CS 1400-03 Introduction to Programming and Problem Solving
Coding Practice #4
(Due: 11:59 PM, Friday, 2/26/2021)

Except Coding Practice #1, I will not grade your coding practice submissions. Instead, they will be treated as participation points. On blackboard, you will receive full points as long as you work on the exercises, which don't necessary mean they are all correct. Please check your own programs carefully and make sure they do generate the desired output.

Objectives:

- Be able to create complete Java programs with
  - Loops
  - Input Validation
  - Nested Loops
  - File Input and Output
  - Decision Structures
- Be able to test and debug a program

Change your working directory to `cs1400/codingPractice` for this assignment.

Task #1 Exam

(1) Rewrite the following program and use a `for`-loop to replace the `while`-loop.
(2) Add statements to verify whether the user input is either 0 or 1. If not, display an appropriate error message and ask the user to enter again until ten correct input values are processed.

```
//analysis of examination results
import java.util.Scanner;
public class Exam
{
   public static void main(String[] args)
   {
      Scanner keyboard = new Scanner(System.in);
      int passes=0, failures=0, students=0, result;
      while (students<10)
      {
         System.out.print("enter result (1=pass, 0=fail): ");
         result = keyboard.nextInt();
         //add your code here for input validation
         if (result==1)
            passes++;
         else
            failures++;
         students++;
      }
      System.out.println(passes + " passed\n"
                         + failures + " failed");
      if (passes < 5)
         System.out.println("Raise tuition");
   }
}
```

Task #2 Pattern

Write a complete Java program, called `Pattern.java`, using nested-`for` loops to produce the following pattern:

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Task #3 Hollow Square

Write a Java program, called `HollowSquare.java`, that will ask the user to provide an integer, say `n`, and use nested-`for` loops to print a hollow square of size `n` with asterisks. Use a `do-while` loop to verify that `n` is in the range of 1 and 20.

For instance,
When `n=1`, print

```
*
```

When `n=2`, print

```
**
**
```

Sample interactions:

```
enter an integer (1-20): 0
enter an integer (1-20): -1
enter an integer (1-20): 21
enter an integer (1-20): 100
enter an integer (1-20): 10
**********
*        *
*        *
*        *
*        *
*        *
*        *
*        *
*        *
**********
```

Task #4 File I/O

Write a Java program, called `SeparateEvenOddNumbers.java`, that reads the name of an input file containing integers, and creates two new files, one containing the even integers and one containing the odd integers which were in the input file. Use the following pattern for filenames: if the input file is **`data`**, then the two output files should be **`data.even`** and **`data.odd`**. If the input file does not exist, give an appropriate error message and terminate the program. The following are examples of the required I/O behavior, where the user's input is shown in bold.

```
fcsang@garrison ~/cs1400/codingPractice $ java SeparateEvenOddNumbers
Enter an input filename: data
Data written to data.even and data.odd
```

For example, if the input file data is

```
2
7
-7
8
2
8
```

then the program produces `data.even` containing even integers

```
2
8
2
8
```

and `data.odd` containing odd integers

```
7
-7
```

Submission:

Generate a `script` file `practice4.txt` with appropriate time stamps and the following steps visible:

1) a `pwd` to show the current working directory
2) a `ls -l` to show in long format the files in your `cs1400/codingPractice`
3) a `cat` to display `Exam.java`
4) compile `Exam.java`
5) run `Exam`
6) a `cat` to display `Pattern.java`
7) compile `Pattern.java`
8) run `Pattern`
9) a `cat` to display `HollowSquare.java`
10) compile `HollowSquare.java`
11) run `HollowSquare`
12) a `cat` to display `SeparateEvenOddNumbers.java`
13) compile `SeparateEvenOddNumbers.java`
14) run `SeparateEvenOddNumbers`

Submit the script file `practice4.txt` on Bb, under the Coding Practice Folder, Practice #4 assignment.