in class day 1

AP CS Unit 5.1: Static Variables and Methods Programs

1. Write a program wherein:
   - Two Gamer objects are instantiated.
   - The goal is to get 100 points or more to win. Therefore you must set ptsToWin to 100.
   - Write a loop where the two Gamer objects each keep getting random amount of points until one or both of them have won.
   - After the loop, simply call the toString method and display the results for each object.

```
public class Gamer {
   private static int ptsToWin;
   private String name;
   private int pts;

   public Gamer( String s ) {
       name = s;
       pts = 0;
   }

   public static void setGoal( int n ){
       ptsToWin = n;
   }

   public void addPts(){
       pts += (int)(8*Math.random());
   }

   public boolean won() {
       if ( pts >= ptsToWin )
           return true;
       else
           return false;
   }

   public String toString() {
       return name + " has " + pts;
   }
}
```

hw day 1

4. Write a program where the user enters a positive, odd integer and a shape similar to those below is displayed. You will need to use nested loops. You may use the Scanner class ~~or the JOptionPane class~~ to get the inputs.

If the number is 1 then	If the number is 7 then	If the number is 15 then
o	ooooooo .ooooo. ..ooo.. ...o...	ooooooooooooooo .ooooooooooooo. ..ooooooooooo.. ...ooooooooo... ....ooooooo.... .....ooooo..... ......ooo...... ....o.......

5. Write a program where the user enters a String with a length greater than one. Then the program prints out the string in the following pattern.

If the String is *hi* then	If the String is *Left* then	If the String is *Monday* then
h	L e f t L e f t L	M o n d a y M o n d a y M o n d a y M o n d a y M

6. Write a program where the user enters a positive integer. Then the program prints out the following pattern.

If the number is 1 then	If the number is 4 then	If the number is 9 then

1	1 2 3 4
	2 3 4
	3 4
	4
		1 2 3 4 5 6 7 8 9
		2 3 4 5 6 7 8 9
		3 4 5 6 7 8 9
		4 5 6 7 8 9
		5 6 7 8 9
		6 7 8 9
		7 8 9
		8 9
		9

AP CS Unit 5a: Arrays Programs

1. Write a program that does the following:

- Prompts the user to enter a positive integer *n*. make this a static method
- Creates an array of length *n* and fills it with random integers between 0 and n (inclusive)
- Displays the contents of the array – method
- Displays the minimum, average, and maximum values of the array – method

Here is a sample run:

```
Enter a positive integer
7
4   1   1   2   3   3   6
Min value: 1
Max value: 6
Average: 2.857142857142857
```

2. Write a program that does the following:

- Prompts the user to enter a positive integer *n*.
- Creates a String array of length n and prompts the user to enter n Strings. Store each String in lower-case.
- Then prompts the user to a letter to search for. Convert that letter to lower-case.
- Then displays the number of times that letter appears in all the strings in the array.

Your program should follow this outline (which uses a helper method).

public class RunStringArray{
    public static void main(String [] args){
        Get the array size.
        Create the array and enter the user's inputs.
        Get the letter to search for.

Use the helper method below to count all the occurrences of that letter in the array. Display the result.

    }

    private static int count(String str, String letter){
        Returns the number of times that letter appears in str.
    }
}

Here is what a sample run should look like. Sample inputs are in italics.

Enter a positive integer
    4
Enter a string
    Every evening
Enter a string
    a fish
Enter a string
    fries
Enter a string
    French fries
What letter do you want to count?
    E

The letter e was found 7 times.

3. Write a program that does the following:

- Creates an array of 5 Point objects. Each Point should have a random x and y coordinate between 0 and 10 (inclusive).
- Display the points and the distances between each point as shown in the example below. Use the Point class's distance method to find the distances.

Notes:
(1) Round the distances to nearest int (don't just cast)
(2) Use \t to align the values
(3) Use this helper method to round the distances to one decimal place

```
// assumes x is positive and less than Integer.MAX_VALUE
private static double roundOnePlace( double x ){
        int num = (int) ( 10*x + 0.5 );
        return num / 10.0;
}
```

(4) Here is a snippet of code to refresh your memory about the Point class

```
import java.awt.Point;
...
Point p1 = new Point( 4, 6 );
Point p2 = new Point( 5, 7 );
double d = p1.distance( p2 );
System.out.println( p1.x +", " + p1.y);
```

Here is a sample output.

```
                        P1      P2      P3      P4    p5
                        10,6    4,1     2,0     4,2   8,2
P1 10,6                 0.0     7.8     10.0    7.2   4.5
P2 4,1                  7.8     0.0     2.2     1.0   4.1
P3 2,0                  10.0    2.2     0.0     2.8   6.3
P4 4,2                  7.2     1.0     2.8     0.0   4.0
P5 8,2                  4.5     4.1     6.3     4.0   0.0
```

4. Complete the program below.

```
public class Run2DArray{
        public static void main( String [] args ){
                int [][] table = makeTable();
                displayTable( table );
                for ( int k = 0; k < table.length; k++ ) {
                        int max = findMax( table[k] );
                        System.out.println( "Max value in row " + k + " is " + max );
                }
        }

        /* Write the makeTable method.  This returns a 2D array of ints.  The number of rows
and columns are random numbers between 2 and 5 (inclusive).  Fill the array with random values
between -10 and +5 (inclusive).  */

        /* Write the displayTable method.  It prints the contents of the array.  Each row is on a
separate line and each column is separated by a tab.  */

        /* Write the findMax method.  This returns the largest value in an array.  */
}
```

Here are two sample outputs.

```
-14     -14     2       -1      -12
-10     -7      -5      0       -14
-13     4       -15     -7      -2
-5      -6      -12     -8      -7
-9      -1      2       -4      -9
Max value in row 0 is 2
Max value in row 1 is 0
Max value in row 2 is 4
Max value in row 3 is -5
Max value in row 4 is 2
```

```
-5      -12
4       -7
3       -13
2       -14
Max value in row 0 is -5
Max value in row 1 is 4
Max value in row 2 is 3
Max value in row 3 is 2
```

5. Write another class with a main method. Create a 3 by 3 array of wombats. The happy/sad thing should be random.

Then loop through the array and call the toString method for each. Then count how many wombats are happy and their total weight. Do the same for the sad wombats. The results should be different each time you run it. Here is one sample output.

```
Happy/10  Happy/23  Happy/15
Happy/13  Happy/24  Sad / 35
Sad / 27  Happy/13  Sad / 14

There is/are 6 happy wombat(s).
Total weight is 98 lbs.

There is/are 3 sad wombat(s).
Total weight is 76 lbs.
```

```
public class Wombat{
    private int weight;   // kg
    private boolean happy;

    public Wombat( boolean b ){
        weight = (int)(26*Math.random()) + 10;
        happy = b;
    }

    public int getWt(){
        return weight;
    }

    public boolean getHappy(){
        return happy;
    }

    public String toString(){
        if ( happy )
            return "Happy/" + weight;
        else
            return "Sad / " + weight;
    }
}
```

6. This is a very short, optional exercise. The Arrays class contains a collection of class methods for working with one-dimensional arrays. Go to the java api and look up the toString and sort methods in the Arrays class. Use this information to complete the missing lines in this program.

```
import java.util.Arrays;

public class RunArraysClass{
        public static void main( String [] args ){
                int [] a = { 6, 3, 7, 5, 5, -2, 0, 4 };
                String str = call the toString method
                System.out.println( str );            // displays  [6, 3, 7, 5, 5, -2, 0, 4]
                call the sort method
                str = call the toString method
                System.out.println( str );            // displays  [-2, 0, 3, 4, 5, 5, 6, 7]

        }
}
```

The Arrays class is not part of the AP curriculum. I mention it here because (1) it reinforces how to use class methods and (2) it's useful.