What to turn in:
Submit your programs on Blackboard. As always, include some comments at the beginning of each program file including your name, the course number, and a brief description of what the code does.

CODING CONVENTIONS (IMPORTANT):
For this homework, and each subsequent homework, you are required to follow the coding conventions listed below. Failure to follow these conventions in full may result in a loss of up to 20% of your grade on the assignment, even if your code works perfectly.

Required comments:
Each file should have comments at the beginning that state your name and a brief description of what the program does. Of course, you are always free to provide additional comments that you feel make the program code easier to understand.

Indentation and spacing:
As stated in Section 1.10.2: "Indent each subcomponent or statement at least two spaces more than the construct within which it is nested." Also, make sure that statements that are nested at the same level have their left-most characters line up evenly.

Good:

```
if (avg >= 92.0)
{
   grade = "A";
   System.out.println("Great job!");
}
```

Bad:

```
if (avg >= 92.0)
{
grade = "A";
   System.out.println("Great job!");
}
```

Bad:

```
if (avg >= 92.0)
{
   grade = "A";
      System.out.println("Great job!");
}
```

Furthermore, "a single space should be added on both sides of a binary operator".

Good:

```
int i = 3 + 4 * 4;
```

Bad:

```
int i= 3+4 *4;
```

Block styles:
Review Section 1.10.3. You may use either the "Next-line style" or the "End-of-line style" of creating code blocks. Each has its advantages and disadvantages. In a given file, you should use one style or the other: do not mix block styles within a single source code file.

Next-line style:

```
public class Test
{
   public static void main(String[] args)
   {
      System.out.println("Block Styles");
   }
}
```

End-of-line style:

```
public class Test {
   public static void main(String[] args) {
      System.out.println("Block Styles");
   }
}
```

Naming conventions:
As specified in Section 2.8 of the textbook, you should:

- Begin the name of each variable and method with a lower case letter. "If a name consists of several words, concatenate them into one, making the first word lowercase and capitalizing the first letter of each subsequent word – for example, the variables **radius** and **area** and the method **showInputDialog**."
- "Capitalize the first letter of each word in a class name – for example, the class names **WindChill**, **Math**, and **JOptionPane**."
- "Capitalize every letter in a constant, and use underscores between words – for example, the constants **PI** and **MAX_VALUE**."

Tip: Always start the name of a .java source file with a capital letter. E.g., **WindChill.java**, *NOT* **windChill.java**. This will force you to start the name of the public class therein with a capital letter.