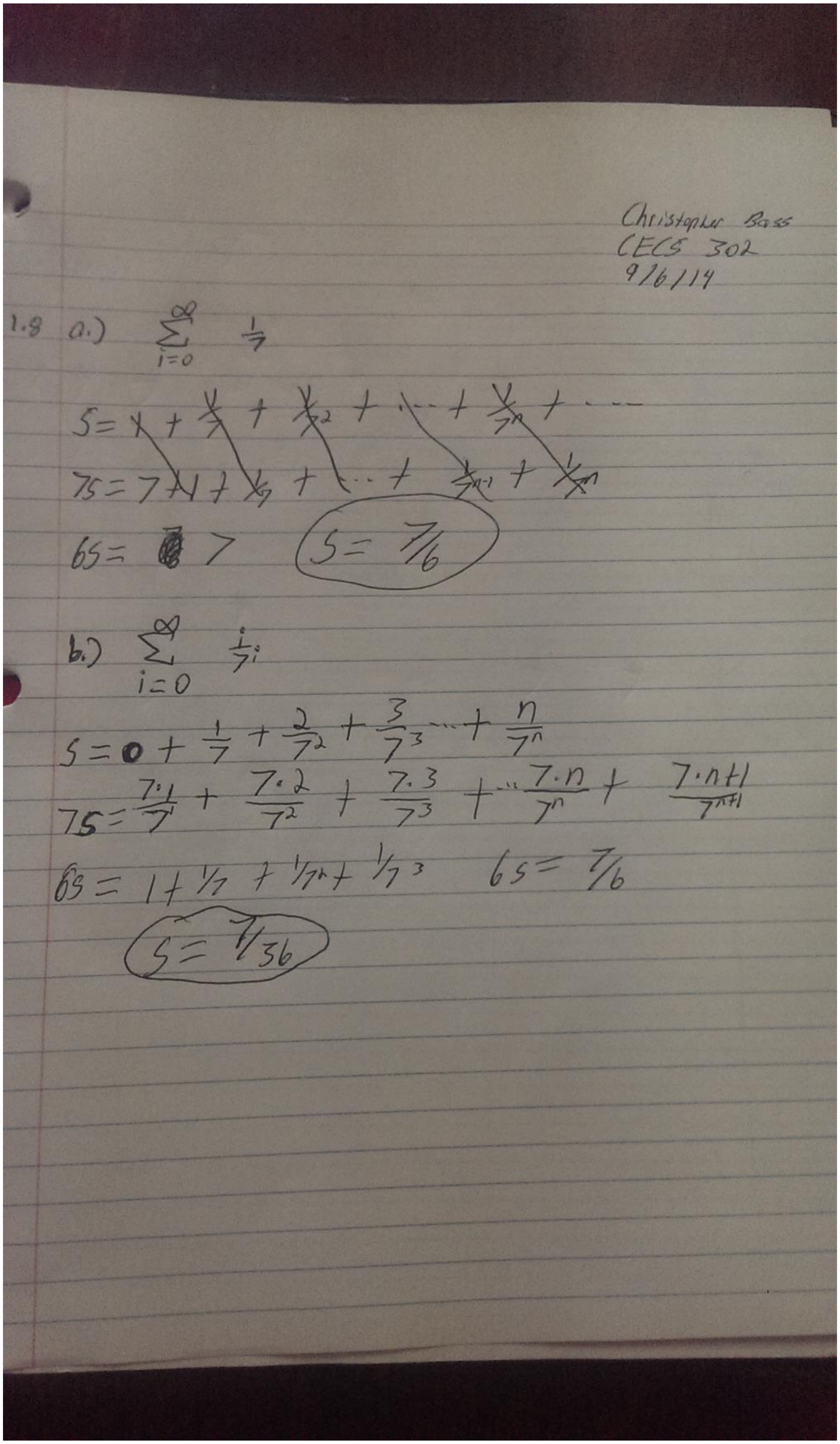
Christopher Bass

CECS 302 50

Assignment 1

9/23/14



Christopher Bass
CECS 302
9/6/14

1.8 a.) $\sum_{i=0}^{\infty} \frac{1}{7}$

$S = \cancel{1} + \cancel{\frac{1}{7}} + \cancel{\frac{1}{7^2}} + \cdots + \cancel{\frac{1}{7^n}} + \cdots$

$7S = 7 + \cancel{1} + \cancel{\frac{1}{7}} + \cdots + \cancel{\frac{1}{7^{n-1}}} + \cancel{\frac{1}{7^n}}$

$6S = 7$ $\boxed{S = 7/6}$

b.) $\sum_{i=0}^{\infty} \frac{i}{7^i}$

$S = 0 + \frac{1}{7} + \frac{2}{7^2} + \frac{3}{7^3} \cdots + \frac{n}{7^n}$

$7S = \frac{7 \cdot 1}{7} + \frac{7 \cdot 2}{7^2} + \frac{7 \cdot 3}{7^3} + \cdots \frac{7 \cdot n}{7^n} + \frac{7 \cdot n+1}{7^{n+1}}$

$6S = 1 + 1/7 + 1/7^2 + 1/7^3$ $6S = 7/6$

$\boxed{S = 7/36}$

(a) $\sum_{i=0}^{\infty} \frac{i^2}{7^i}$

$S = 0 + \frac{1^2}{7} + \frac{2^2}{7^2} + \frac{3^2}{7^3} \cdots + \boxed{\frac{n^2}{7^n}} + \cdots$

$7S = 0 + \cancel{7}\left(\frac{1}{\cancel{7}}\right) + 7\left(\frac{2^2}{7^2}\right) + \cancel{7}\left(\frac{3^2}{7^3}\right) + \cdots + 7\boxed{\frac{n^2}{7^n}}$

$1 + \frac{2^2}{7} + \frac{3^2}{7^2} + \cdots \frac{n^2}{7^{n-1}} + \boxed{\frac{(n+1)^2}{7^n}}$

$6S = 1 + \frac{2^2-1^2}{7} + \frac{3^2-2^2}{7^2} + \cdots + \frac{(n+1)^2-n^2}{7^n} + \cdots$

$= 1 + \frac{\cancel{(2-1)}(2+1)}{7} + \frac{(3-2)(3+2)}{7^2} + \cdots + \frac{(n+1-n)(n+1+n)}{7^n}$

$= 1 + \frac{3}{7} + \frac{5}{7^2} + \cdots + \frac{2n+1}{7^n}$

$= 1 + \frac{2+1}{7} + \frac{2\cdot 2+1}{7^2} + \cdots + \frac{2n+1}{7^n}$

$= 1 + \frac{2}{7} + \frac{2\cdot 2}{7^2} + \cdots + \frac{2n}{7^n}$

$\frac{1}{7} + \frac{1}{7^2} + \cdots + \frac{1}{7^n} + \cdots$

$= 7/6 + 2\left[\frac{1}{7} + \frac{2}{7^2} + \cdots + \frac{n}{7^n} + \cdots\right]$

$6S = 7/6 + 2(7/36)$

$= 7\left[1/6 + 1/18\right]$

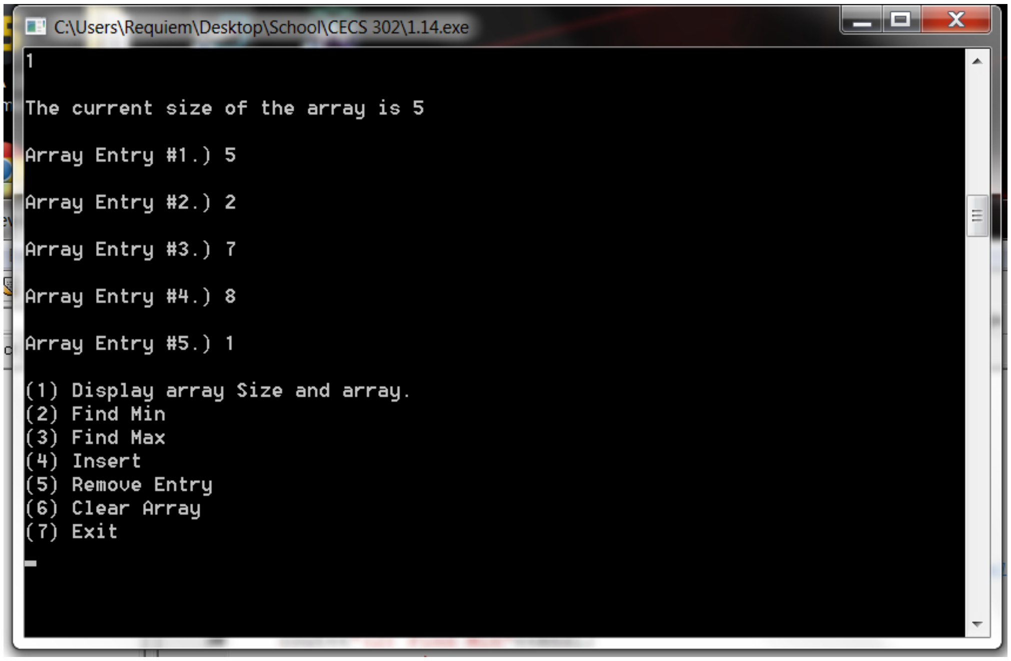
$6S = 7\left(\frac{3+1}{18}\right) \quad S = \frac{7}{6} \cdot \frac{\cancel{4}\,2}{\cancel{18}\,9}$

$S = \boxed{\frac{14}{54}} = \boxed{\frac{7}{27}}$

1.14 pg 48

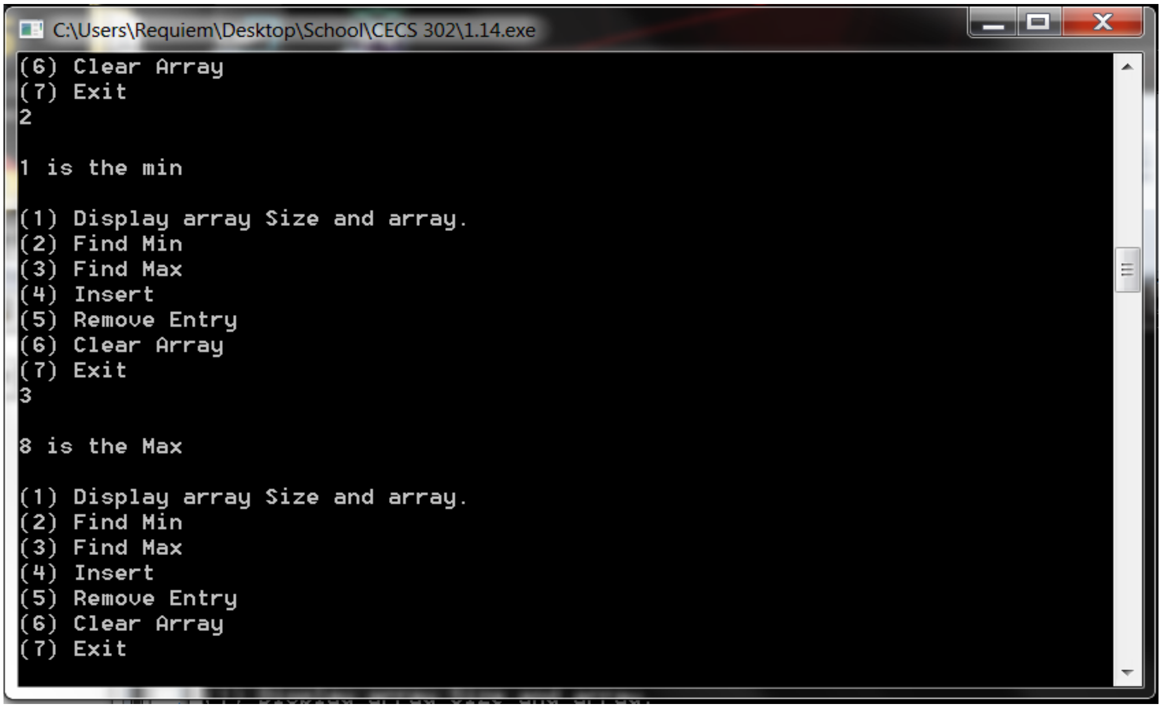
“*Design a class template OrderedCollection, that stores a collection of Comparables (in an array), along with the current size of the collection. Provide public functions isEmpty, makeEmpty, insert, remove, findMin, and findMax return references to the smallest and largest respectively, comparable in the collection. Explain what can be done if these operations are performed on an empty collection.”*

When the program is first run it will not have an empty array. The user will have to use the Insert function by selecting number 4 to add numbers to the array, which I have demonstrated having already being done and then used the display array and size function to display them. If the array is empty there is a check for each function that will see what the size of the array is and if it is empty it will simply display that the array is empty.



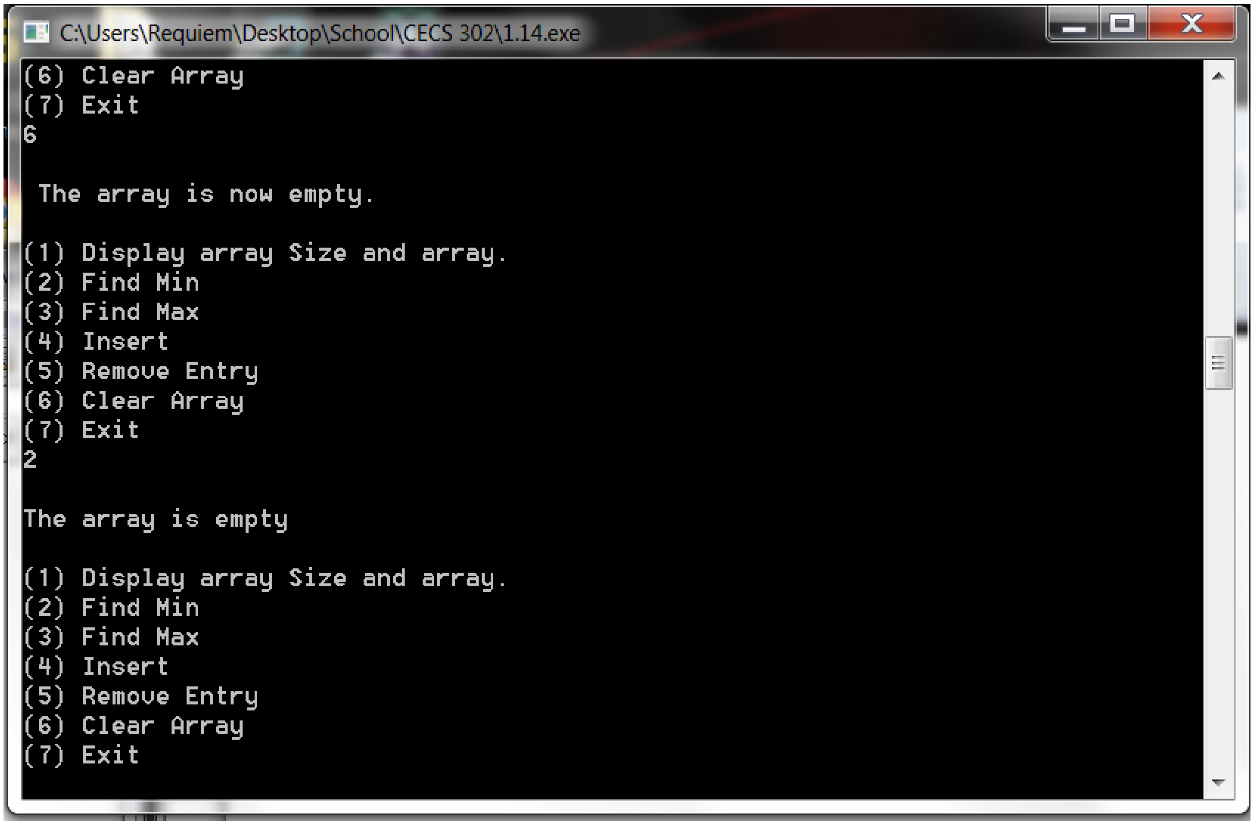
```
1

The current size of the array is 5

Array Entry #1.) 5

Array Entry #2.) 2

Array Entry #3.) 7

Array Entry #4.) 8

Array Entry #5.) 1

(1) Display array Size and array.
(2) Find Min
(3) Find Max
(4) Insert
(5) Remove Entry
(6) Clear Array
(7) Exit
_
```

Below shows the findMin and findMax functions with the above inputs.



```
(6) Clear Array
(7) Exit
2

1 is the min

(1) Display array Size and array.
(2) Find Min
(3) Find Max
(4) Insert
(5) Remove Entry
(6) Clear Array
(7) Exit
3

8 is the Max

(1) Display array Size and array.
(2) Find Min
(3) Find Max
(4) Insert
(5) Remove Entry
(6) Clear Array
(7) Exit
```

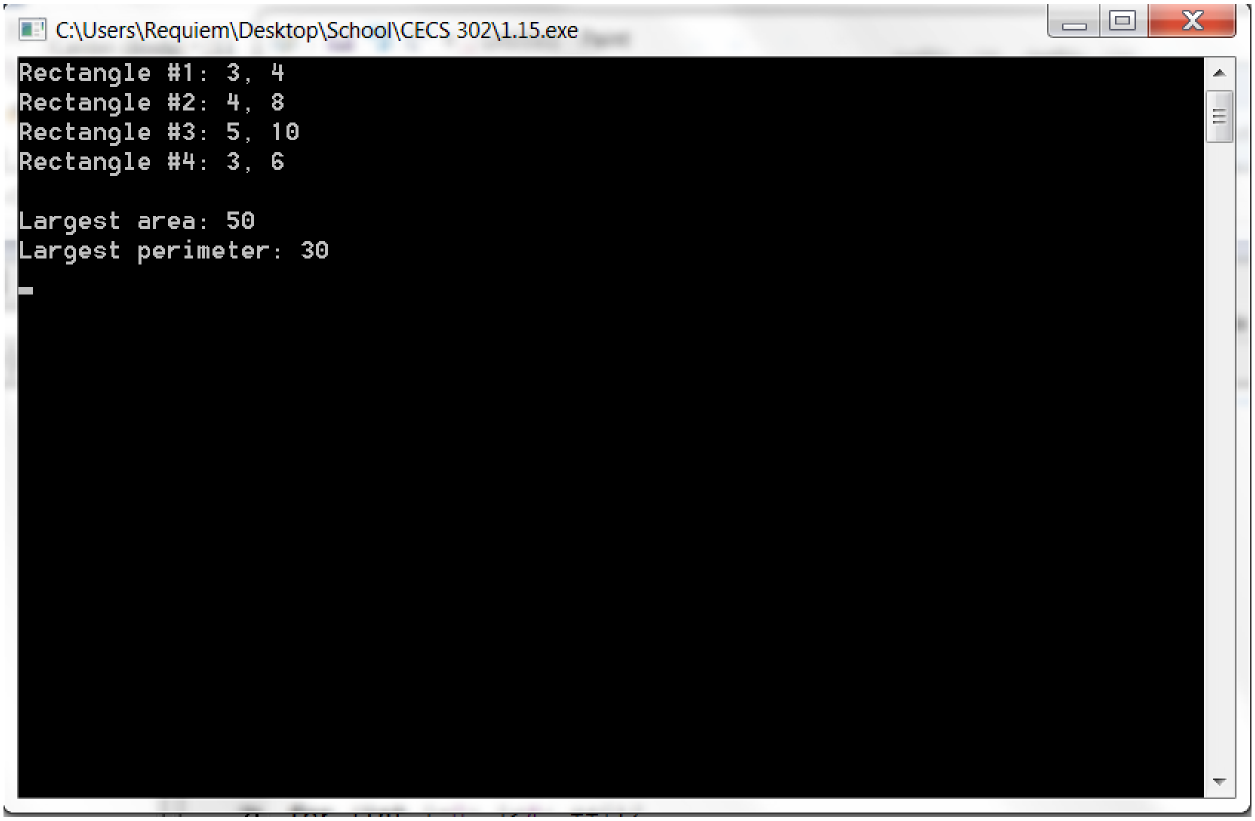
Below shows the makeEmpty function being used and then what happens when you try to use one of the functions on an empty array.



```
(6) Clear Array
(7) Exit
6

 The array is now empty.

(1) Display array Size and array.
(2) Find Min
(3) Find Max
(4) Insert
(5) Remove Entry
(6) Clear Array
(7) Exit
2

The array is empty

(1) Display array Size and array.
(2) Find Min
(3) Find Max
(4) Insert
(5) Remove Entry
(6) Clear Array
(7) Exit
```

1.15 pg 48

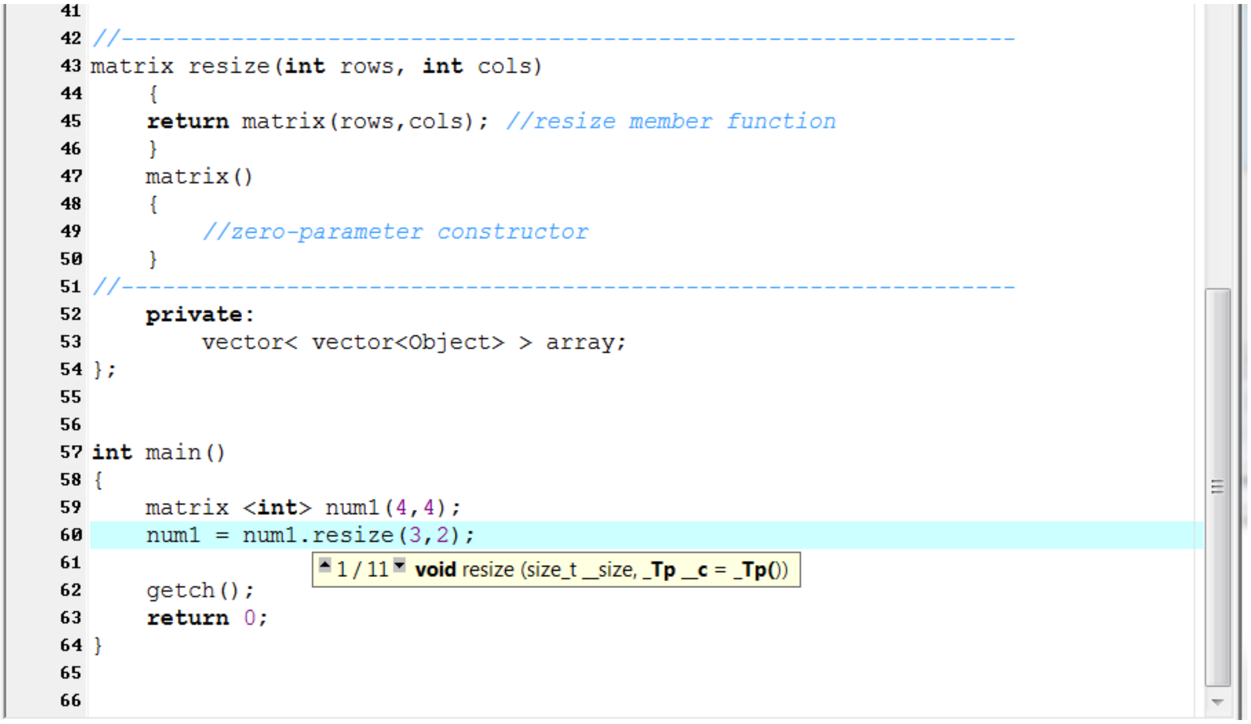
“Define a Rectangle class that provides getLength and getWidth. Using the findMax routines in figure 1.25, write a main that creates an array of Rectangle and finds the largest Rectangle first on the basis of area and then on the basis of perimeter.”



```
Rectangle #1: 3, 4
Rectangle #2: 4, 8
Rectangle #3: 5, 10
Rectangle #4: 3, 6

Largest area: 50
Largest perimeter: 30
```

1.16 pg 48

“For the matrix class, add a resize member function and a zero-parameter constructor.”



```
//-----------------------------------------------------------------
matrix resize(int rows, int cols)
    {
    return matrix(rows,cols); //resize member function
    }
    matrix()
    {
        //zero-parameter constructor
    }
//-----------------------------------------------------------------
    private:
        vector< vector<Object> > array;
};


int main()
{
    matrix <int> num1(4,4);
    num1 = num1.resize(3,2);

    getch();
    return 0;
}
```