Breaking it Down Problem

One problem that I find in new Java programmers is a tendency to put too much code in "main()". It is important to learn how to break a program down into methods and classes. To give you practice at this, I have a program that has too much code in main (It is called "TooMuchInMain").

Your job will be to restructure the code into appropriate methods to make it more readable. I will provide you with lots of coaching on how to do this. :-) See the "BreakingItDown Starting Point called " below for breaking down the "TooMuchInMain" program.

In order to show you what this looks like I have a program below that does too much in main. This program has the following characteristics:

- It repeatedly allows the user to select the mathematical operation desired and operands. Then the mathematical problem is calculated and printed out.
- It continues until the user enters a 'q' for quit.
- This program is robust in that it doesn't blow up even if the user enters in bad data. I will give you sample output of this program to show you this characteristic.
- Except for the fact that is long and all of the code is in main, it is better written and easier to understand than most programs that I see where the programmer has put too much code in main. However, it is harder to understand than it needs to be if we would just break the logic down to smaller methods.
- You can copy the code below and run it yourself to try it out. Here is some sample output that I have created:

```
Enter an Operator: +  -  *  /  q for quit: +
Enter operand 1
123
Enter operand 2
45
123 + 45 = 168
======================
Enter an Operator: +  -  *  /  q for quit: -
Enter operand 1
12x
Your last input was bad, try again
56
Enter operand 2
67
56 - 67 = -11
======================
Enter an Operator: +  -  *  /  q for quit: @
Your operator is bad ... try again:
Enter an Operator: +  -  *  /  q for quit: *
Enter operand 1
23!
Your last input was bad, try again
23
Enter operand 2
```

```
45
23 * 45 = 1035
======================
Enter an Operator: +  -  *  /  q for quit: q
Finished Calculations
```

TooMuchInMain program:

```
package breaking_it_down;

import java.util.Scanner;

public class TooMuchInMain {
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int operand1, operand2, answer;
        char operator='q';
        boolean keepCalculating = true;

         while (keepCalculating)
            {
             //************************************************
             // This next block will get the user desired operator
             // getOperator should contain the following logic
             //************************************************
                boolean operator_is_good=false;

                do
                {
                    System.out.print("Enter an Operator: +  -  *  /  q for
quit: ");
                    String strOperator = scan.nextLine();
                    strOperator = strOperator.trim();
                    if (strOperator.length() == 0)
                        continue; // Need to try this again with no input
                    operator = strOperator.charAt(0);
                    operator_is_good=false;

                    switch (operator)
                    {
                    case 'q':
                    case '+':
                    case '-':
                    case '*':
                    case '/':
                        operator_is_good = true;
                        break;
                    default:
                        System.out.println("Your operator is bad ... try
again:");
                        break;
                    }
                } while (!operator_is_good);
```

```
                //*********************************************
                // At this point operator contains a proper operator chosen
by the user
                // This will show up in doCalculation
                //*********************************************
                if (operator == 'q')
                {
                    keepCalculating=false; // not really needed because we
are doing a break... but clarifies what will happen
                    break;
                }

              //*********************************************
                // Now get operand 1
                // getOperand() will get the next hunk of logic
              //*********************************************
                int which =1;
                System.out.println("Enter operand "+which);
                String input;

                boolean operand_is_bad;

                do
                {
                    operand_is_bad=false;

                    input = scan.nextLine();
                    input = input.trim();
                    if (input.length() == 0)
                        operand_is_bad=true;
                    for (int i=0; i < input.length(); i++)
                    {
                        char c = input.charAt(i);
                        if (c < '0' || c > '9')
                        {
                            // Oops, bad digit
                            operand_is_bad=true;
                            System.out.println("Your last input was bad, try
again");

                        }
                    }
                } while (operand_is_bad);

               // The following statement will be covered later in the
course.
               // For now, just know that it converts a String to an integer
               // Note that Integer.parseInt is picky and blows up with any
bad characters
                operand1 =  Integer.parseInt(input);


                  //*********************************************
                    // Now get operand 2
                    // Note this code is mostly the same as the above code.
                    // You can just reuse the same getOperand method again
                  //*********************************************
                    which =2;
```

```
                    System.out.println("Enter operand "+which);
                    do
                    {
                        operand_is_bad=false;

                        input = scan.nextLine();
                        input = input.trim();
                        if (input.length() == 0)
                            operand_is_bad=true;
                        for (int i=0; i < input.length(); i++)
                        {
                            char c = input.charAt(i);
                            if (c < '0' || c > '9')
                            {
                                // Oops, bad digit
                                operand_is_bad=true;
                                System.out.println("Your last input was bad,
try again");
                            }
                        }
                    } while (operand_is_bad);

                // The following statement will be covered later in the
course.
                // For now, just know that it converts a String to an integer
                // Note that Integer.parseInt is picky and blows up with any
bad characters

                    operand2 =  Integer.parseInt(input);

                    //***********************************************
                    // Now we have operator, operand1, and operand2.
                    // Time to calculate the answer
                    // doArithmetic will take the following logic
                  //***********************************************
                    switch(operator)
                    {
                    case '+':
                        answer = operand1 + operand2;
                        break;
                    case '-':
                        answer = operand1 - operand2;
                        break;
                    case '*':
                        answer = operand1 * operand2;
                        break;
                    case '/':
                        answer = operand1 / operand2;
                        break;
                    default:
                        System.out.println("We shouldn't get here in
doArithmentic!!!!");
                        answer = -1;
                        break;
                    }

                    //***********************************************
```

```
                    // Time to format the answer
                    // the format routine will get the following statement
                    //*****************************************
                    System.out.println(operand1 + " "+ operator + " "+
operand2 + " = "+ answer);

                System.out.println("======================");
            } // end of while (continue_calculating)

            System.out.println("Finished Calculations");

    } // end of main


} // end of class TooMuchInMain
```

BreakingItDown Starting Point for your solution where you have the above logic broken down:

Copy into your JH3_worksheet.txt file output from YOUR program that replicates the output shown above for the TooMuchInMain program.

HINT: Note you can copy a lot of code segments from the "TooMuchInMain" into your "BreakingItDown" program.

```
package breaking_it_down;

import java.util.Scanner;

public class BreakingItDown {
    // instance variable
    Scanner scan = new Scanner(System.in);

    // Top Level routine for doing a single calculation
    boolean doCalculation()
    {
        int operand1, operand2, answer;
        char operator = getOperator();
        if (operator == 'q')
            return false;

        operand1 = getOperand(1);
        operand2 = getOperand(2);
        answer = doArithmetic(operator, operand1, operand2);
        format(operator, operand1, operand2, answer);

        return true;
    }

    // Get the Operator reliably
    char getOperator()
    {
```

```
        char operator='q';

     // **********************************
     // Logic to get a correct operator from the user goes here
     // **********************************

        return operator;
    }


    // Get one operand reliably
    int getOperand(int which)
    {
        System.out.println("Enter operand "+which);
        String input;

     // **********************************
     // Logic to get a correct  operand from the user goes here.
     // Hint: collect the users operand into the "input" variable
     // Remove excess whitespace (trim call).  Make sure all characters are
     // digits (i.e. between '0' and '9').  If not, then ask the user for
     // another input.  Once you have a good "input" string, the statement
     // below will convert it to an integer.
     // **********************************


     // The following statement will be covered later in the course.
     // For now, just know that it converts a String to an integer
     // Note that Integer.parseInt is picky and blows up with any bad
characters
     // **********************************
        return Integer.parseInt(input);
    }



    // Do the Actual Calculation required
    int doArithmetic(char operator, int operand1, int operand2)
    {
        int answer;
        // **********************************
        // Your logic to compute the answer
        // **********************************
        return answer;
    }

    // Format the final results
    void format(char operator, int operand1, int operand2, int answer)
    {

        // **********************************
        // Your logic to format the result.  For example:
        // 23 * 45 = 1035
        // **********************************
    }

    // We like to keep minimal code in main
```

```
    public static void main(String[] args)
    {
        // Construct our class
        BreakingItDown bid = new BreakingItDown();

        // Call doCalculation for each calculation we are doing.
        while (bid.doCalculation())
        {
           System.out.println("======================");
        } // end of while (continue_calculating)

        System.out.println("Finished Calculations");

    }

}
```