This problem set is due **Thursday, October 13 at 9:00pm.** Note that the due date that you see on Blackboard is not accurate since it shows the time in EST. You should go by the due date in this handout.

- To solve each problem below, you will be implementing a class.
- Please carefully read and follow the directions exactly for each problem. Files and classes should be named exactly as directed in the problem (including capitalization!) as this will help with grading.
- You should create your programs using your preferred text-editor or the Eclipse text editor.
- Your programs should be formatted in a way that's readable. In other words, indent appropriately, use informative names for variables, etc. If you are uncertain about what is a readable style, see the examples from lectures and textbook as a starting point for a reasonable coding style.
- Your programs should compile and run without errors. Please do not submit programs that do *not* compile! Its better to submit partial implementation that compiles as opposed to complete implementation that does not compile. You may use either Eclipse or the command-line to compile and run your programs.
- Be sure to *include your name and your email address as comments at the top of each file that you submit.*

**What Java Features to Use**

For this assignment, you are not allowed to use more advanced features than what we have covered in Lecture 1 through Lecture 9 (Chapters 1 through 6 of our textbook).

**What to Submit**

Submit the following file(s) on Blackboard. Please do **not** submit .class files or Eclipse-specific project files or any I did not ask for.

- Sphere.java
- Sum.java
- Prime2.java
- Palindrome.java
- Password.java
Naming Conventions In Java And Programming Style In General

Refer to the ones given in PS 2.

Problem 1

Read the FAQ next to this handout for hints and corrections before you try each of the problems below.

Write a program (Sphere) that includes the following methods in addition to a main.

- a method called sphereVolume that accepts a radius parameter and returns the volume of a sphere with that radius. For example:
  - sphereVolume(2.0) should return 33.510321638291124 or a number close to it.
  - sphereVolume(5.0) should return 523.5987755982989 or a number close to it.

- a method called cylinderSurfaceArea that accepts a radius and height as arguments and returns the surface area of a cylinder with those dimensions. For example:
  - cylinderSurfaceArea(3.0, 4.5) should return 141.3716694115407 or a number close to it.
  - cylinderSurfaceArea(4.5, 6.0) should return 296.8805057642354 or a number close to it.

For PI, use the Java-supplied static constant: Math.PI and your program should read the radius, height values from the user. Here are four sample runs:

Enter a sphere radius: 2.0
Sphere volume: 33.510321638291124

Enter a sphere radius: 5.0
Sphere volume: 523.5987755982989

Enter a cylinder radius and height: 3.0 4.5
Cylinder surface area: 141.3716694115407

Enter a cylinder radius and height: 4.5 6.0
Cylinder surface area: 296.8805057642354

Hand in Sphere.java.

Problem 2

Write a program (Sum.java) with a main method that prompts the user to enter an integer and sums the digits in the integer. You must define a method of the following header and use it:

    public static int sumDigits(int n)

Here are three sample runs:

Enter an integer: 0
Sum of digits of 0 is 0
Enter an integer: 9
Sum of digits of 9 is 9

Enter an integer: 34521
Sum of digits of 34521 is 15

Hand in Sum.java.

Problem 3
Write a program (Prime2.java) with a main method that prompts the user to read an integer and prints all twin primes that are less than the integer. Twin primes are a pair of prime numbers that differ by 2. For example, 3 and 5 are twin primes. So are 11 and 13. Here are two sample runs:

Enter an integer: 13
(3, 5)
(5, 7)

Enter an integer: 14
(3, 5)
(5, 7)
(11, 13)

You must define two methods of the following headers and use them:

    public static boolean isPrime(int n)
    public static boolean isDivisibleBy(int i, int j)

Hand in Prime2.java.

Problem 4
Write a program (Palindrome.java) with a main method that prompts the user to read a string and determines if it is a palindrome. You must define two methods of the following headers and use them:

    public static String reverse(String s)
    public static boolean isPalindrome(String s)

Here are three sample runs:

Enter a word: Apple
Apple is not a palindrome

Enter a word: Appa
Appa is a palindrome

Enter a word: alula
alula is a palindrome

Hand in Palindrome.java.
Problem 5

Write a program (Password.java) with a main method that prompts the user to read a password and determines if the password read in is a valid password or not. A password is valid if it passes all of the following rules:

- A password must have at least eight characters.
- A password must consist of only letters and digits.
- A password must contain at least two digits.

You must define three methods of the following headers and use them:

```java
public static boolean isValidPassword(String s)
public static boolean isLettersAndDigits(String s)
public static int countDigits(String s)
```

Here are three sample runs:

Enter a password: AppleNot
AppleNot is not valid

Enter a password: Apple2Not3
Apple2Not3 is valid

Enter a password: Apple2NotYet
Apple2NotYet is not valid

Hand in Password.java.