You may discuss any of the assignments with your classmates and tutors (or anyone else) but all work for all assignments must be entirely your own. Any sharing or copying of assignments will be considered cheating.

You should not use any features of Java that have not been covered in class. If you have doubt if you are allowed to use certain structures, just ask your instructor.

**Problem 1 (10 points): Setup Your Eclipse to Run Processing**

Follow the instructions posted on the website at [http://cs.nyu.edu/~joannakl/cs101.06_f15/notes/processingInEclipse.pdf](http://cs.nyu.edu/~joannakl/cs101.06_f15/notes/processingInEclipse.pdf) to setup your Eclipse to run Processing.

Write and run the test program at the end of those instruction and take a screen-shot once the window of the program is running - submit that image for this problem. On my machine it looks something like this:

Feel free to play around with this program: you can change colors, sizes, etc. Just don’t break it!

Call your image file: **Bubbles.jpg** or **Bubbles.png**.
Problem 2 (45 points): Converting Hex, Octal and Binary to Decimal

The following program demonstrates how an integer variable can be assigned a literal that is represented in four different number systems: decimal, hexadecimal, octal and binary. The program also shows how an integer variable can be printed using different representations.

```java
public class PrintedIntegerLiterals {
    public static void main(String[] args) {
        // assign a value represented by different number notations to an integer variable
        int x = 10; // value in decimal notation
        int y = 0x17C4C; // value in hexadecimal notation
        int z = 042501; // value in octal notation
        int w = 0b11010; // value in binary notation

        // the values are printed in their decimal form by default regardless how they were assigned
        System.out.println("x = "+ x);
        System.out.println("y = "+ y);
        System.out.println("z = "+ z);
        System.out.println("w = "+ w);
        System.out.println();

        // given an int variable, printf can be used to display its value in other number systems:
        System.out.printf("Decimal: %d\n", x);
        System.out.printf("Hexadecimal: %X\n", x);
        System.out.printf("Octal: %o\n", x);
        System.out.printf("Binary: %s\n", Integer.toBinaryString(x));

        System.out.printf("Decimal: %d\n", y);
        System.out.printf("Hexadecimal: %X\n", y);
        System.out.printf("Octal: %o\n", y);
        System.out.printf("Binary: %s\n", Integer.toBinaryString(y));
    }
}
```

Output:

```
x = 10
y = 97356
z = 17729
w = 26

Decimal: 10
Hexadecimal: A
Octal: 12
Binary: 1010

Decimal: 97356
Hexadecimal: 17C4C
Octal: 276114
Binary: 101111110001001100
```

The above program demonstrates how four different numeral systems can be used in a Java program. A constant can be typeset using a base of 2 (binary), 8 (octal), 10 (decimal) or 16 (hexadecimal) - this is often referred to as a radix of a number. The integer values stored in variables can also be displayed using different format specifiers in the `printf()` method: %d for decimal, %o for octal, %X for hexadecimal. Notice, that there is no format specifier that can be used for binary representation. For that, we can convert a number to its binary string representation using `Integer.toBinaryString()`.
Your Assignment

Write a program that prompts the user for a number and then prints it using all four representations. Your program should accept a number in any representation. It needs to verify that the number contains only valid symbols for a given number representation.

Validation of Input

Here are some rules that should help you design the validation code:

- Leading digit equal to 1-9 implies that it is a decimal number. The remaining symbols should be digits 0-9. All symbols are used for the value.
- Leading zero implies that it is NOT a decimal number.
  - Second character equal to 'x' or 'X' implies that it is a hexadecimal number. The remaining symbols should be digits 0-9, and characters A-F (or a-f).
  - Second character equal to 'b' or 'B' implies that it is a binary number. The remaining symbols should be digits 0-1. The first two symbols are not used for the value.
  - Second character equal to a digit 1-7 implies it is an octal number. The remaining symbols should be digits 0-7. The first symbol is not used for the value.
  - Anything else in the second position, implies an invalid input. The program should print an error message and terminate right away.
- Anything else in the leading position, implies an invalid input. The program should print an error message and terminate right away.

(HINT: read the number in as String so you can verify it and figure out which representation it is in. You can then use int parseInt(String s, int radix) method from the String class to convert the string to the corresponding numerical value once the number has been verified and you determined its radix.)

Program Flow

Here is a detailed outline of the program:

- Display the welcome banner.

```
====================
Number conversions
====================
```

- Prompt the user for an integer.
- Validate the number and determine its value. If the entered number is invalid, print an error message and terminate the program.

```
ERROR: you did not enter a valid number.
```

- Display the number using four different representations. For example

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal:</td>
<td>16</td>
</tr>
<tr>
<td>Hexadecimal:</td>
<td>10</td>
</tr>
<tr>
<td>Octal:</td>
<td>20</td>
</tr>
<tr>
<td>Binary:</td>
<td>10000</td>
</tr>
</tbody>
</table>
```

Your program may, but does not have to use methods.

How To Work On This Program

At first, just write a program that omits the validation step. Use the sample program at the beginning of this assignment as a guide, but prompt the user for a number instead of hardcoding it into the program. Once this step is working (assuming the user always enters a valid number, otherwise your program may crash), then move on to adding the validation code. For that, you should follow the outline from the Validation of Input section above. You need to determine the radix of the number in order to use int parseInt(String s, int radix) method from the String class.
Sample Runs of the Program:

Assume user enters 25.

```
Number conversions
====================
Enter a number using decimal, hexadecimal, octal or binary:
25

Decimal: 25
Hexadecimal: 19
Octal: 31
Binary: 11001
```

Assume user enters 0xabc.

```
Number conversions
====================
Enter a number using decimal, hexadecimal, octal or binary:
0xabc

Decimal: 2748
Hexadecimal: ABC
Octal: 5274
Binary: 101010111100
```

Assume user enters 0B101.

```
Number conversions
====================
Enter a number using decimal, hexadecimal, octal or binary:
0B101

Decimal: 5
Hexadecimal: 5
Octal: 5
Binary: 101
```

Assume user enters 97356.

```
Number conversions
====================
Enter a number using decimal, hexadecimal, octal or binary:
97356

Decimal: 97356
Hexadecimal: 17C4C
Octal: 276114
Binary: 10111110001001100
```
Assume user enters 0789.

============================================
   Number conversions
============================================

Enter a number using decimal, hexadecimal, octal or binary:
0789

ERROR: you did not enter a valid number.
Problem 3 (45 points): Caesar Cipher

You are going to write a simple encryption/decryption program. The algorithm used is very easy to break, so do not start using your program to "securely store your bank account!"

Encryption is a process of encoding a message or information in such a way that only authorized parties can read it. (At least this is the idea until somebody discovers how to break an encryption pattern.)

Caesar cipher is one of the simplest encryption techniques. It was used by Julius Caesar to encrypt his private correspondence. It is sometimes called shift cipher because each letter in the unencrypted message is replaced by a different letter which is a fixed number of places down the alphabet (hence shifting).

The figure above illustrates a left shift of three, so that each occurrence of E in the plaintext becomes B in the ciphertext.

When encrypting a message, a person needs to lookup each plain letter, find its corresponding cipher letter and perform a substitution. The following example uses a right shift of three:

<table>
<thead>
<tr>
<th>Plain:</th>
<th>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</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher:</td>
<td>D E F G H I J K L M N O P Q R S T U V W X Y Z A B C</td>
</tr>
</tbody>
</table>

The message "Hello class" is encrypted by replacing each character with a character three spaces to the right in the alphabet resulting in "Khoor fodvv". The message "Ohz Brun" is decrypted by replacing each character with a character three spaces to the left in the alphabet resulting in "New York". It is easy to perform both tasks once we know the shift value and the alphabet.

Your Assignment

Write a program that performs encryption/decryption based on the Caesar cipher. Here is a detailed outline of the program:

- Display the application banner:

```
======================
Encrypt/Decrypt Tool
======================
```

- Ask the user if they want to perform encryption or decryption.

```
What would you like to do? (E)ncrypt or (D)ecrypt?
```

The program should accept both upper and lower case letters. If the user enters a different letter, the program should display an error message ERROR: invalid choice and terminate immediately.

- Ask the user for the key value. The key value is the shift amount. It should be an integer in the range -26-26. A shift of zero indicates that no encryption is performed since each letter in the plaintext is replaced with itself in the ciphertext. A positive number means shift right during encryption and left during decryption. A negative number means shift left during encryption and right during decryption.

- Ask the user for the message. You may assume that the message is contained in one line of text, but may contain spaces and punctuation characters.

- Compute the encrypted/decrypted message and print it to the screen. All alphabetic characters should be replaced by their encrypted/decrypted equivalents. Any other characters (including digits) should be left unmodified. Your program should preserve case of the letters: encrypting 'a' using the above cipher should result in 'd' and encrypting 'W' should result in 'Z'.

**Your program should use methods:** you should have an encrypt method and a decrypt method. Both methods should take the string to be encrypted/decrypted as a parameter along with an integer variable that hold the key value. Both methods should return a String.
Sample Runs of the Program:

Assume user selects encryption, shift of 5 and the message "This is a nice day!"

```
======================
Encrypt/Decrypt Tool
======================
What would you like to do? (E)ncrypt or (D)ecrypt? e
Enter your encryption key: 5
Enter your message: This is a nice day!
Your encrypted message is: Ymnx nx f snhj ifd!
```

Assume user selects decryption, shift of 5 and the message "Ymnx nx f snhj ifd!"

```
======================
Encrypt/Decrypt Tool
======================
What would you like to do? (E)ncrypt or (D)ecrypt?: D
Enter your encryption key: 5
Enter your message: Ymnx nx f snhj ifd!
Your decrypted message is: This is a nice day!
```

Assume user selects decryption, shift of 7 and the message "Ymnx nx f snhj ifd!" (Note that this is not the same decryption key that was used for encryption, so the message is displayed, but does not match the original.)

```
======================
Encrypt/Decrypt Tool
======================
What would you like to do? (E)ncrypt or (D)ecrypt?: d
Enter your encryption key: 7
Enter your message: Ymnx nx f snhj ifd!
Your decrypted message is: Rfgq gq y lgac byw!
```

Assume user selects decryption, shift of 2 and the message "Orug ri wkh Ulqjv"

```
======================
Encrypt/Decrypt Tool
======================
What would you like to do? (E)ncrypt or (D)ecrypt? D
Enter your encryption key: 3
Enter your message: Orug ri wkh Ulqjv
Your encrypted message is: Lord of the Rings
```
Assume user enters invalid choice

------------
Encrypt/Decrypt Tool
------------

What would you like to do? (E)ncrypt or (D)ecrypt? A

ERROR: invalid choice.
Grading

**Does the program compile?** If not, you will lose all the points for that problem.

**Is the program properly documented?** (worth approximately 20% of each problem)

Proper documentation at this point in the course includes:

- preamble with the name of the author, date of creation and brief description of the program (the description should specify what the program does, not that it is a solution to problem 1 of homework 1);
- appropriately chosen variable names, i.e., descriptive names (a good name for the variable that stores the bonus amount in the last problem is `bonus`, not `x`);
- comments inside the code describing steps needed to be taken to accomplish the goal of the program;
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

**Is the program well developed?** (worth approximately 40% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.

**Is the program correct?** (worth approximately 40% of each problem) Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

**What and how to submit?**

You should submit three source code files combined into a single zip file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have `.java` extensions. Name your classes as specified in the problems.

If you wish to use your (one and only) freebie for this project (one week extension, no questions asked), then complete the form at [http://goo.gl/forms/fpUJrF64b5](http://goo.gl/forms/fpUJrF64b5) **before the due date for the assignment**. All freebies are due seven days after the original due date and should be submitted to NYU Classes.

**Questions**

Post any questions you have regarding this assignment to Piazza under the "homeworks" topic.