CS-200-1: Programming I, Fall 2014
Northeastern Illinois University
Homework #6
Due: Wednesday, November 5th by 2:50 p.m.

Assignment Specifications: Read all instructions carefully!
Make sure the following are in a .zip file - Do **NOT** submit files individually to D2L!!

- ☐ Your source code (the .java files).
- ☐ Your output in .txt file(s).
- ☐ Make sure your name and assignment number are in the .txt file and the .java file(s) (as comments for the .java files).
- ☐ Turn your homework in to D2L before class (no late homework will be accepted - see syllabus for policies).

Problem #1

- Trace through (i.e. look at one line at a time) the .java program written out below.
- Determine what is printed out to the console. In a .txt file (named Homework6_P1.txt), put the output (i.e. what is printed out to the console) exactly as it would appear in the console.
- You may use jGrasp to help you as you trace through this, although this should be a way to check your tracing.
- Put the Homework6_P1.txt file into a folder named Homework6.

```
public class HW6Tracing
{
    public static void main(String[] args)
    {
        int a = 5, b = 2, c = -4, d = 10;
        System.out.println(addNegatives(a, c));
        System.out.println(multiplyOpposites(c, b));
    }

    public static int negative(int num)
    {
        return -1 * num;
    }

    public static int addNegatives(int num1, int num2)
    {
        int sum = negative(num1) + negative(num2);
        return sum;
    }

    public static int multiplyOpposites(int num1, int num2)
    {
        int product = negative(num1) * num2;
        return product;
    }
}
```

Problem #2

- Create a new .java file named SortNumbers.java.
- Write a method that has the following header:
  `public static void displaySortedNumbers(int a, int b, int c)`
- The method should display three numbers in increasing order.
- In the main method, ask the user to enter three numbers. Use/invoke the displaySortedNumbers method to display the entered numbers in order.
- Your output should match the sample output below.
- Put the SortNumbers.java file into the Homework6 folder.

```
Enter a number: 10
Enter a number: 3
Enter a number: 17
3 10 17
```

```
Enter a number: 4
Enter a number: -6
Enter a number: -8
-8 -6 4
```

Problem #3

- Open the .java file named Operations.java from the Needed Files folder provided for you.
- Write two methods named multiply and divide. The methods should have the following headers:
  `public static int multiply(int a, int b)`

  `public static int divide(int a, int b)`
- The multiply method should multiply the parameter a by the parameter b. However, you may not use the multiplication operator (*). You may only use the existing methods of add or subtract.
- The divide method should use integer division to divide the parameter a by the parameter b. However, you may not use the multiplication operator (*), the division operator (/) or the mod (%) operator.
- After you complete the multiply and divide methods, uncomment the commented lines in the main method.
- If you wrote the methods correctly, your output should match the sample output below.
- Hint for the multiply method: A for-loop may come in handy.
- Hint for the divide method: A while-loop may come in handy.

- Put the Operations.java file into the Homework6 folder.

```
34 * 67 is 2278
3729 * 22 is 169
```

Problem #4

- Create a .java file named Palindromes.java.
- A word that is a palindrome has the same characters when reading from beginning to end as it does reading from the end to the beginning.
- Write two methods that have the following method headers:
  `public static String reverse(String word)`

  `public static boolean isPalindrome(String word)`
- The reverse method should return the reverse of the String parameter.
- The isPalindrome method should use the reverse method to determine whether a word is a palindrome and return true if it is a palindrome and false otherwise.
- In the main method, ask the user to enter a word.
- Use the palindrome method to determine whether the entered word is a palindrome.
- Your output should match the sample output below and your program should contain a total of three methods: `main`, `reverse`, `isPalindrome`
- Put the Palindromes.java file into the Homework6 folder.

```
Enter a word: bananas
bananas is not a palindrome
```

```
Enter a word: redivider
redivider is a palindrome
```