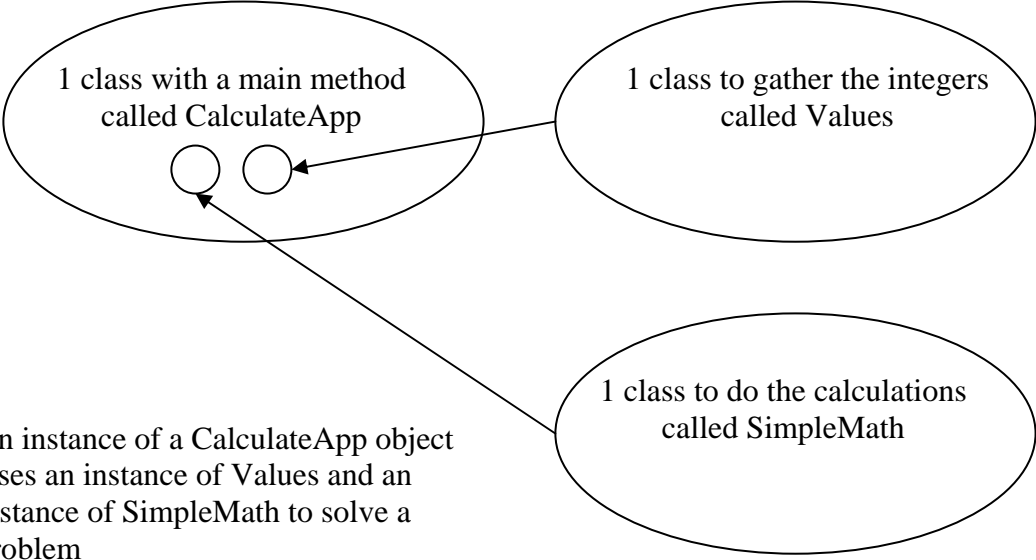


SimpleMath	book assignment 2.17
<p>Here is one example from a student this semester.</p> <p>It is a fine example, but another student asked how could we divide this problem into its basic components into several classes?</p> <p>This exercise will create one solution that divides this problem into components and brings them together in a test client or application.</p>	<pre> /*IUP COSC 210 ***** Students name hidden 9/10/2008 This program does many math operations. It inputs three integers from the user using Scanner and then does operations to display the sum, average, product, smallest and largest numbers. Numbers are not in floating point, aka one no decimal points. */ import java.util.Scanner; // needed to use Scanner public class Math { public static void main(String[] args) { Scanner input = new Scanner(System.in); int number1; // all the variables int number2; int number3; int sum; int average; int product; System.out.println("Welcome to a math program."); System.out.println("Please enter the first integer: "); number1 = input.nextInt(); // user input goes here System.out.println("Please enter the second integer: "); number2 = input.nextInt(); // user input goes here System.out.println("Please enter the third integer: "); number3 = input.nextInt(); // user input goes here sum = number1 + number2 + number3; //sum System.out.printf("%d", sum); System.out.println(" is the sum."); average = (number1 + number2 + number3) / 3; // average System.out.printf("%d", average); System.out.println(" is the average."); product = number1 * number2 * number3; //product System.out.printf("%d", product); System.out.println(" is the product."); // finding smallest number if (number1 <= number2) if (number1 <= number3) { System.out.printf("%d", number1); System.out.print(" (the first number) is the smallest number.\n"); } if (number2 <= number1) if (number2 <= number3) System.out.printf("%d", number2); if (number2 <= number1) if (number2 <= number3) </pre>

```
        System.out.print( " (the second number) is the smallest number.\n");
    if (number3 <= number1)
        if (number3 <= number2)
        {
            System.out.printf( "%d", number3);
            System.out.print( " (the third number) is the smallest number.\n");
        } // note: it works with a double double if or with brackets

//finding largest number
if (number1 >= number2)
    if (number1 >= number3)
    {
        System.out.printf( "%d", number1);
        System.out.print( " (the first number)is the largest number.\n");
    }
if (number2 >= number1)
    if (number2 >= number3)
    {
        System.out.printf( "%d", number2);
        System.out.print( " (the second number)is the largest number.\n");
    }
if (number3 >= number2)
    if (number3 >= number1)
    {
        System.out.printf( "%d", number3);
        System.out.print( " (the third number)is the largest number.\n");
    }

// checking to see if they are the same
if (number1 == number2)
    {
        System.out.printf( "%d", number1);
        System.out.print( "the first number is the same as the second number");
        System.out.printf( "%d\n",number2);
    }
if (number1 == number3)
    {
        System.out.printf( "%d", number1);
        System.out.print( "the first number is the same as the third number");
        System.out.printf( "%d\n",number3);
    }
if (number3 == number2)
    {
        System.out.printf( "%d", number2);
        System.out.print( "the second number is the same as the third number");
        System.out.printf( "%d\n",number3);
    }
}
```

<p>How would we start?</p> <p>What are the main items needed for our solution?</p> <p>How many classes do we want to use?</p> <p>I decided to use 3...could be done many ways this is one.</p>	 <p>An instance of a CalculateApp object Uses an instance of Values and an instance of SimpleMath to solve a problem</p>
<p>The Values class</p>	<p>1) Let us decide to have the Values class prompt the user for the integers. What else?</p> <ol style="list-style-type: none"> 2) Let's store our integers in an array of integers. 3) Let's have values not have a limit on the number of Integers. 4) Let's have methods to get at our array of Integers. <p>To begin:</p> <pre>public class Values {</pre>
<p>1) Let us decide to have the Values class prompt the user for the integers.</p>	<pre>import java.util.Scanner;</pre> <pre>public class Values {</pre> <pre> Scanner input = new Scanner(System.in);</pre> <pre>}</pre> <p>We know we are going to need a Scanner so import it and declare a Scanner object with the identifier input. We will use the input object later.</p>
<p>2) Lets store our integers in an array of integers.</p>	<pre>import java.util.Scanner;</pre> <pre>public class Values {</pre> <pre> Scanner input = new Scanner(System.in);</pre> <pre> private int values [] = null;</pre> <pre>}</pre> <p>We know we are going to need an array of integers so declare one with the identifier values. We will use the values array later.</p>
<p>3) Lets have our solution not have a limit on the number of entries.</p>	<p>Notice the integer array is set equal to null. Lets remember to initialize the array to a size equal to the number of integers a user decides on.</p>

<p>4) Lets have methods to get at our array of Integers.</p>	<pre>import java.util.Scanner; public class Values { Scanner input = new Scanner(System.in); private int values [] = null; private void createValues(){ System.out.println("How many values would you like to enter today?"); int counter = input.nextInt(); values = new int[counter]; //here is where we initialize our array for (int x=0;x<counter;x++){ System.out.println("Enter a value please"); values[x]=input.nextInt(); } } public int[] getValues(){ return values; } }</pre> <p>Add a method called CreateValues(), it does not return anything and takes no parameters.</p> <ul style="list-style-type: none"> - prompt the user for how many integers they want to enter - store the users entry into an integer called counter - use counter to initialize the int array - loop a number of times equal to counter and prompt the user saving their entry into the array <p>Add another method called getValues(), it returns an array of int and does not take any parameters.</p> <ul style="list-style-type: none"> - this method returns the array of int <p>There is a problem here if we initialize a Values object and try to invoke the getValues() method the array is still null. This will produce a NullPointerException or something to that affect. We will solve this in the next step.</p>
<p>5) Lets fix the NullPointerException problem.</p>	<p>We want to make sure that any time someone initializes a Values object, they initialize an array full of Integers. How do we do this?</p> <p>Add logic to the constructor for Values to take care of it.</p> <pre>import java.util.Scanner; public class Values { Scanner input = new Scanner(System.in); private int values [] = null; public Values() { this.createValues(); } private void createValues(){ System.out.println("How many values would you like to enter today?"); int counter = input.nextInt();</pre>

	<pre> values = new int[counter]; for (int x=0;x<counter;x++){ System.out.println("Enter a value please"); values[x]=input.nextInt(); } public int[] getValues(){ return values; } </pre>
The SimpleMath class	<p>1) Let us decide to have the SimpleMath class have a method for each of the calculations. 2) calculateSum returns an integer and takes an integer array parameter. 3) calculateAverage returns an double and takes an integer array parameter. 4) calculateProduct returns an integer and takes an integer array parameter. 5) calculateSmallest returns an integer and takes an integer array parameter. 6) calculateLargest returns an integer and takes an integer array parameter.</p> <p>For product, I decided to allow the decimal portion.</p> <p>To begin:</p> <pre> public class SimpleMath { } </pre>
We will need some instance variables to contain the sum, average, product, smallest, largest values.	<pre> public class SimpleMath { int sum = 0; double average = 0.0; int product = 1; int smallest = 0; int largest = 0; } </pre>
Add the calculateSum method	<pre> public class SimpleMath { int sum = 0; double average = 0.0; int product = 1; int smallest = 0; int largest = 0; public int calculateSum(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ sum = sum + inValues[x]; } return sum; } } </pre> <p>The method takes an int array in and returns a single int.</p> <ul style="list-style-type: none"> - figure out the length of the array and store it - loop through the array and calculate the sum - return the sum

<p>Add the calculateAverage method</p>	<pre>public class SimpleMath { int sum = 0; double average = 0.0; int product = 1; int smallest = 0; int largest = 0; public int calculateSum(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ sum = sum + inValues[x]; } return sum; } public double calculateAverage(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ average = average + inValues[x]; } return average/len; } }</pre> <p>The method takes an int array in and returns a single double.</p> <ul style="list-style-type: none"> - figure out the length of the array and store it - loop through the array and calculate the sum - return the average by dividing the sum by the length of the array
<p>Add the calculateProduct method</p>	<pre>public class SimpleMath { int sum = 0; double average = 0.0; int product = 1; int smallest = 0; int largest = 0; public int calculateSum(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ sum = sum + inValues[x]; } return sum; } public double calculateAverage(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ average = average + inValues[x]; } return average/len; } public int calculateProduct(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ product = product * inValues[x]; } } }</pre>

	<pre> return product; } } </pre> <p>The method takes an int array in and returns a single int.</p> <ul style="list-style-type: none"> - figure out the length of the array and store it - loop through the array and calculate the product - return the product
<p>Add the calculateSmallest method</p>	<pre> public class SimpleMath { int sum = 0; double average = 0.0; int product = 1; int smallest = 0; int largest = 0; public int calculateSum(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ sum = sum + inValues[x]; } return sum; } public double calculateAverage(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ average = average + inValues[x]; } return average/len; } public int calculateProduct(int[] inValues){ int len = inValues.length; for (int x=0;x<len;x++){ product = product * inValues[x]; } return product; } public int calculateSmallest(int[] inValues){ int len = inValues.length; smallest = inValues[0]; for (int x=1;x<len;x++){ if (inValues[x] < smallest) smallest = inValues[x]; } return smallest; } } </pre> <p>The method takes an int array in and returns a single int.</p> <ul style="list-style-type: none"> - figure out the length of the array and store it - set the variable smallest to the first value in the array - loop through the array starting with the second value - if the value in the array is smaller than smallest replace smallest with the value in the array - return smallest

**Add
calculateLargest
method**

```

public class SimpleMath {
    int sum = 0;
    double average = 0.0;
    int product = 1;
    int smallest = 0;
    int largest = 0;

    public int calculateSum(int[] inValues){
        int len = inValues.length;
        for (int x=0;x<len;x++){
            sum = sum + inValues[x];
        }
        return sum;
    }

    public double calculateAverage(int[] inValues){
        int len = inValues.length;
        for (int x=0;x<len;x++){
            average = average + inValues[x];
        }
        return average/len;
    }

    public int calculateProduct(int[] inValues){
        int len = inValues.length;
        for (int x=0;x<len;x++){
            product = product * inValues[x];
        }
        return product;
    }

    public int calculateSmallest(int[] inValues){
        int len = inValues.length;
        smallest = inValues[0];

        for (int x=1;x<len;x++){
            if (inValues[x] < smallest)
                smallest = inValues[x];
        }
        return smallest;
    }

    public int calculateLargest(int[] inValues){
        int len = inValues.length;
        largest = inValues[0];

        for (int x=1;x<len;x++){
            if (inValues[x] > largest)
                largest = inValues[x];
        }
        return largest;
    }
}

```

The method takes an int array in and returns a single int.

- figure out the length of the array and store it
- set the variable largest to the first value in the array
- loop through the array starting with the second value

	<ul style="list-style-type: none"> - if the value in the array is larger than largest replace largest with the value in the array - return largest
The CalculateApp class	<p>1) Let us decide to have the CalculateApp class prompt the user for whether they want to do Math. What else?</p> <ol style="list-style-type: none"> 2) Initialize a SimpleMath object. 3) If they want to do Math, initialize a Values object. 4) Get the values from your new Values object. 5) Run the calculate methods on those values. <p>To begin:</p> <pre>public class CalculateApp { }</pre>
1) have the CalculateApp class prompt the user for whether they want to do Math.	<pre>import java.util.Scanner; public class CalculateApp { public static void main(String[] args) { Scanner inYes = new Scanner(System.in); System.out.println("would you like to do Math with some integers?"); System.out.println("Enter y for yes, anything else for no please."); char input = inYes.nextLine().charAt(0); } }</pre>
2) Initialize a SimpleMath object.	<pre>import java.util.Scanner; public class CalculateApp { public static void main(String[] args) { SimpleMath calculateIt = new SimpleMath(); Scanner inYes = new Scanner(System.in); System.out.println("would you like to do Math with some integers?"); System.out.println("Enter y for yes, anything else for no please."); char input = inYes.nextLine().charAt(0); } }</pre>
3) If they want to do Math, initialize a Values object.	<pre>import java.util.Scanner; public class CalculateApp { public static void main(String[] args) { SimpleMath calculateIt = new SimpleMath(); Scanner inYes = new Scanner(System.in); System.out.println("would you like to do Math with some integers?"); System.out.println("Enter y for yes, anything else for no please."); char input = inYes.nextLine().charAt(0); if (input=='y'){ Values newValues = new Values(); } } }</pre>

<p>4) Get the values from your new Values object.</p>	<pre>import java.util.Scanner; public class CalculateApp { public static void main(String[] args) { SimpleMath calculateIt = new SimpleMath(); Scanner inYes = new Scanner(System.in); System.out.println("would you like to do Math with some integers?"); System.out.println("Enter y for yes, anything else for no please."); char input = inYes.nextLine().charAt(0); if (input=='y'){ Values newValues = new Values(); int calculateValues[] = newValues.getValues(); } else { System.out.println("Thank you, anyway"); } } }</pre>
<p>5) Run the calculate methods on those values.</p>	<pre>import java.util.Scanner; public class CalculateApp { public static void main(String[] args) { SimpleMath calculateIt = new SimpleMath(); Scanner inYes = new Scanner(System.in); System.out.println("would you like to do Math with some integers?"); System.out.println("Enter y for yes, anything else for no please."); char input = inYes.nextLine().charAt(0); if (input=='y'){ Values newValues = new Values(); int calculateValues[] = newValues.getValues(); System.out.println("The common calculations for those numbers are:"); System.out.print("Sum: "); System.out.println(calculateIt.calculateSum(calculateValues)); System.out.print("Product: "); System.out.println(calculateIt.calculateProduct(calculateValues)); System.out.print("Average: "); System.out.println(calculateIt.calculateAverage(calculateValues)); System.out.print("Smallest: "); System.out.println(calculateIt.calculateSmallest(calculateValues)); System.out.print("Largest: "); System.out.println(calculateIt.calculateLargest(calculateValues)); System.out.println(" "); System.out.println("The numbers were: "); for (int y=0;y<calculateValues.length;y++){ System.out.print(calculateValues[y] + " "); } } else { System.out.println("Thank you, anyway"); } } }</pre>