Programming Fundamentals COIT 11222

Assessment item 1—Java Console Program

Due date:	**Week 6 T119** – Midnight, **Friday** 26 April 2019 Refer below for complete assessment item 1 requirements (**Assignment One**)	**ASSESSMENT**
Weighting:	20%	
Length:	N/A	**1**

Objectives

This assessment item relates to the unit learning outcomes as in the Unit Profile.

Details

For this assignment, you are required to develop **Java Console Programs** to demonstrate you can use Java constructs including input/output via a command line and using GUI dialogs, Java primitive and built-in data types, Java defined objects, selection and looping statements, methods, and various other Java commands. Your program must produce the correct results.
You are only allowed to use techniques which have been covered in the first five weeks of the subject and within the assignment literature, you must use a **Scanner** object for console input and no advanced data structures like arrays will be used.

What to submit for this assignment

The Java source code:

You will be able to complete the assignment in weekly parts in which you will produce five java source files. (More details below)

Week1.java, Week2.java, Week3.java, Week4.java and Week5.java.

Once you have completed all of the programs and you are ready to submit, compress all source files into a single zip file for submission, **do not include your report in the zip file**. Only submit a zip not a rar file. It is important the file names are correct.

- **Ass1.zip**

Also submit a report including, how long it took to create the programs (approximately), any problems encountered and screen shots of the output produced. (Use Alt-PrtScrn to capture just the console window or your dialogs and you can paste them into your Word document) You should test every possibility in the program and annotate your test screen shots.

- **ReportAss1.docx**

You will submit your files by the due date using the "**Assignment 1 Submission**" link on the Moodle unit website in the **Assessment Block** or in the relevant week.

Assignment specification

This assignment will require you to write small five programs, do not panic! They will be small programs which will cover the first five weekly topics. Usually students were required to write one largish program to demonstrate the topics for the first five weeks. Students get themselves into trouble when the first assignment is due as they have not practiced the basics skills necessary to complete the assignment. With the assignment divided into five programs you can complete each question as we cover the weekly topics, do not let yourself fall behind.

General Instructions

Each program must contain a header comment which contains: Your name and student number, the name of the file, the date and a brief description of the purpose of the program:

```
// Programmer: Eric Gen S01234567
// File: Week1.java
// Date: April 26 2019
// Purpose: COIT11222 assignment one question one T119
// Use println method to print initials using asterisks
```

All programs will be aligned and indented correctly, and contains relevant comments for declarations and statements. All variables and objects will be declared with a meaningful name and use lowercase camel notation:

```
String customerName;
```

All coding will be contained within a main method except for question five when a method will be created and used.

For this assignment you will not worry about checking numeric ranges or data types.

Refer to a Java reference textbook and the unit material (available on the unit WEB site) for further information about the Java programming topics required to complete this assignment.

Check the marking guide (last page) to ensure you have completed every task. You need to match all output exactly as the sample screenshots shown below.

Distance and Melbourne students can email questions directly to me, other campus students should seek help from your local tutor, and you can still contact me if it is urgent, I usually respond to emails very promptly.

Good luck --- Bruce McKenzie COIT11222 unit coordinator term 1 2019

b.mckenzie@cqu.edu.au

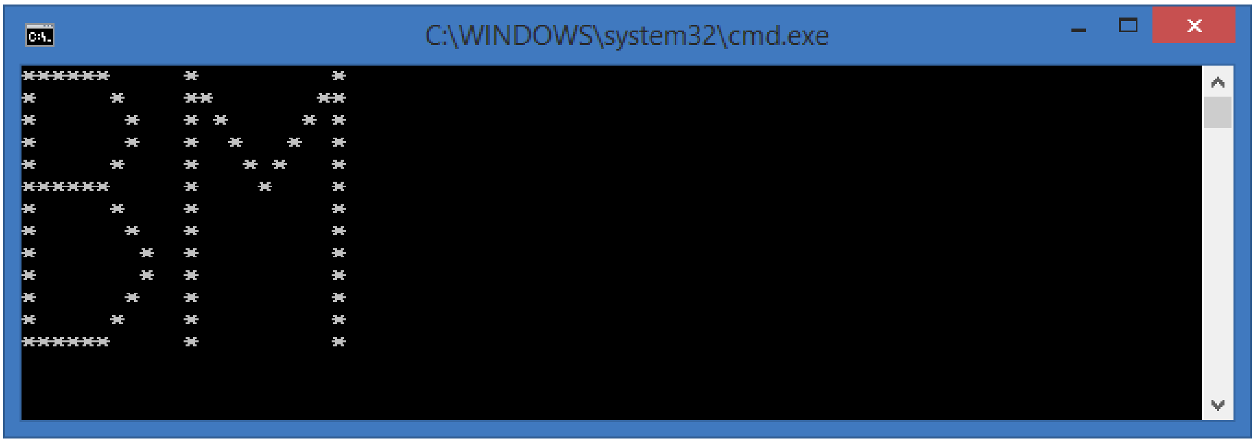
Question one (week one topic). Writing output to the screen.

Once you have written your first “Hello World” program you will be able to complete question one.

Implementation

Create a class called **Week1** (file: Week1.java) and within it a main method.

Use the command `System.out.println("");` to print out the first initial of your first and last names as a matrix of asterisks. For example this is my first and last initials printed.



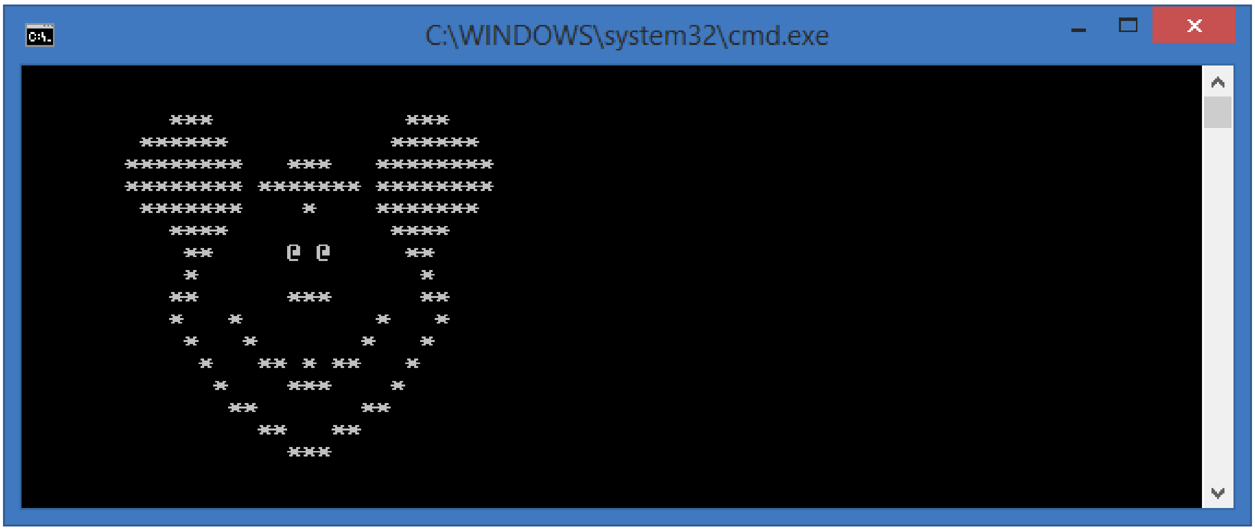
```
******     *         *
*     *    **       **
*      *   * *     * *
*      *   *  *   *  *
*     *    *   * *   *
******     *    *    *
*     *    *         *
*      *   *         *
*       *  *         *
*       *  *         *
*      *   *         *
*     *    *         *
******     *         *
```

The first line of asterisks is printed with this command:

```
System.out.println("******     *         *");
```

You may need to use some graph paper to plot where you need to print your asterisks.

If you like you could submit a picture. An attempt at Mickey Mouse! Just do your initials as it takes a while to create a picture.



```

          ***             ***
        ******           ******
       ********   ***   ********
       ******** ******* ********
        *******    *    *******
          ****           ****
           **     @ @     **
           *               *
          **      ***      **
          *   *         *   *
           *   *       *   *
            *   ** * **   *
             *    ***    *
              **       **
                **   **
                  ***
```

Question two (week 2 topics) Input of data types and arithmetic expressions

Rocky Dry Cleaners program

Rocky Dry Cleaners are requesting a program which allows staff to input a customer's name and the number of plain garments to be dry cleaned. For simplicity we are only considering plain garments which include shirts, trousers, dresses and jackets etc. The program will compute the cost of the order at **$8.50 per garment**, you should store this value as a constant.

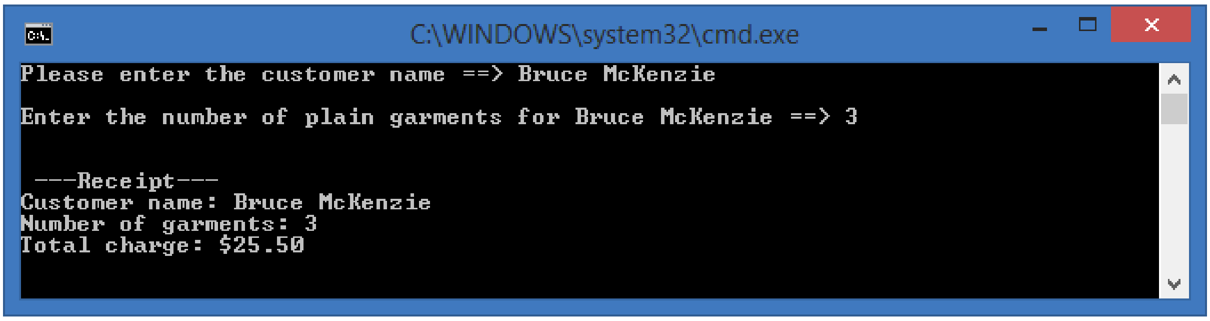
This program will prompt for and read in a customer name using a Scanner object.

The customer name will be stored in a String object.

The program will then output the customer name in a prompt to read in the number of garments to be cleaned (as a whole number i.e. an integer).

Finally the program will display the receipt for the customer.

You need to replicate the output **exactly** as shown below, including the correct line spacing.



```
Please enter the customer name ==> Bruce McKenzie

Enter the number of plain garments for Bruce McKenzie ==> 3


 ---Receipt---
Customer name: Bruce McKenzie
Number of garments: 3
Total charge: $25.50
```

Implementation

Create a class called **Week2** (file:Week2.java) and within it a main method as per question one.

Import the Scanner class i.e.

```
import java.util.Scanner;
```

Within your main create two Scanner objects named **inText** and **inNumber**. One for reading text and the other for reading the numbers, it does not really matter here to have separate Scanner objects but there will be problems later when reading a series of text and numbers (see text pg 77 or pg 81 8th edition).

Create a prompt using `System.out.print();` To ask the user for the customer name.

Declare a String object **customerName** to store the customer name and use your **inText** Scanner object and the inbuilt method `inText.nextLine();`

The customer name is now stored in the String object **customerName**.

We can now create a prompt using the customer name to ask for the number of garments to be cleaned

Hint: you can join variables and strings using the concatenation operator +

```
"Enter the number of garments for " + customerName + " ==> "
```

Declare an integer variable to store the number of garments and use your **inNumber** Scanner object and the inbuilt method `inNumber.nextInt();` to read the number of garments.

Declare a double variable to represent the charge for the order.

The arithmetic expression to calculate the total charge is very simple:

charge = number of garments * charge per garment

Note: the charge per garment must be stored as a constant (use the final keyword).

Finally output a receipt for the order as per the sample above.

The total charge must be displayed to two decimal points use printf and a format string as follows:

```
System.out.printf("$%.2f", charge);
```

Question three (week three topics) Decision statements

The management of Rocky Dry Cleaners would like to encourage customers to get more garments dry cleaned and to possibly get new customers. It has been decided to offer as a special, three garments to be dry cleaned for $20.00 which would be a saving of $5.50 and any subsequent garments in the order would be $6.50 per garment.

In summary:

One to two garments: $8.50 per garment.
Three garments: $20.00.
More than three garments: $20.00 plus $6.50 per garment after that.

Create a class **Week3** (file: Week3.java) and a main method and copy your code from question two into the main method of week three main.

After you have read the relevant details of the order i.e. name and number of garments, you will have to create a series of **if – else if** statements to calculate the final charge.

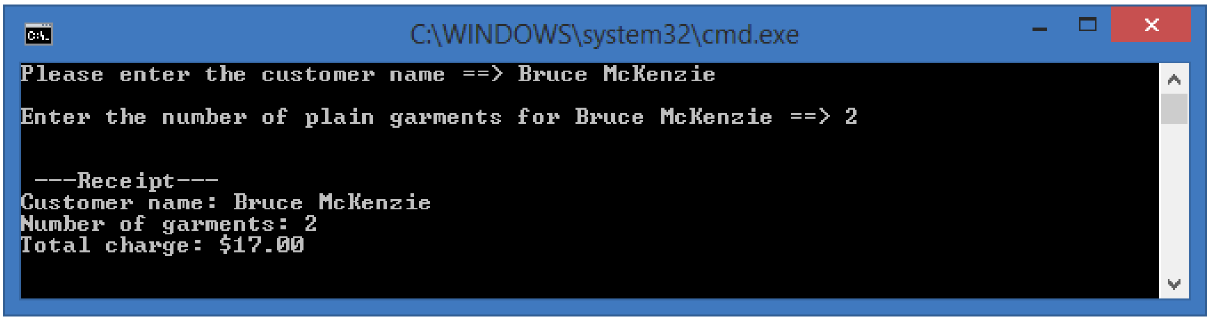
For over three garments;

charge = three garments charge + (number of garments – 3) * charge over three

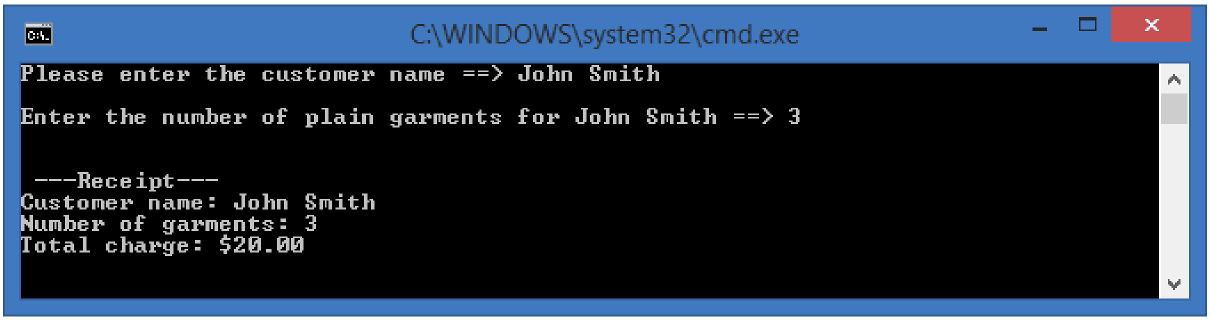
When you have calculated the total charge output a receipt as you have done in week two code.

Note: you must use constants for all the numeric literals in the else if statements.

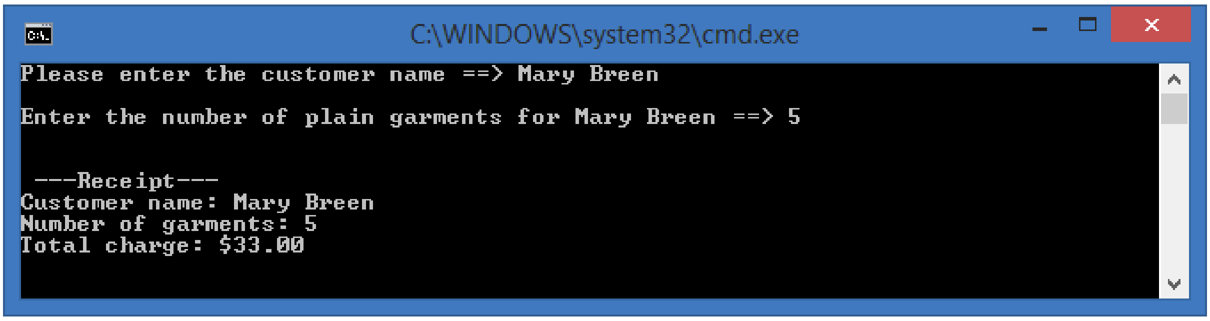
Your output needs to match exactly the output as shown below.



```
Please enter the customer name ==> Bruce McKenzie

Enter the number of plain garments for Bruce McKenzie ==> 2


 ---Receipt---
Customer name: Bruce McKenzie
Number of garments: 2
Total charge: $17.00
```



```
Please enter the customer name ==> John Smith

Enter the number of plain garments for John Smith ==> 3


 ---Receipt---
Customer name: John Smith
Number of garments: 3
Total charge: $20.00
```



```
Please enter the customer name ==> Mary Breen

Enter the number of plain garments for Mary Breen ==> 5


 ---Receipt---
Customer name: Mary Breen
Number of garments: 5
Total charge: $33.00
```

Question four (week four topics) Repetition while and for loops

Create a class **Week4** (file:Week4.java) to demonstrate the use of a repetition statement.

Implementation

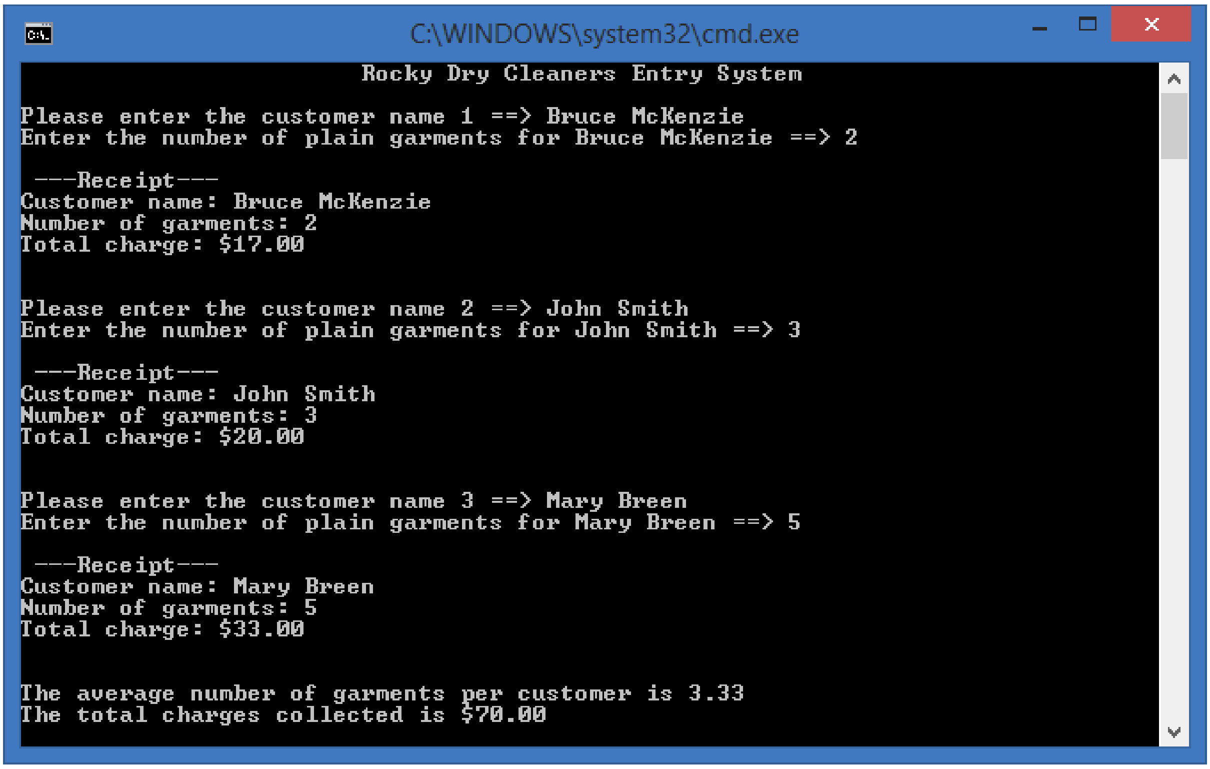
Using your solution to question three and a `while` or `for` loop, repeat the entry of customer names and the number of garments **N** times where **N is the largest digit in your student ID**, if your largest digit is less than three then let N = 3. Hint: use N = 3 while testing and submit using the correct N value.

N will be declared as an integer constant using the `final` keyword.

You are to print a title before the input of the customer names and number of garments (see sample output). Note the different line spacing. You will also number the customer in the customer name prompt.

Ensure you are using a separate Scanner objects for reading numbers and text (why?).

When all of the customer names and number of garments have been entered and their individual charges calculated, the average of the number of the garments for each order and the total of the charges collected will be reported. **Please note you do not need to store the data in an advanced structure such as an array**. You will need to have an integer variable to add up the number of garments to calculate the average, and a double variable to add up the charges.



```
                        Rocky Dry Cleaners Entry System

Please enter the customer name 1 ==> Bruce McKenzie
Enter the number of plain garments for Bruce McKenzie ==> 2

 ---Receipt---
Customer name: Bruce McKenzie
Number of garments: 2
Total charge: $17.00


Please enter the customer name 2 ==> John Smith
Enter the number of plain garments for John Smith ==> 3

 ---Receipt---
Customer name: John Smith
Number of garments: 3
Total charge: $20.00


Please enter the customer name 3 ==> Mary Breen
Enter the number of plain garments for Mary Breen ==> 5

 ---Receipt---
Customer name: Mary Breen
Number of garments: 5
Total charge: $33.00


The average number of garments per customer is 3.33
The total charges collected is $70.00
```

Your average garments per order calculation has to produce a floating point result. To get a floating point result you will need to promote one of the operands to a double.

i.e. `average = totalGarments * 1.0 / N`

Question five (week five topics) Methods and GUI I/O

Create a class **Week5** (file:Week5.java) by using your solution to question four. This question is identical to question four as the program will read in N customer names and number of garments and calculate the charges for the orders, however we are going to create a method to calculate the charges and we will be using GUI dialog boxes for our I/O.

Implementation

Methods

You will create a value returning method which will accept the number of garments as a parameter.

Use the following method header:

```
private static double calculateCharge(int garment)
```

Copy and paste your "if else if" code from question four for calculating the charge into the body of our new method calculateCharge. You should also copy the constants for the numeric literals into the method too. Use the `return` statement to return the charge.

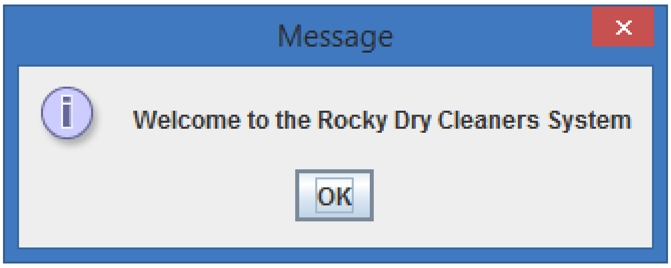
You can now use your method in the main method loop.

```
charge = calculateCharge(garments);
```

GUI I/O

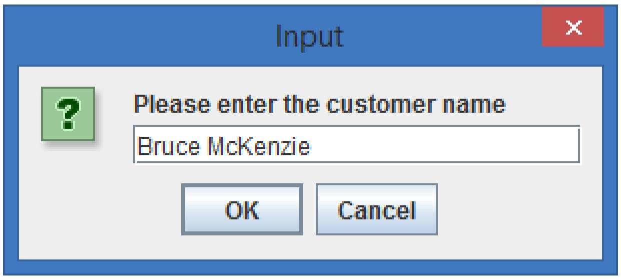
We will revisit the week two lecture topic using `JOptionPane` for accepting GUI input and outputting information.

First we will output a welcome message using `JOptionPane.showMessageDialog` (Replace your console print output).



Welcome to the Rocky Dry Cleaners System

Next we will replace the Scanner objects by using `JOptionPane.showInputDialog`.



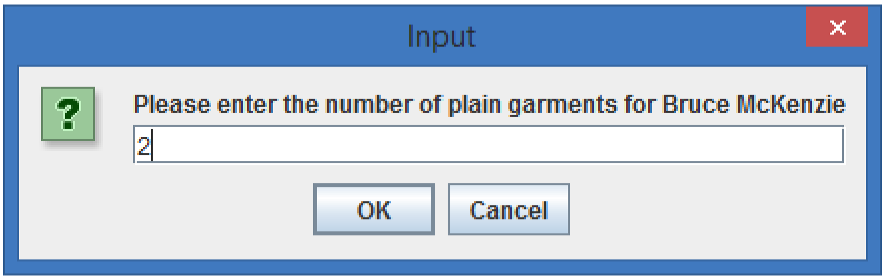
Please enter the customer name

Bruce McKenzie

The showInputDialog method will return the string entered into the dialog text field

```
customerName = JOptionPane.showInputDialog(null, "Prompt");
```

Next you will need to prompt for the number of garments.


Input
Please enter the number of plain garments for Bruce McKenzie
2
OK
Cancel

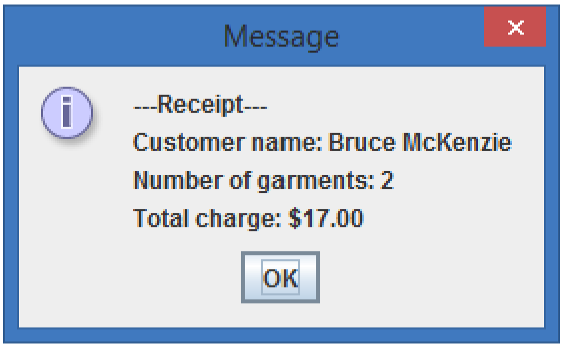
We receive input from the dialog as a string, in order to convert strings to an integer we need the Integer wrapper class and the parseInt method (text pg 347 or pg 370 8th Edition).

```
int anInteger =
      Integer.parseInt(JOptionPane.showInputDialog(null, "Prompt"));
```

After reading in and converting the number of garments to an integer you can use this value to calculate the charge for the order using your method:

```
charge = calculateCharge(garments);
```

Output the receipt using `JOptionPane.showMessageDialog(null, "text")`



---Receipt---
Customer name: Bruce McKenzie
Number of garments: 2
Total charge: $17.00

To format your output in the text argument in showMessagedialog you can use:

```
String.format("Format string",args)
```

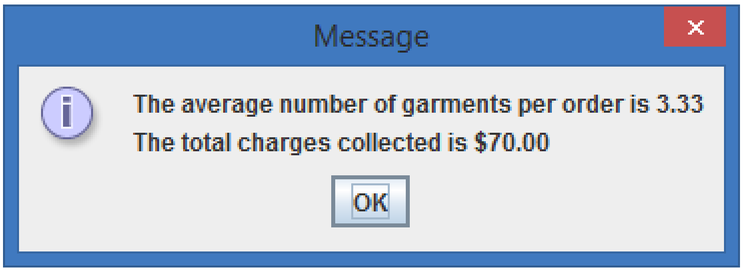
See the example below for using place holders to format strings, integers and doubles.

```
String.format("%s\n%d\n$%.2f",
      customerName, garments, charge)
```

`%s` is for a string.
`%d` is for an integer.
`%.2f` is for a floating point number (including double) formatted to two decimal places.

The `\n` will produce a newline and you will need to add extra text to the format string to match the output above.

When the N orders have been entered you will output the average number of garments per order and the total charges collected both to two decimal places, you can use a similar format string as above.



The average number of garments per order is 3.33
The total charges collected is $70.00

The marking scheme is on the next page.

Marking scheme

Total number of marks – 20	
Code in general	
Code is indented and aligned correctly, layout including vertical white space is good	1
Code has header comment which includes student name, student ID, date, file name and purpose of the class	0.5
Code is fully commented including all variables	1
Variables have meaningful names and use camel notation	0.5
Variables are the correct type	0.5
Question one	
Output as per specification	1
Question two	
Strings are read correctly using Scanner object	0.75
The integer is read correctly using a Scanner object	0.75
The order charge is computed and displayed correctly to two decimal places	0.5
Output is formatted correctly (matches sample output)	0.5
Question three	
If else statements are correct and constants are used	1.5
Correct charge is calculated and displayed correctly to two decimal places	0.5
Output is formatted correctly (matches sample output)	0.5
Question four	
Constant N used equal to highest digit in student ID	0.5
N customer names and number of garments are read in a loop	1
Program title "Rocky Dry Cleaners Entry System" printed	0.25
Charges printed for all orders	0.25
Average garments per order and total charges are calculated and printed correctly to two decimal places	1
Output is formatted correctly (matches sample output)	0.5
Question five	
Method implementation (uses a parameter)	1
Dollar (double) value returned from method correctly	0.5
Method call correct (uses an argument)	0.5
GUI welcome message	0.25
Strings are read correctly from GUI Input dialogs	0.25
Number of garments read correctly from GUI Input dialog and converted to an integer	1
N customer names and number of garments are read in a loop	0.25
Average and total are calculated and printed correctly to two decimal places	0.5
Dialogs appear as per specification (matches sample output)	0.25
General	
Correct files submitted including types and names (zip and Word)	0.5
Only techniques covered during weeks 1-5 and specification are used	0.5
Report	
Report presentation and comments including how long the programs took to create and any problems encountered	0.5
Screen shots of testing and annotations	1