Methods

Examine the following starter code provided by the instructor. **promptForInteger()** and **promptForDouble()** are methods, or functions, that modularize or compartmentalize specific code blocks. Specifically, the methods below allow the developer to prompt the user for a value of a specific type and then return the value entered by the user (to then store in a variable). Now, when the developer needs to prompt a user for a value, they can do so with one line of code that both prompts and returns a value. In a later lab you will see a way to organize these methods into a reusable utility that you can import into any project. For now, you will need to create a prompt method for several other Java data types.

```java
public class Lab12Starter {
    public static void main(String[] args) {
        int intVal = promptForInteger("Enter an integer: ");
        double doubleVal = promptForDouble("Enter a double: ");

        System.out.println("intVal = " + intVal);
        System.out.println("doubleVal = " + doubleVal);
    }

    public static int promptForInteger(String prompt) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(prompt);

        return scanner.nextInt();
    }

    public static double promptForDouble(String prompt) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(prompt);

        return scanner.nextDouble();
    }
}
```

For this lab you will create various prompt methods in your main class. for the following additional data types; short, long, float, char, String. You should create a variable for each type to store the return value of your prompt method. Then print the values in each of variables.

Notes:

1. Notice the similarities and differences in the existing methods. You will emulate these patterns with your additions.
2. The prompts will always accept a String as a parameter. This String gets displayed to the user before accepting input.
3. The return type for each method must be changed for each method you create. When a method returns, or ends, it can pass a value back to the code that called or invoked it. This lets you store the return value, or result, of a method in a variable. Not all methods return values; sometimes a method might just complete a task and exit, this depends on your program design. All the methods in this lab will have a return type and return the value entered by the user.
4. Make sure to call the appropriate Scanner method for the data type you are returning in your method. The Scanner method returns the value entered by the user, which is the value that you want to return from your prompt method. You could save the result in a variable and return that variable, if you had a reason to, but as you will see in the example code, the method is simply invoked in the return statement.

Example Output

```
Enter an integer: 42
Enter a double: 648486.6502
Enter a short: 12
Enter a long: 86523526421
Enter a float: 21.3626
Enter a character: Q
Enter a String: Hello World
intVal = 42
doubleVal = 648486.6502
shortVal = 12
longVal = 86523526421
floatVal = 21.3626
charVal = Q
stringVal = Hello World
```

Deliverables

Make sure your code has the required file header and correctly formatted identifier names, as outlined in the CS Java Documentation Policy under Course Info on D2L.

To receive credit for this lab you must

1. Demonstrate the code and execution to the instructor during this lab, during office hours, or during the next lab period.
2. Zip the src folder in your project directory and upload the instructor approved .java files to the Lab 12 D2L drop box.