This problem set is due on Thursday, October 6 at 11:59pm.

- 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 8 (Chapters 1 through 5 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.

Grade.java
Decoder.java
Digits.java
Prime.java
Lotto.java
Stat.java
Diamond.java
Naming Conventions In Java And Programming Style In General
Refer to the ones given in PS 2.

Problem 1
Write a program (Grade.java) with a main method that prompts the user to enter a letter grade, A, B, C, D, or F and displays its corresponding numeric value 4, 3, 2, 1, or 0 respectively. Any other character is invalid and should be identified as such. Below are examples of running the program four times:

Enter a letter grade: B
The numeric value for grade B is 3

Enter a letter grade: F
The numeric value for grade F is 0

Enter a letter grade: M
M is an invalid input

Enter a letter grade: $
$ is an invalid input

Hand in Grade.java.

Problem 2
Write a program (Decoder.java) with a main method that prompts the user to enter three characters and decodes the three characters as follows:

- **First character**: may be S for student, F for faculty, or T for sTaff.
- **Second character**: may be F for female or M for male.
- **Third character**: may be N for living oN campus or F living oFf campus.

For each character you should check for invalid character. Input that has fewer than three characters or more than three characters should be considered as invalid input as well. Below are examples of running the program six times:

Enter three characters: SFN
Female student living on campus

Enter three characters: FMF
Male faculty living off campus

Enter three characters: KMF
KMF is an invalid input

Enter three characters: FMP
FMP is an invalid input

Enter three characters: SFNT
SFNT is an invalid input
Enter three characters: SF
SF is an invalid input

Hand in Decoder.java.

Problem 3
Write a program (Digits.java) with a main method that prompts the user to read a positive integer smaller than 1,000,000 and prints each digit of the integer separated by a blank space backward. Below are examples of running the program three times:

Enter an integer: 532190
0 9 1 2 3 5

Enter an integer: 1453095
Too large

Enter an integer: -45
Too small

Hand in Digits.java.

Problem 4
Write a program (Prime.java) with a main method that prompts the user to read a number and determines if the number is a prime number or not. If a prime number has been read in, your program will indicate as such and terminate. If the number read in is not prime, then it should repeatedly ask for a new number until it reads a prime number. Here are two sample runs:

Enter an integer: 7
7 is prime

Enter an integer: 8
8 is not prime
Enter an integer: 27
27 is not prime
Enter an integer: 13
13 is prime

Hand in Prime.java.

Problem 5
Write a program (Lotto.java) with a main method that does the following:

- Set up two random integers that are different from each other. (Hint: Generate a number. Use a loop to continuously generate the second number until it is different from the first number.) These are the lotto numbers.
- Prompt the user for two numbers and read them in. These are the player’s numbers.
- If the two player’s numbers match both of the lotto numbers, award $50 by printing it out appropriately.
• If only one matches, award $10.
• If none matches, output "Sorry, no match".

Hand in Lotto.java.

Problem 6
Write a program (Stat.java) with a main method that prompts the user to input any number of integers. The last number has to be a zero. Then, it does the following:

• Prints the largest number in the input.
• Prints the smallest number in the input.
• Prints the count of the numbers in the input excluding the zero at the end.
• Prints the occurrence of the largest number.
• Prints the average of the numbers in the input.

Here are two sample runs:

Enter integers (terminated with a 0): 0
No useful numbers entered

Enter integers (terminated with a 0): 3 5 2 5 5 2 1 3 5 0
Max: 5
Min: 1
Count all: 9
Count max: 4
Average: 3.44

Note that the average is printed with only two digits below the decimal point. Hand in Stat.java.

Problem 7
Write a program (Diamond.java) with a main method that randomly generates an integer between 1 and 9 inclusive and displays a diamond. Below are two sample runs:

Random number: 3
  1
  2 1 2
3 2 1 2 3
  2 1 2
    1

Random number: 7
  1
  2 1 2
3 2 1 2 3
4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
6 5 4 3 2 1 2 3 4 5 6
Hand in Diamond.java.