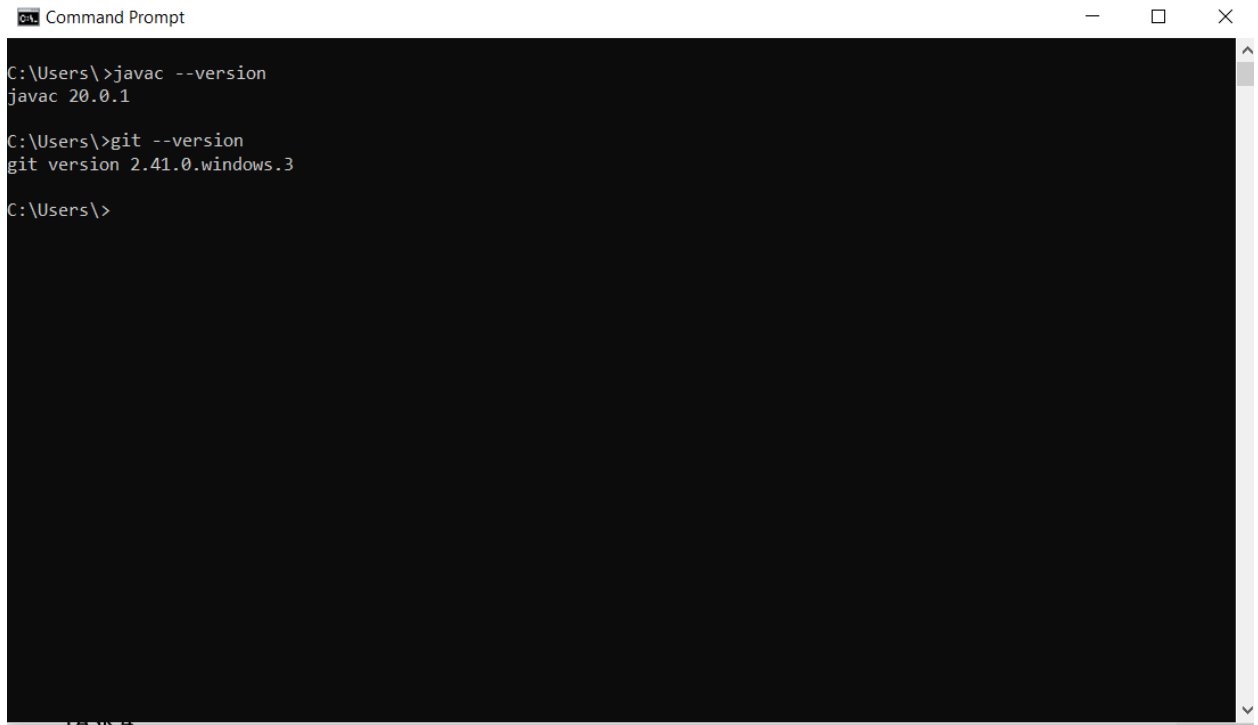




```
Command Prompt
C:\Users\ >javac --version
javac 20.0.1
C:\Users\ >
```



The image shows a Windows Command Prompt window titled "Command Prompt". The window has a standard Windows title bar with minimize, maximize, and close buttons. The command history is as follows:

```
C:\Users\>javac --version
javac 20.0.1

C:\Users\>git --version
git version 2.41.0.windows.3

C:\Users\>
```

At the bottom left of the image, the text "TASK 4." is partially visible.

The screenshot shows an IDE window with three tabs: 'pom.xml (lab1a)', 'ProductMgmtApp.java', and 'Product.java'. The 'ProductMgmtApp.java' tab is active, displaying Java code. A search bar at the top shows the search term 'json' with 4 results. The code includes imports for various Java utilities and a main method that calls 'Product.getData()' and 'printProducts()'. The 'printProducts' method uses 'Arrays.stream()' and 'Collectors.toList()' to process a list of products, and then calls 'printJSON()', 'printXML()', and 'printCSV()'. The 'printJSON' method is also shown, using 'Gson.toJson()' to convert the product list to JSON. The left sidebar shows the 'Project' and 'Structure' views.

```
6 import java.lang.reflect.Array;
7 import java.util.Arrays;
8 import java.util.Collections;
9 import java.util.Comparator;
10 import java.util.List;
11 import java.util.stream.Collectors;
12
13 public class ProductMgmtApp {
14     public static void main(String[] args) {
15         Product products [] = Product.getData();
16         printProducts(products);
17     }
18 }
19
20 1 usage
21 static void printProducts(Product[] products){
22     List<Product> productList = Arrays.stream(products)
23         .sorted(Comparator.comparing(Product::getName))
24         .collect(Collectors.toList());
25     printJSON(productList);
26     printXML(productList);
27     printCSV(productList);
28 }
29
30 1 usage
31 public static void printJSON(List<Product> products){
32     System.out.println("JSON format:\n"+new Gson().toJson(products));
33 }
```

```
Run ProductMgmtApp

JSON format:
[{"productId":2927458265,"name":"Apple","dateSupplied":"2022-12-09","quantityInStock":18,"unitPrice":1.09}, {"productId":3128874119,"name":"Banana","dateSupplied":"2023-01-24","qu

XML format:
<Products>
  <Product>
    <productId>2927458265</productId><name>Apple</name><dateSupplied>2022-12-09</dateSupplied><quantityInStock>18</quantityInStock><unitPrice>1.09</unitPrice>
  </Product>
  <Product>
    <productId>3128874119</productId><name>Banana</name><dateSupplied>2023-01-24</dateSupplied><quantityInStock>124</quantityInStock><unitPrice>0.55</unitPrice>
  </Product>
  <Product>
    <productId>9189927460</productId><name>Carrot</name><dateSupplied>2023-03-31</dateSupplied><quantityInStock>89</quantityInStock><unitPrice>2.99</unitPrice>
  </Product>
</Products>

CSV format

2927458265, Apple, 2022-12-09, 18, 1.09
3128874119, Banana, 2023-01-24, 124, 0.55
9189927460, Carrot, 2023-03-31, 89, 2.99

Process finished with exit code 0
|
```