First I started by reading all data within the FIle I had set that fits within the ASCII notation parameters which are equal to the entirety of the English alphabet and then printed both the number of words and the word length.

```
try {
    dataFile = new File("C:\\Users\\77132002\\git\\CS30P3F2024\\Chapter11\\src\\skillBuilders\\WordCount.txt");
    in = new FileReader(dataFile);
    readFile = new BufferedReader(in);

    while (readFile.ready()){
        int notation = readFile.read();

        //Correlation of # between ASCCI notation
        if(notation >= 65 && notation <= 90 || notation >= 97 && notation <= 122)
        {
            checkLetter = true;
            charNum++;
        }



        else
        {
            if(checkLetter)
            {
                wordNum++;
            }

            avgLength = format.format((double)charNum/wordNum);


        }

    }

    System.out.println("# of words: " + wordNum);
    System.out.println("Average word length: " + avgLength + " letters");


}
```

And to end it off I closed the try catch statement with two catch statements pertaining to the visibility of the file to the program and an average IOException for any errors that may occur.

```
catch (FileNotFoundException e)
{
    System.out.println("File does not exist or could not be found.");
    System.err.println("FileNotFoundException: " + e.getMessage());

}
catch (IOException e)
{
    System.out.println("Problem ready file");
    System.err.println("IOException: " + e.getMessage());
}
```