Recitation #5, Worksheet - C++ OOP

In groups of 3-4, Understand C++ Classes/Objects:

array.h

```cpp
#ifndef ARRAY_H
#define ARRAY_H

#include <iostream>

using namespace std;

class array {
  //Constructors + Big 3
  public:
    array();
    ~array();
    array(const array&);
    array& operator=(const array&);

  //Public functions
  public:
    float top() const;
    float at(int idx) const;
    void set(int idx, float val);
    void append(float val);
    void remove(int idx);
    void insert(int idx, float val);

  //Getters/Setters
  public:
    int get_size() const;

  //Private Variables
  private:
    float *data;
    int size;
};

void print_array(array& a);

#endif
```

array.cpp

```cpp
//Your TAs will guide you through the process of implementing the array.cpp based on the above array.h and the provided driver.h
```
Pre-Recitation #6 Design – Due Sunday, May 1, 2016 5:00pm on Canvas

Design Plan for Assignment #3:

The process of object-oriented analysis typically includes the following steps:

1. **Identify the classes and objects to be used in the program.**
   Remember, a class is a package that consists of data and procedures that perform operations on the data. In order to determine the classes that will appear in a program, the programmer should think of the major data elements and decide what procedures or actions are required for each class.

2. **Define the attributes for each class.**
   A class’s attributes are the data elements used to describe an object instantiated from the class. They are the values needed for the object to function properly in the program.

3. **Define the behaviors for each class.**
   Once the class’s attributes have been defined, the programmer must identify the activities, or behaviors, each class must be capable of performing. In C++, a class’s behavior is its member functions.
Define the relationships between classes.

The last step in our object-oriented analysis phase is to define the relationships that exist between and among the classes in a program. The possible relationships may be formally stated as

- Access
- Ownership (Composition)
- Inheritance

Informally, these three relationships can be described as

- Uses-a
- Has-a (opposite of Part-of)
- Is-a

The first relationship, access, allows an object to modify the attributes of another object. Normally, an object has attributes not accessible to parts of the program outside the object. These are known as private attributes. An access relationship between two objects means that one object will have access to the other object’s private attributes. When this relationship exists, it can be said that one object uses the other.

The second relationship, ownership, means that one object has another object as one of its members. In OOP terminology, this type of relationship is also called composition. When this relationship exists, it can be said that one object has the other (or one object is part of the other).

The third relationship is inheritance. Sometimes a class is based on another class. This means that one class is a specialized case of the other. When this relationship exists, it can be said that one object is from the other object.

Testing Plan for Assignment #3:

Create a test plan with the test cases (bad, good, and edge cases). What are the expected results?

<table>
<thead>
<tr>
<th>Values</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>You must have a positive number for board dimensions</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Ask the user for the number of columns</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You must submit your design and test plan to your recitation group on Canvas by Sunday, May 1, 2016 5:00pm, and you must provide 3 peer-to-peer reviews before your recitation date and time.