to 50): \n";
    std::cin >> arrayLength;
    randInput(grades, arrayLength);
    if (arrayLength > 0) {
        display(grades, arrayLength);
```

```
105     } else {
106         std::cout << "No random grades were generated due to invalid input. \n";
107     }
108 
109     return 0;
110 };
111 
```