Activity 5

1. Write a program FourChargeClient.java that takes a `double` command-line argument `r`, creates four `Charge` objects that are each distance `r` from the center of the screen (.5, .5), and prints the potential at location (.25, .5) due to the combined four charges. All four charges should have the same unit charge.

```
ant -f C:\\java\\Section1.1 -Dapplication.args=.2 run
init:
deps-jar:
Updating property file: C:\java\Section1.1\build\built-jar.properties
compile:
run:
total potential = 2.5593786255292056E11
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Write a program Hex2Decimal.java that converts from a hexadecimal string (using A-F for the digits 11-15) to decimal.

   Answer: the following solution uses several string library methods and Horner's method.

```
public static int hex2decimal(String s) {
   String digits = "0123456789ABCDEF";
   s = s.toUpperCase();
   int val = 0;
   for (int i = 0; i < s.length(); i++) {
      char c = s.charAt(i);
      int d = digits.indexOf(c);
      val = 16*val + d;
   }
   return val;
}
```

   Alternate solution: `Integer.parseInt(String s, int radix)`. More robust, and works with negative integers.

```
ant -f C:\\java\\Section1.1 -Dapplication.args=1f run
init:
Deleting: C:\java\Section1.1\build\built-jar.properties
deps-jar:
Updating property file: C:\java\Section1.1\build\built-jar.properties
compile:
run:
Decimal: 31
Hexa:1F
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. **VIN numbers.** A VIN number is a 17-character string that uniquely identifies a motor vehicle. It also encodes the manufacturer and attributes of the vehicle. To guard against accidentally entering an incorrect VIN number, the VIN number incorporates a check digit (the 9th character). Each letter and number is assigned a value between 0 and 9. The check digit is chosen so to be the weighted sum of the values mod 11, using the symbol X if the remainder is 10.

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 2 3 4 5 6 7 8 - 1 2 3 4 5 - 7 - 9 2 3 4 5 6 7 8 9

1st 2nd 3rd 4th 5th 6th 7th 8th 9th 10  11  12  13  14  15  16  17
 8   7   6   5   4   3   2  10   0   9   8   7   6   5   4   3   2
```

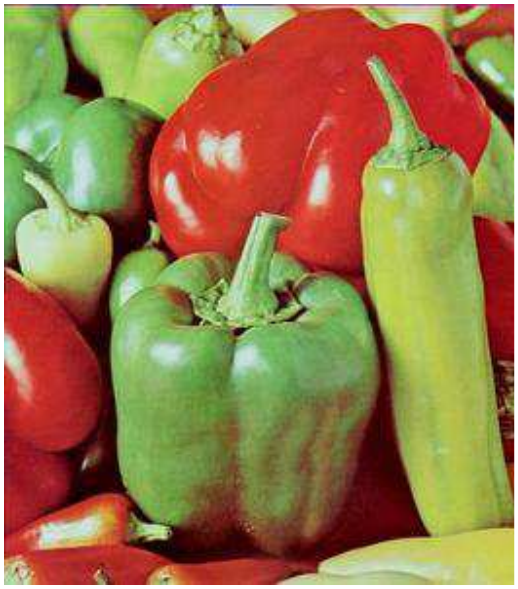
For example the check digit of the partial VIN number 1FA-CP45E-?-LF192944 is X because the weighted sum is 373 and 373 mod 11 is 10.

```
 1   F   A   C   P   4   5   E   X   L   F   1   9   2   9   4   4
 1   6   1   3   7   4   5   5   -   3   6   1   9   2   9   4   4
 8   7   6   5   4   3   2  10   -   9   8   7   6   5   4   3   2
-----------------------------------------------------------------------
 8  42   6  15  28  12  10  50   -  27  48   7  54  10  36  12   8
```

Write a program VIN.java that takes a command line string and determines whether or not it is a valid VIN number. Allow the input to be entered with upper or lower case, and allow dashes to be inserted. Do thorough error checking, e.g., that the string is the right length, that no illegal characters are used (I, O, Q), etc.

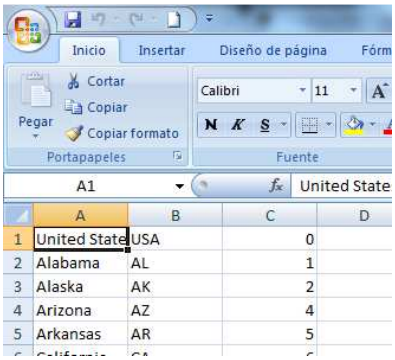
```
ant -f C:\\java\\Section1.1 -Dapplication.args=1B4YEM9P4KP186543 run
init:
Deleting: C:\java\Section1.1\build\built-jar.properties
deps-jar:
Updating property file: C:\java\Section1.1\build\built-jar.properties
Compiling 1 source file to C:\java\Section1.1\build\classes
compile:
run:
Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. **Flip horizontally.** Write a program FlipX.java that takes a command line argument which is the name of a JPG or PNG file, displays it in a window, flips the image horizontally, and displays the resulting image in the window. We illustrate using standard computer graphics test images - baboon.jpg and peppers.jpg.



5. **Color separation.** Write a program ColorSeparation.java that takes the name of an image file as a command line input, and creates three images, one that contains only the red components, one for green, and one for blue.

6. **wget.** Write a program Wget.java that takes the name of a URL as a command-line argument and saves the referenced file using the same filename.



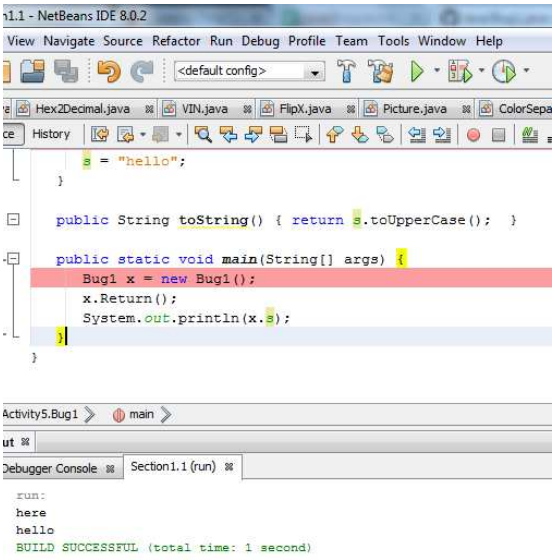
7. **Capitalize.** Write a program Capitalize.java that reads in text from standard input and capitalizes each word (make first letter uppercase and make the remaining letters lowercase).

```
ant -f C:\\java\\Section1.1 "-Dapplication.args=now is the time for all good" debug
init:
Deleting: C:\java\Section1.1\build\built-jar.properties
deps-jar:
Updating property file: C:\java\Section1.1\build\built-jar.properties
Compiling 1 source file to C:\java\Section1.1\build\classes
compile:
Now Is The Time For All Good
debug:
BUILD SUCCESSFUL (total time: 2 minutes 50 seconds)
```

8. Why does program Bug1.java create a `java.lang.NullPointerException` when executed?

```
public class Bug1 {
   private String s;
   public void Bug1()       { s = "hello"; }
   public String toString() { return s.toUpperCase(); }
   public static void main(String[] args) {
      Bug1 x = new Bug1();
      StdOut.println(x);
   }
}
```

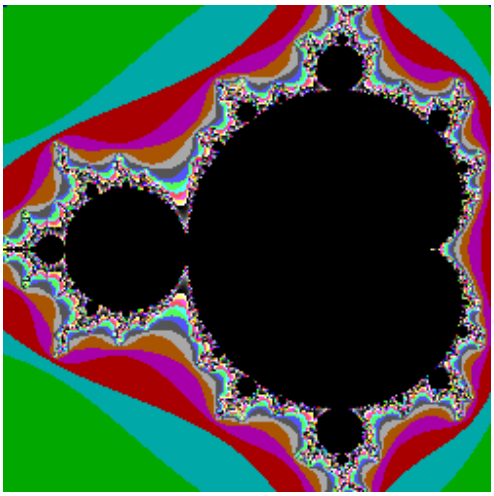
Answer: the programmer probably intended to make the no argument constructor set the string to `hello`. However, it has a return type (`void`) so it is an ordinary instance method instead of a constructor. It just happens to have the same name as the class.

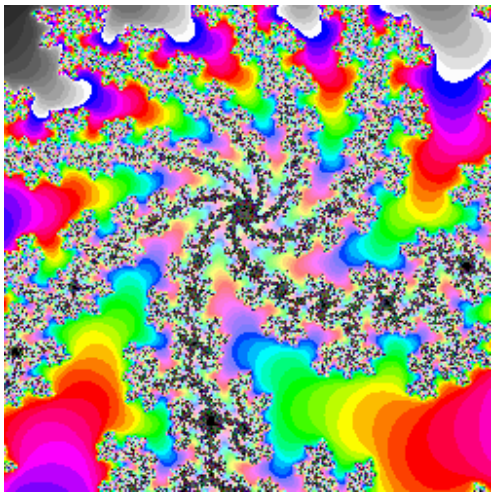


9. Write a program RootsOfUnity.java that takes a command line argument N and uses `Complex` to compute and print out the N Nth roots of unity.

```
compile:
run:
1.0
error = 0.0

6.123233995736766E-17 + 1.0i
error = 2.4492935982947064E-16

-1.0 + 1.2246467991473532E-16i
error = 4.898587196589413E-16

-1.8369701987210297E-16 - 1.0i
error = 7.347880794884119E-16

BUILD SUCCESSFUL (total time: 5 seconds)
```

10. Write a program ColorMandelbrot.java that plots a color version of the Mandelbrot set. Read in a 256-by-3 array of color values from standar input into an array, and then use the ith color if the Mandelbrot function takes i iterations. Use the data file mandel.txt as an example.





```
-1.5 -1.0 2.0 2.0
```

```
0.10259 -0.641 0.0086 0.0086
```

11. **Rational numbers.** Create a data type Rational.java and BigRational.java for positive rational numbers.

```
run:
5/6
1
28/51
17/899
0
BUILD SUCCESSFUL (total time: 2 seconds)
```

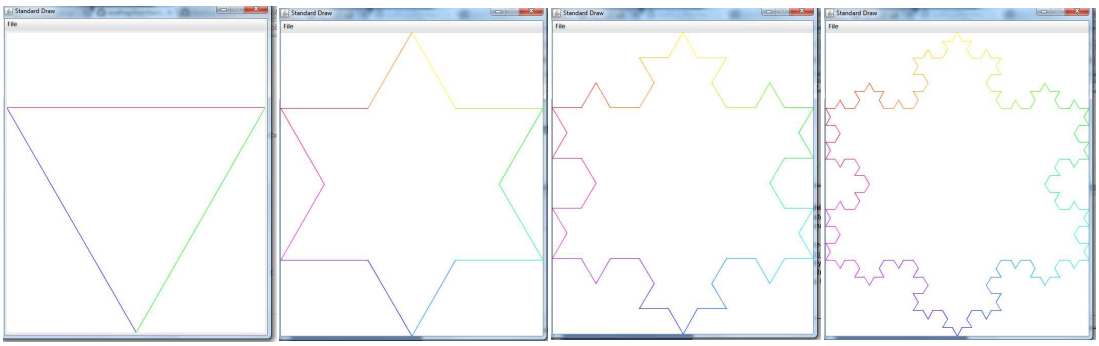
```
compile:
run:
2
5/2
8/3
65/24
BUILD SUCCESSFUL (total time: 6 seconds)
```

12. **Rational numbers.** Modify Rational.java to provide support for negative rationals and zero.

```
run:
5/6
1
28/51
17/899
0
5/6
-5/6
BUILD SUCCESSFUL (total time: 1 second)
```

13. **Koch snowflake with rainbow of colors.**

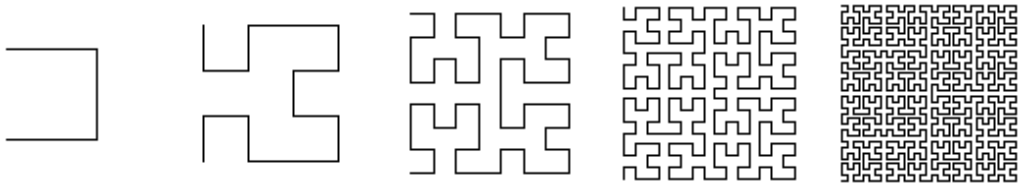
    The *Koch snowflake* of order n consists of three copies of the Koch curve of over n. We draw the three Koch curves one after the other, but rotate 120° clockwise in between. Below are the Koch snowflakes of order 0, 1, 2, and 3. Write a program KochRainbow.java that plotsthe Koch snowflake in a continuous spectrum of colors from red, to orange, yellow, green, blue, and indigo, and violet.



14. **Turtle graphics (hard).** Write a program to produce each of the following recursive patterns without lifting the pen or tracing over the same line segment more than once.

Activity 5

a. *Hilbert space-filling curve.* (Hilbert.java or SingleHilbert.java) Informally, a space-filling curve is a continuous curve in the unit square that passes through every point. In 1890, Giuseppe Peano discovered the first such space-filling curve. In 1891, David Hilbert discovered a simpler version, which came to be known as the Hilbert curve.



b. *Sierpinski arrowhead.*

c. *Sierpinski curve.*