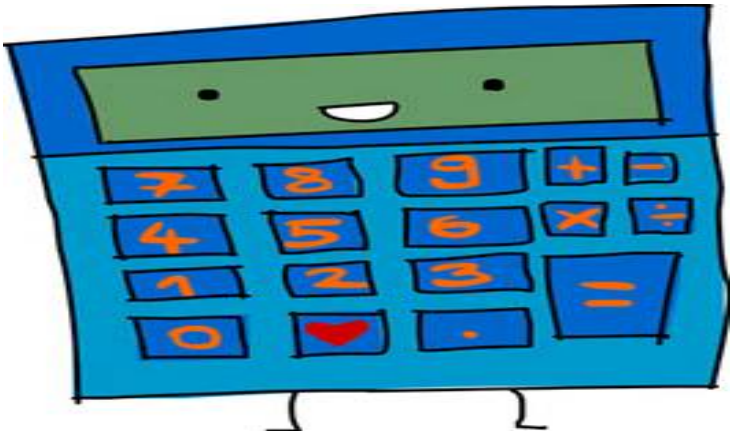

Due Date:

1. Soft copy : October 22th by midnight
2. Hard copy: October 22 / 23(depending on your lecture day)

What to turn in:

1. You must email a soft copy to: csus.csc15@gmail.com. The subject of the email must indicate the following. An email without the proper subject will not be graded.
    a. Last name, first name
    b. HW #
    c. Section number
    d. Lecture time
2. A hard copy must be turned in at the beginning of the lecture on the assigned due date. The hard copy must include the output of your program.

Grading policies: you will be graded based on

1. How well your program runs
2. Proper variable names, meaningful names
3. Proper indentation
4. Comments, you must provide a block comments for your program, explaining what the program does, also you must provide comments before each method explaining what the method does.
5. Implementing all the methods properly, you cannot add any more methods to your program.
6. Your program must work with any data
7. Test your program with the expressions provided at the bottom

HW #1: Problem: Make a simple English calculator that does the following:

1. Takes one string from the keyboard, containing two single digits (0 to 9), a char representing one of the five operations and one ";" . The operations are : +(addition), subtraction) -, /(division), *(multiplication), ^(exponentiation)

2. The user will enter all the information on one line, you need to use the nextLine() method to read the input. Once you read the input, you need to use the methods form the String class to separate the tokens to two operands one operation as String data type. Awe need to convert the String operands to integer value. You can use the available **Integer** class in java to convert a string to its equivalent integer value. For example **Integer.paresInt("3")** will return the integer value 3.
3. Once you have the two operands and the operation, Output the description of the operation in plain English, as well as the numeric result.

For instance, if the input is **5 * 3;** then the output should be*: five multiply by three is 15.*

Note that the result is given as a number, not a word.

If the input is **2 – 9;** then the output should be: *two minus nine is -7*

If the two numbers are 5 and 0, and the operation is /, then the output should be: *division by zero is not allowed.*

If the two numbers are 25 and 3, and the operation is +, then the output should be*: invalid number*

As for the operators, they should be translated into English as follows:

+	plus
*	Multiply by
/	Divided by
-	minus
^	To the power of
%	Modulus

You must use switch statements at least once throughout your code.

You need to consider all the special cases such as:

1. You must check the format of the expression first. If it is in the valid format then you start separating the tokens. A valid format is: **number operator number;** (tokens are separated with a space and expression ends with a semicolon. If the format is invalid, output a message indicating **invalid format.**
2. For division, there is a special constraint: you cannot divide by 0, and you should therefore test whether the second number is 0. If it is 0, then you should output a message saying that you are not allowed to divide by zero.
3. The operator is not one of the preceding five operators; in that case output a message saying that the operator is not valid.
4. One of the two numbers is not a valid digit; again you should output a message to that effect.
5. Your code must run 5 times, using " for " loop that repeats five times. For one run of your code, the user should be able to enter 5 different expressions.

Programming tips:

Always do incremental programming, implement one method at a time, compile it, and test it. Then start the next method. Avoid implementing all the methods at one since you might be overwhelmed with the amount of syntax errors

Here is the list of the methods that you must create. To get the full credit you must write all the methods that are listed below. You are not allowed to make any changes to the list of the parameters and the return types:

1. **Public static void main(String[] args):** This method which is the main method gets the user's input and calls different method to calculate the expression.
2. **Public static void validCharacter(Strin operand):** This method gets one String as its parameter, and return s true if the first letter in the string is one of the letters '0', '1',…'9'. Otherwise it should return false. This method is important since we are trying to convert a string into an integer.
3. **Public static string token (String s):** This method receives one String including spaces, and return the first token in the string. For example the call token ("5 + 3;") should return 5 and it should remove the token from the String s so that the next call to the method token would be token ("+ 3;") should return + and the call token ("3;") should return 3;. This method will give you all the needed tokens. Pay attention that the last token has a ';' at the end and you must remove it before processing it.
4. **Public static String removeToken(String s):** This method removes the first token in the given string. For example removeToken("3 + 6;") should return the String "+ 6;"
5. **Public static boolen validFormat(String expression):** This method gets a parameter of type String representing the expression and returns true if the expression has the correct format. A valid format contains 3 tokens. Therefore you need to count the number of the tokens in the string. A valid expression cannot stars with a space, operands and operators must be separated with exactly one space, expression must end with a ;.
6. **Public static boolean validNumber (int operand):** This method receives an integer value and returns true if the integer number is between 0 and 9.
7. **Public static String stringToNum(int num) :** This method receives an integer value between 0 and 9 and then returns the equivalent string for the given number. For example the call stringToNum(5) should return **five**.
8. **Public static String operatorDescription(char operator):** This method receives a character as its parameter representing the operation and returns the equivalent description for that operation. For example operatorDescription('+') should return plus. Refer to the provided table for the return value corresponding to each operation.
9. **Public static void description ():** This method outputs the description of your program. Telling the user what this program does, what is a valid format….
10. **Public static int calculate (int op1, int op2, char operation):** This method gets three parameters, representing the two operands and the operation and returns the result of the arithmetic operation.

11. **Public static void output (String op1, String op2, string operation, int result):** This method receives two String representing the operands and one string representing the operation and outputs the result. The output of your code must match the following output

Here is a sample output. You must run your code with the expression provided below.

```
Welcome to my calculator website
**************************************************
    your operands must be between 0-9
    The operations can only be +, -, *, /, ^, %
    Each token must be separated with exactly one space
    No space at the beginning of your expression
    You must end your expression with a ;

    ******************************************************
    Enter your expression:   4+  *;
    please check your expression.
    ******************************************************
    your operands must be between 0-9
    The operations can only be +, -, *, /, ^, %
    Each token must be separated with exactly one space
    No space at the beginning of your expression
    You must end your expression with a ;

    ******************************************************
    Enter your expression:     4 + 6
    please check your expression.
    ******************************************************
    your operands must be between 0-9
    The operations can only be +, -, *, /, ^, %
    Each token must be separated with exactly one space
    No space at the beginning of your expression
    You must end your expression with a ;

    ******************************************************
    Enter your expression:4 + 7;
    Four Plus Seven = 11
    Enter your expression:3 / 0;
```

```
Three Divide By Zero = UNDEFINED
Enter your expression:   9 + 6   ;
please check your expression.
**************************************************
your operands must be between 0-9
The operations can only be +, -, *, /, ^, %
Each token must be separated with exactly one space
No space at the beginning of your expression
You must end your expression with a ;

**************************************************
Enter your expression:9 + 6;
Nine Plus Six = 15
Enter your expression:567 * 5;
please check your expression
**************************************************
your operands must be between 0-9
The operations can only be +, -, *, /, ^, %
Each token must be separated with exactly one space
No space at the beginning of your expression
You must end your expression with a ;

**************************************************
Enter your expression:5 ^ 2;
Five To The Power Of Two = 25
Enter your expression:5 % 3;
Five modulus Three = 2
Enter your expression:4 ^ 4;
Four To The Power Of Four = 256
Enter your expression: a + b;
please check your expression
**************************************************
your operands must be between 0-9
The operations can only be +, -, *, /, ^, %
Each token must be separated with exactly one space
No space(s) at the beginning of your expression
You must end your expression with a ;
```