# Mastery Metric Conversion Reflection Log

Credit Name: CSE 2110 Procedural Programming 1
Assignment Name: Mastery Metric Conversion

## (note: the screenshots are mixed between Eclipse and a Java compiler because I did some of the log at home)

How has your program changed from planning to coding to now? Please explain?

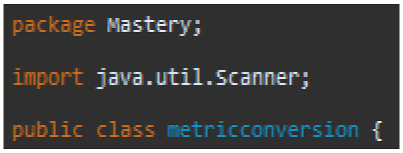
From planning to coding I took the IPO chart that I made and started to flush out each line into code. Each part became more complex as I made a line of code for each conversion and code for the calculation to do the conversion. I also made a switch statement in order to efficiently go through each different case/choice (you could use an if/else statement/s, but the switch statement is quicker.

From coding to now I did not change anything really, all I did was create and edit the error log during my time coding and a little bit after finishing. I also changed the print statement “Select a unit conversion:” to “Select a unit conversion (enter the number that correlates to your conversion):”

Steps

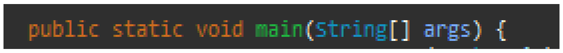
(the package this file is in is Mastery)

I started off by importing java.util.scanner which allows me to get different data types from inputs. The public class is declared as metricconversion (public makes it accessible from anywhere).



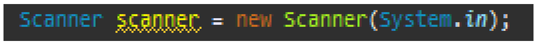
```
package Mastery;

import java.util.Scanner;

public class metricconversion {
```

Below it shows the public method “main” is static (you can access it without creating an object), void (method doesn’t return any value), main (method name), and String[ ] args (String represents the data type [ ] indicates that it will be an array, and args (arguments) is the name of the array).



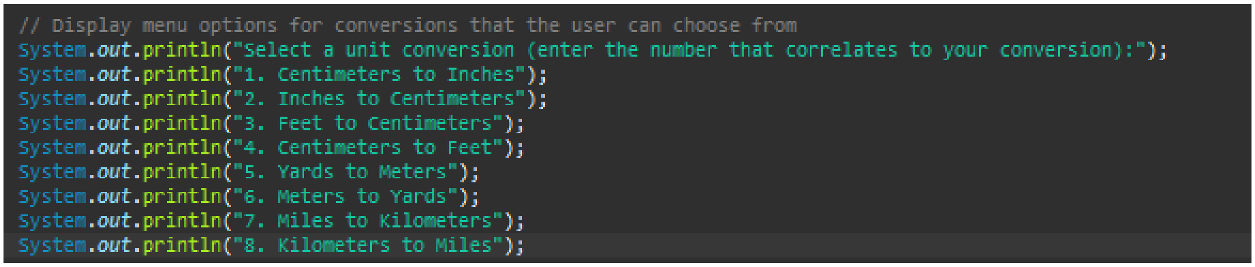
```
public static void main(String[] args) {
```

The scanner object input is created below to take in inputs from the user.



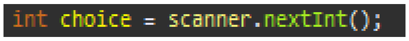
```
Scanner scanner = new Scanner(System.in);
```

Next, I created 8 different prompt statements, these include each of the different conversions that the user can choose from (Ex. feet to centimeters ect.) I also used println so they all appeared on different lines in a neat format. The print statement at the top tells the user to select a unit conversion (if they select 1 then their conversion is centimeters to inches)



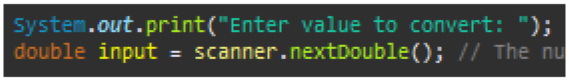
```
// Display menu options for conversions that the user can choose from
System.out.println("Select a unit conversion (enter the number that correlates to your conversion):");
System.out.println("1. Centimeters to Inches");
System.out.println("2. Inches to Centimeters");
System.out.println("3. Feet to Centimeters");
System.out.println("4. Centimeters to Feet");
System.out.println("5. Yards to Meters");
System.out.println("6. Meters to Yards");
System.out.println("7. Miles to Kilometers");
System.out.println("8. Kilometers to Miles");
```

After this I made an integer variable called choice that was equal to the number that the user selected for their unit conversion. (will be used to determine which switch case is selected)



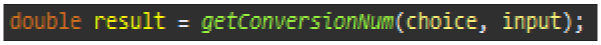
```
int choice = scanner.nextInt();
```

Next, I prompted the user to enter the value(number) the want to convert and saved their answer as a double variable called input.



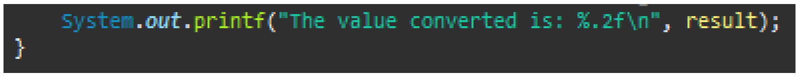
```
System.out.print("Enter value to convert: ");
double input = scanner.nextDouble(); // The nu
```

I then made a double variable called result that was equal to the result of the calculations in the method getConversionNum that used choice and input parameters.



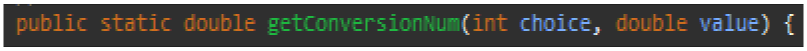
```
double result = getConversionNum(choice, input);
```

After this, I printed out the converted value (result) and rounded it to 2 decimal places using %.2f, I also closed the main method using a brace so that I could begin the next method.



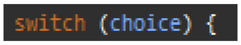
```
    System.out.printf("The value converted is: %.2f\n", result);
}
```

Next, I created the second method called getConversionNum which I passed the parameters int choice (to determine which switch statement to use) and double value (the number that is going to be converted). This method was public static, and double (so that it can return a decimal value)



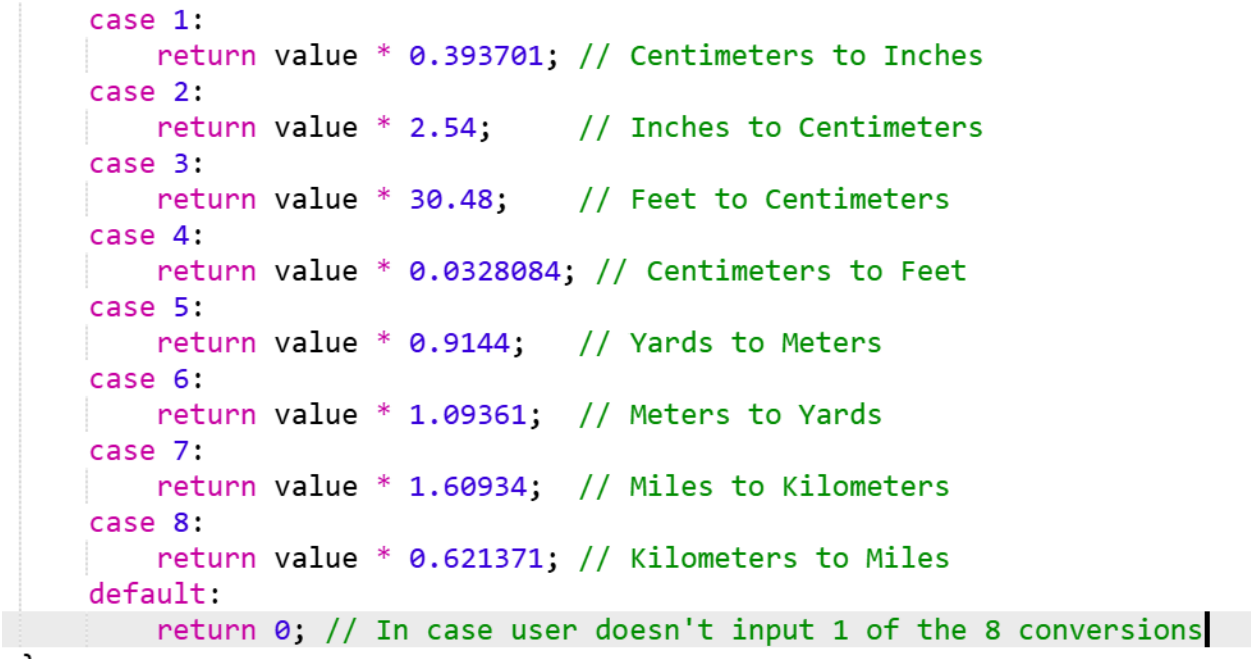
```
public static double getConversionNum(int choice, double value) {
```

I then made the switch statement which took in the user input choice to select one of the 8 cases (note if something other than 1-8 was chosen the default case would return 0 because the user didn't select anything within the range of conversion choices.



```
switch (choice) {
```

Next, I made the 8 cases one for each conversion, plus an extra default case in case something invalid was imputed. Ex. if conversion 1 was selected then case 1 with conversion centimeters to inches would be selected. This has the specified conversion rate * the user's value which results in the new converted value, which is used in the output statement(there is a comment describing which each conversion is). (default case in case any of the 9 conversions isn't selected)



```
    case 1:
        return value * 0.393701; // Centimeters to Inches
    case 2:
        return value * 2.54;     // Inches to Centimeters
    case 3:
        return value * 30.48;    // Feet to Centimeters
    case 4:
        return value * 0.0328084; // Centimeters to Feet
    case 5:
        return value * 0.9144;   // Yards to Meters
    case 6:
        return value * 1.09361;  // Meters to Yards
    case 7:
        return value * 1.60934;  // Miles to Kilometers
    case 8:
        return value * 0.621371; // Kilometers to Miles
    default:
        return 0; // In case user doesn't input 1 of the 8 conversions
```

After I closed the switch statement with a brace

```
}
```

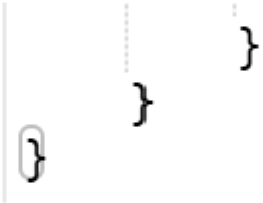
I then closed the getConversionNum method with a brace.

```
}
```

And lastly, I closed the public class metricconversion

```
}
```

Together they looked like this.



```
        }
    }
}
```