BITP 3113
OBJECT ORIENTED PROGRAMMING

LAB:
WEEK 03 – USING JAVA

LECTURER NAME:
DR EMALIANA BINTI KASMURI

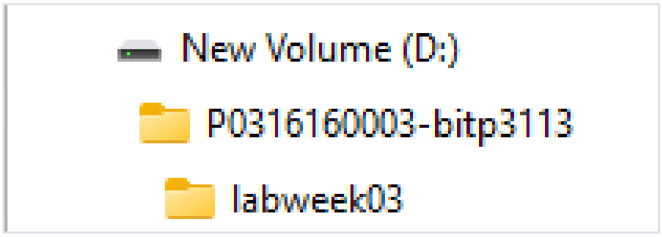
STUDENT NAME:
MUHAMMAD AFIQ MUHAIMIN BIN MOHD ZAINI

MATRIC NUMBER:
B032410648

SEMESTER: 2-2024/25

Exercise 6: Preparing Lab Exercise Environment

1. Create a folder named **bitp3113** on your computer. Preferably in the **C:** or **D:** drive.
2. Prefix the folder name with your matric number. For example, P0316160003-bitp3113.
3. Create a subfolder named **labweek03**. The structure should be similar as shown in Figure 10.



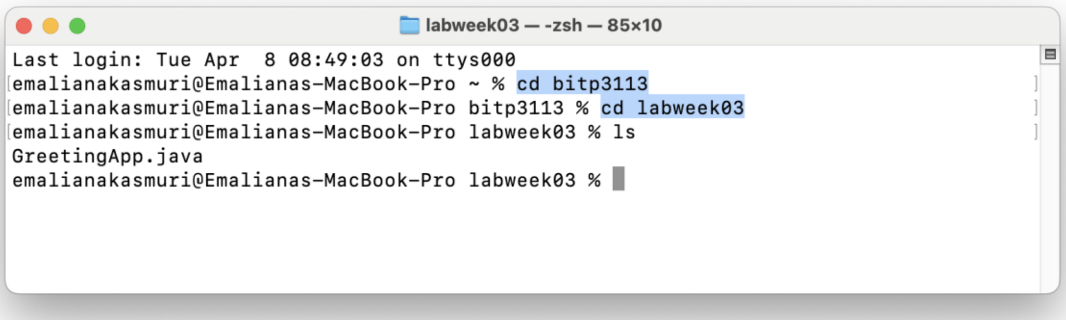
```
New Volume (D:)
  P0316160003-bitp3113
    labweek03
```

Figure 10: Lab exercise folder structure

This folder will store all Java codes related to lab exercises in week 03.

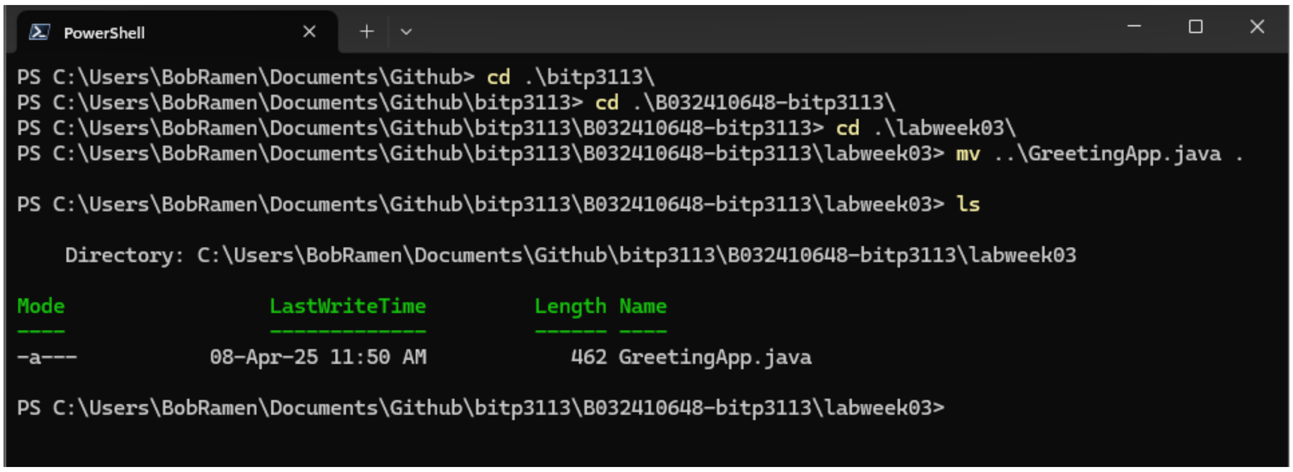
Exercise 7: Execute a Java Program

1. Download the **GreetingApp.java** from ulearn.
2. Move the file into folder named **labweek03**.
3. Open **MS Prompt** (for Windows) or **Terminal** (for MacOS) from the computer.
4. Change the directory to **labweek03** using the `cd` command. The outcome should be similar as shown in Figure 11.



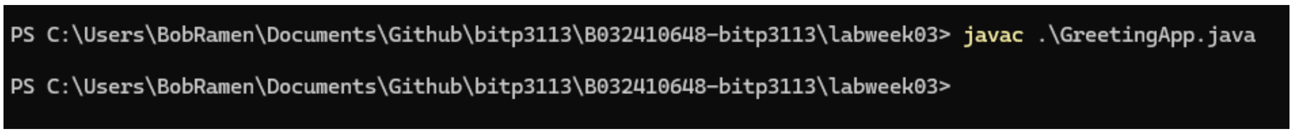
```
Last login: Tue Apr  8 08:49:03 on ttys000
emalianakasmuri@Emalianas-MacBook-Pro ~ % cd bitp3113
emalianakasmuri@Emalianas-MacBook-Pro bitp3113 % cd labweek03
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

Figure 11: The `cd` command in Terminal



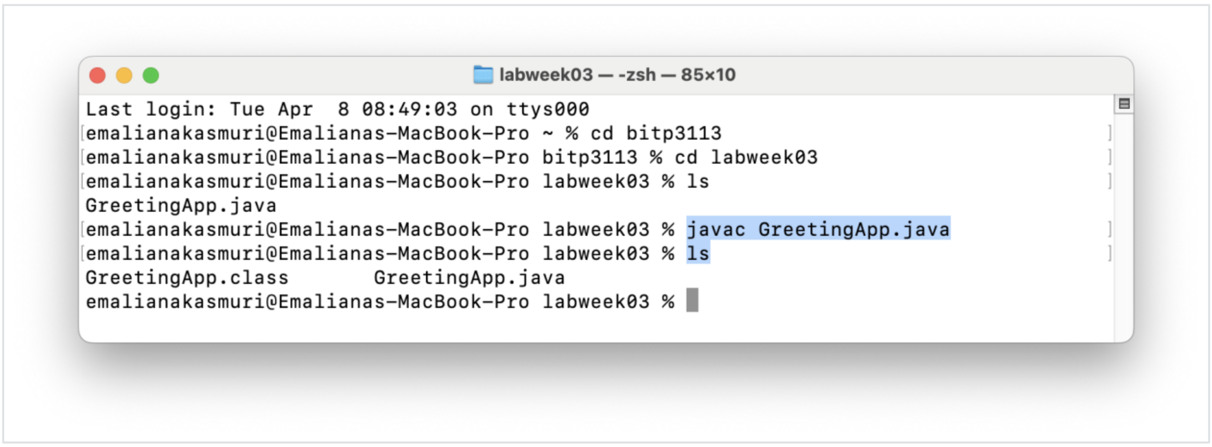
```
PS C:\Users\BobRamen\Documents\Github> cd .\bitp3113\
PS C:\Users\BobRamen\Documents\Github\bitp3113> cd .\B032410648-bitp3113\
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113> cd .\labweek03\
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> mv ..\GreetingApp.java .

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          08-Apr-25 11:50 AM            462 GreetingApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

5. Type the `javac GreetingApp.java` command on the terminal to compile the Java class.



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\GreetingApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

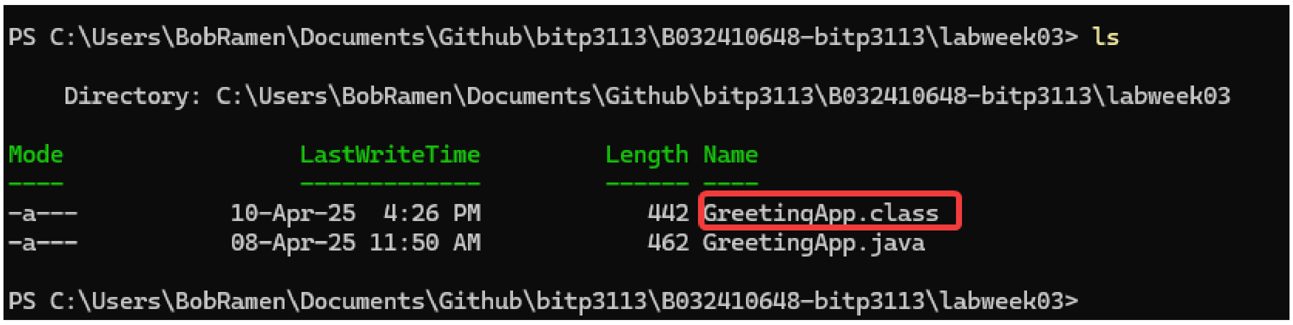
6. Then, type the `ls` command on the terminal to view list of files in the directory. The outcome should be similar as shown in Figure 12.



```
labweek03 — -zsh — 85×10
Last login: Tue Apr  8 08:49:03 on ttys000
emalianakasmuri@Emalianas-MacBook-Pro ~ % cd bitp3113
emalianakasmuri@Emalianas-MacBook-Pro bitp3113 % cd labweek03
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % javac GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
GreetingApp.class       GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

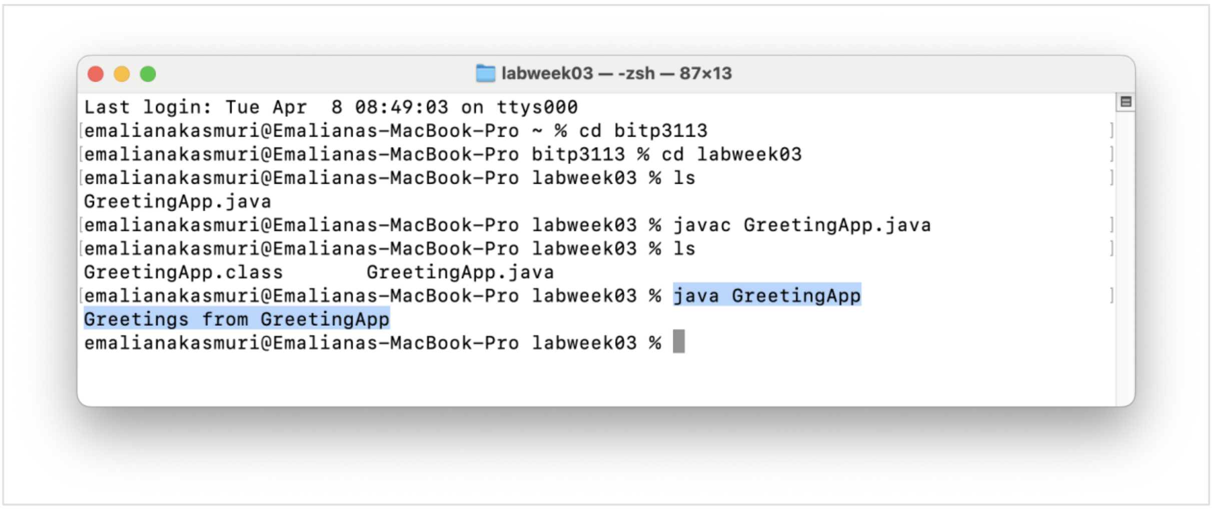
Figure 12: Outcome from `javac` command on GreetingApp.java

The **.class** file existence indicate the Java file is successfully compiled.



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          10-Apr-25   4:26 PM            442 GreetingApp.class
-a---          08-Apr-25  11:50 AM            462 GreetingApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

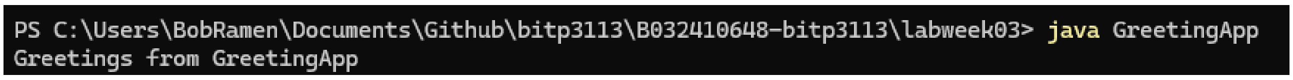
7. After that, type the `java Greeting` command in the terminal. The outcome should be similar as shown in Figure 13.



```
labweek03 — -zsh — 87×13
Last login: Tue Apr  8 08:49:03 on ttys000
emalianakasmuri@Emalianas-MacBook-Pro ~ % cd bitp3113
emalianakasmuri@Emalianas-MacBook-Pro bitp3113 % cd labweek03
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % javac GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
GreetingApp.class       GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % java GreetingApp
Greetings from GreetingApp
emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

Figure 13: The outcome from java command on GreetingApp

The greeting statement from the **GreetingApp** marks your first successful Java application execution. Congratulations!



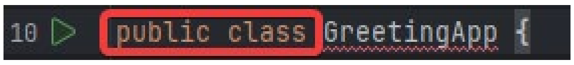
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java GreetingApp
Greetings from GreetingApp
```

Exercise 8: Observe the Java Program

Java used the same syntax as C/C++. The program is written using a combination of Java keywords and method calling. The following are the steps to observe a program.

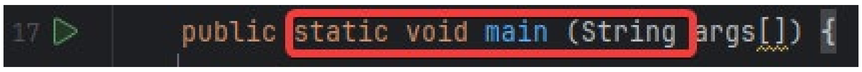
1. Open **GreetingApp.java** in Notepad or TextEditor.
2. Turn on the line number from the editor.
3. **Locate the following keyword from the source code.**

```
a. public = Line 10
b. class = Line 10
```



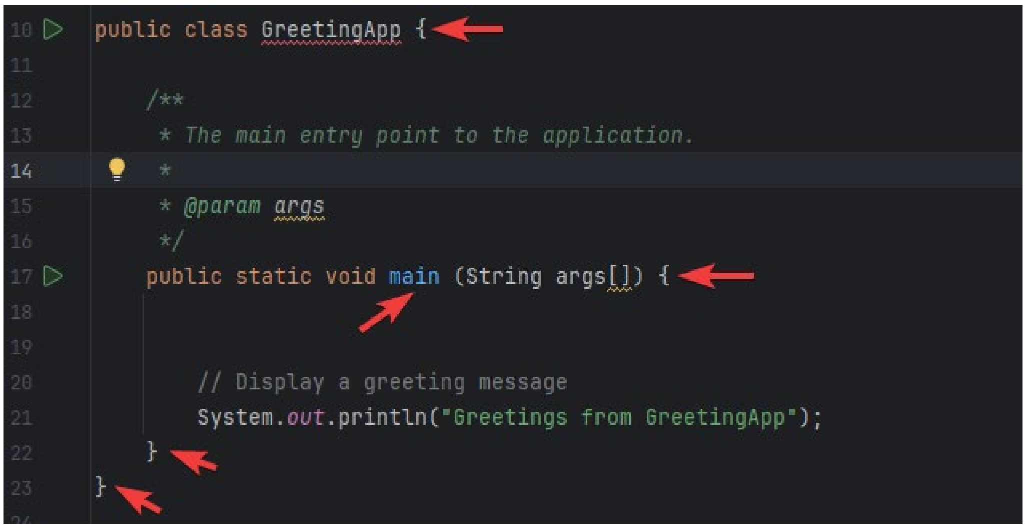
```
10  public class GreetingApp {
```

```
c. static = Line 17
d. void = Line 17
e. String = Line 17
f. Main = Line 17
```



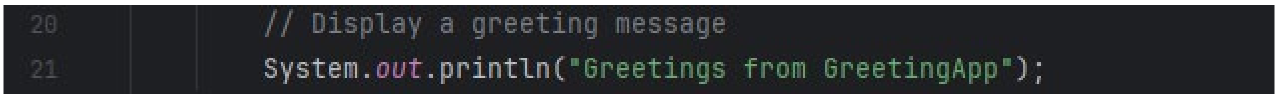
```
17      public static void main (String args[]) {
```

4. **Identify the curly brackets `{ }` for the class and the `main()` method.**



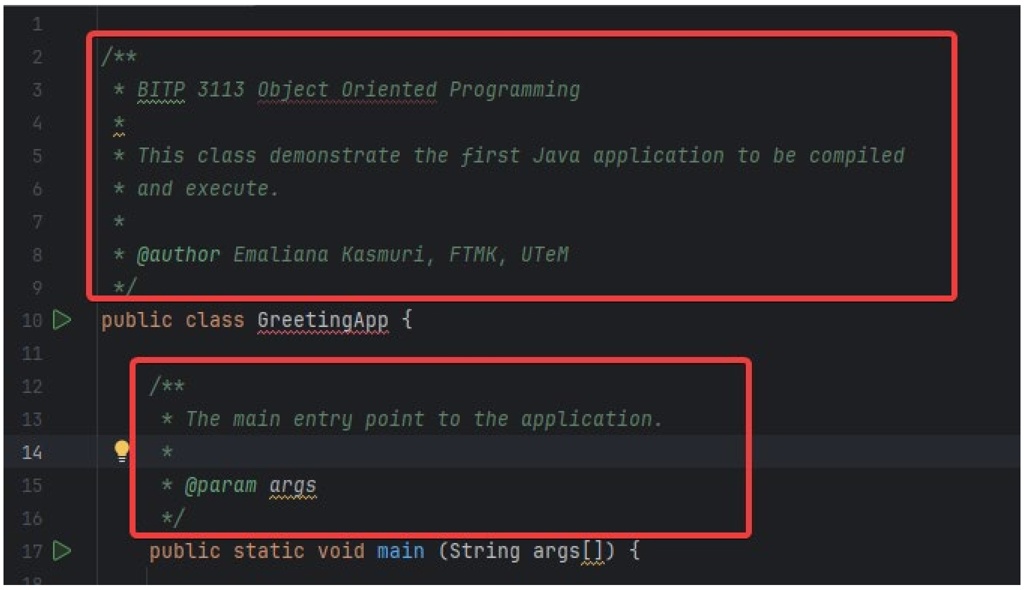
```
10  public class GreetingApp {
11
12      /**
13       * The main entry point to the application.
14       *
15       * @param args
16       */
17      public static void main (String args[]) {
18
19
20          // Display a greeting message
21          System.out.println("Greetings from GreetingApp");
22      }
23  }
```

5. **Identify the method to display the greeting message on the console.**
   Line 21



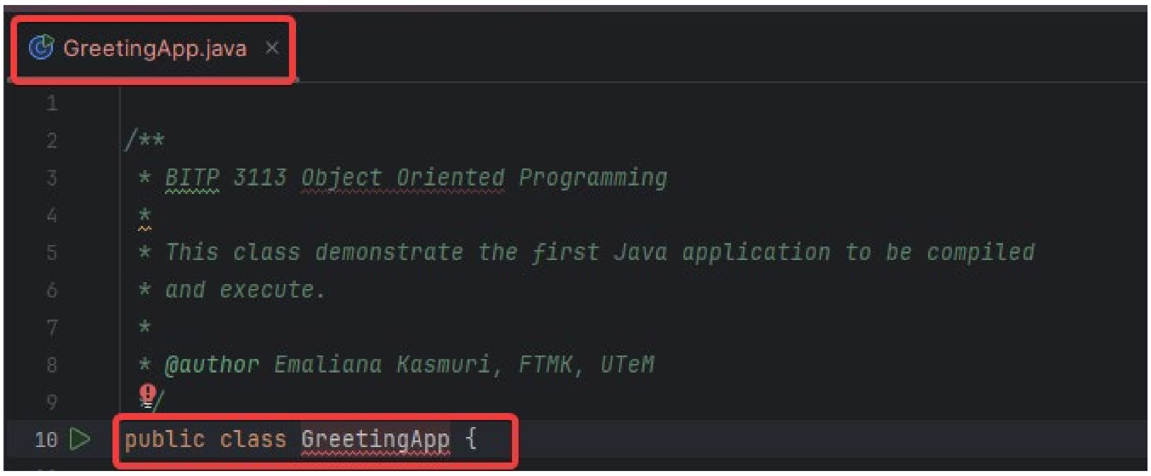
```
20          // Display a greeting message
21          System.out.println("Greetings from GreetingApp");
```

6. **Identify the comment block.**
   Line 2 – 9 and Line 12 – 16



```
1
2   /**
3    * BITP 3113 Object Oriented Programming
4    *
5    * This class demonstrate the first Java application to be compiled
6    * and execute.
7    *
8    * @author Emaliana Kasmuri, FTMK, UTeM
9    */
10  public class GreetingApp {
11
12      /**
13       * The main entry point to the application.
14       *
15       * @param args
16       */
17      public static void main (String args[]) {
```

7. **Observe the class and the file name.**
   The class name and the file name is the same which is GreetingApp



GreetingApp.java

```
1
2   /**
3    * BITP 3113 Object Oriented Programming
4    *
5    * This class demonstrate the first Java application to be compiled
6    * and execute.
7    *
8    * @author Emaliana Kasmuri, FTMK, UTeM
9    */
10  public class GreetingApp {
```

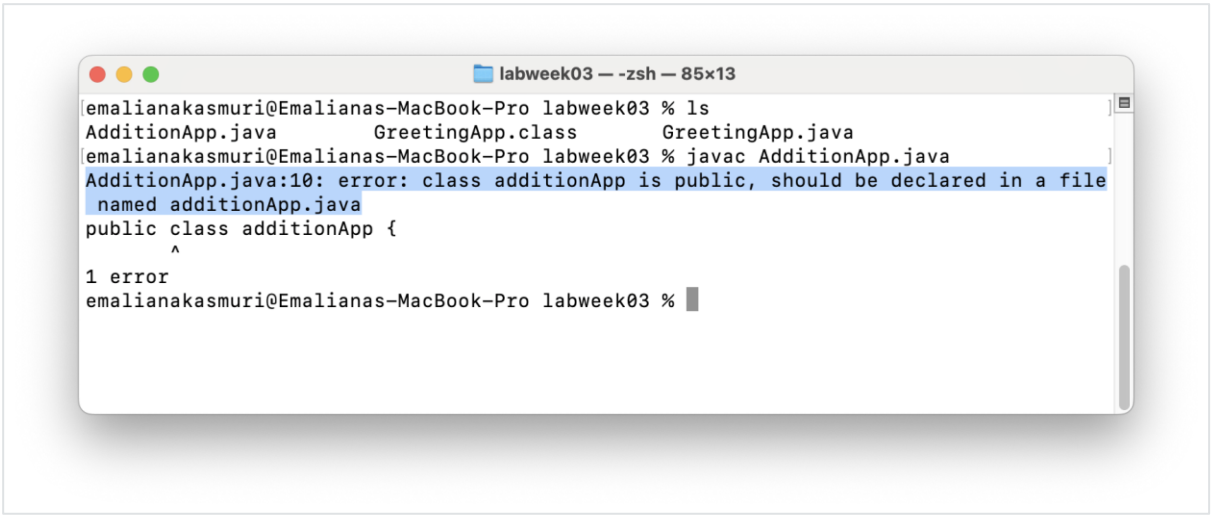
8. Get yourself familiar with the source code and the programming style.

Java Class Name and File

The Java class name must be the same as file name. The class must be saved with `.java` extension, as shown in the previous example. A good class name shall always start with an upper-case letter, for example `GreetingApp`.

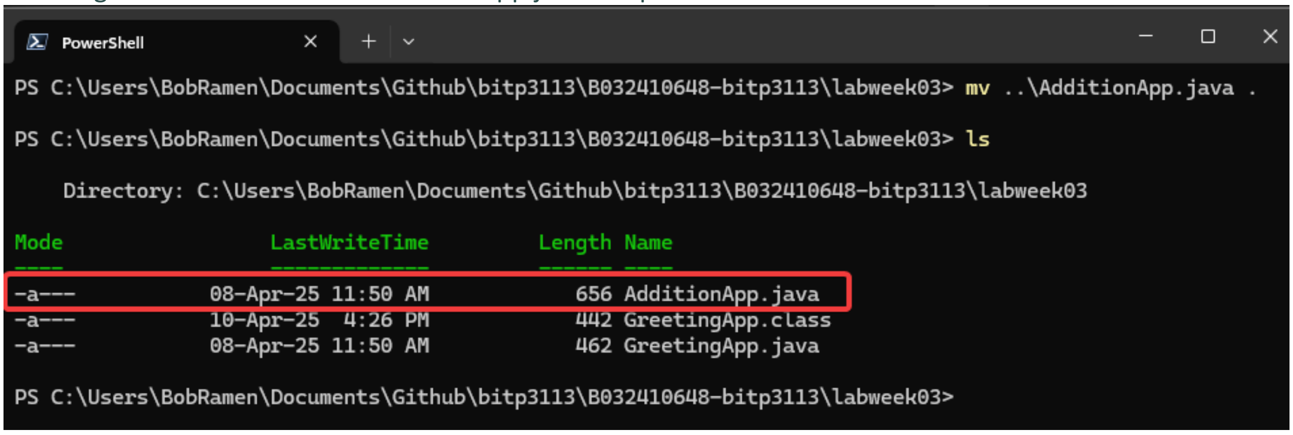
Exercise 9: Producing Compilation Error

1. Download a file named **AdditionApp.java**.
2. Move the file into folder named **labweek03**.
3. Compile the file using `javac` command. The outcome shall be similar as shown in Figure 14.

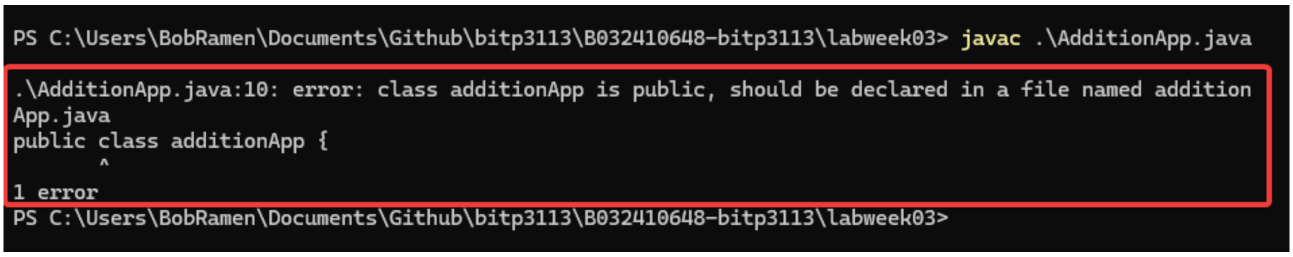


```
labweek03 — -zsh — 85×13
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
AdditionApp.java        GreetingApp.class       GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % javac AdditionApp.java
AdditionApp.java:10: error: class additionApp is public, should be declared in a file
 named additionApp.java
public class additionApp {
       ^
1 error
emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

Figure 14: Outcome from AdditionApp.java compilation

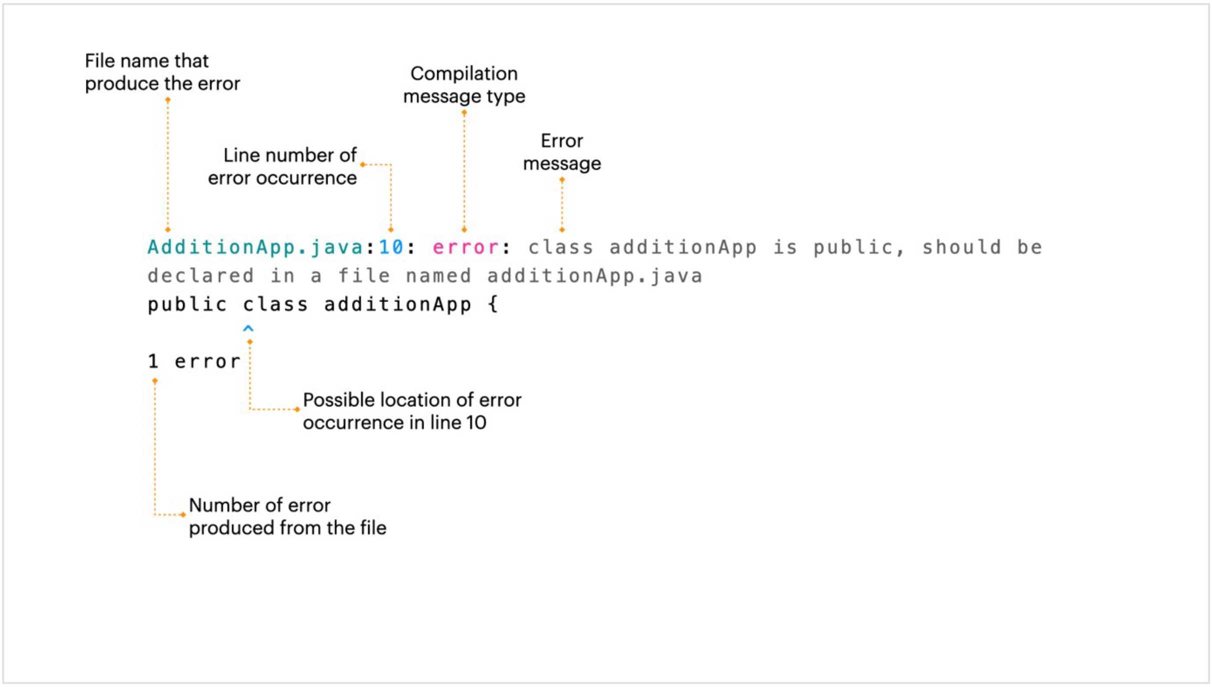


```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> mv ..\AdditionApp.java .
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          08-Apr-25 11:50 AM            656 AdditionApp.java
-a---          10-Apr-25  4:26 PM            442 GreetingApp.class
-a---          08-Apr-25 11:50 AM            462 GreetingApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\AdditionApp.java
.\AdditionApp.java:10: error: class additionApp is public, should be declared in a file named addition
App.java
public class additionApp {
       ^
1 error
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

4. The compilation shall produce 1 error message as highlighted in Figure 14. Observe the error message anatomy in Figure 15.



```
AdditionApp.java:10: error: class additionApp is public, should be
declared in a file named additionApp.java
public class additionApp {
       ^
1 error
```

File name that produce the error
Compilation message type
Line number of error occurrence
Error message
Possible location of error occurrence in line 10
Number of error produced from the file

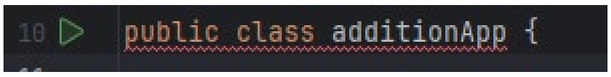
Figure 15: Anatomy of Java compilation error

Exercise 10: Debugging the Compilation Error

1. Open **AdditionApp.java** using the previous text editor.
2. Turn on the line number.
3. Bring the cursor the line where the error has occurred.
4. **Apply the knowledge comprehended on Java class and file name to fix the error.**

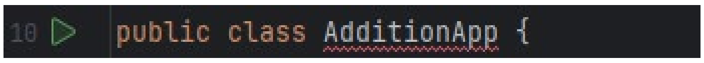
   Change the classname from additionApp to AdditionApp

   Before



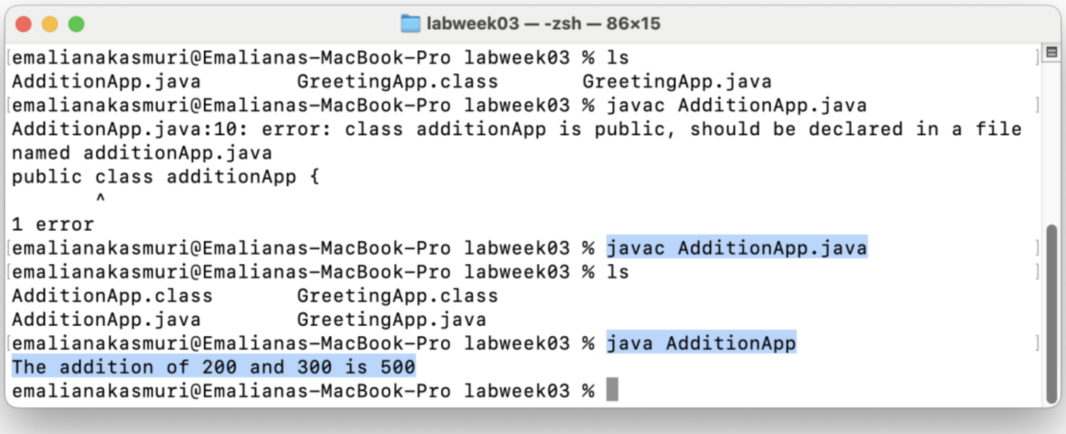
```
10  public class additionApp {
```

   After



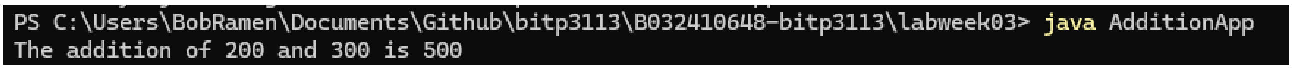
```
10  public class AdditionApp {
```

5. Save the file.
6. Compile the file.
7. Execute the application. The outcome shall be similar as shown in Figure 16.



```
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
AdditionApp.java        GreetingApp.class       GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % javac AdditionApp.java
AdditionApp.java:10: error: class additionApp is public, should be declared in a file
named additionApp.java
public class additionApp {
       ^
1 error
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % javac AdditionApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % ls
AdditionApp.class       GreetingApp.class
AdditionApp.java        GreetingApp.java
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % java AdditionApp
The addition of 200 and 300 is 500
emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

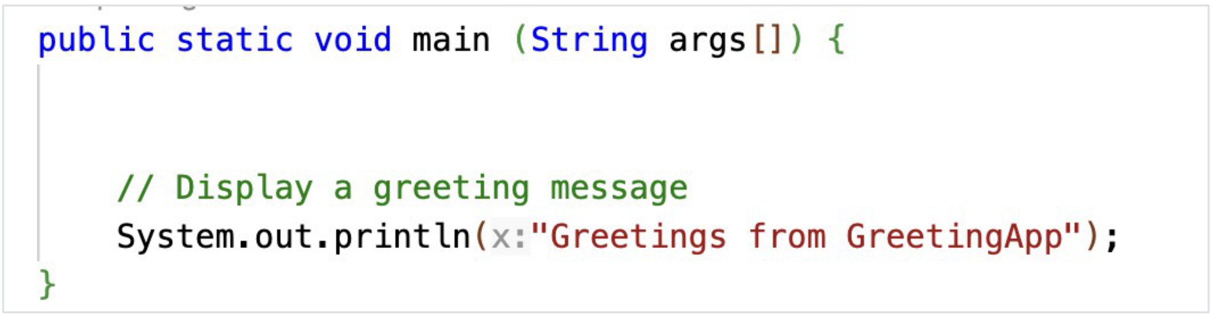
Figure 16: Outcome from AdditionApp execution



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java AdditionApp
The addition of 200 and 300 is 500
```

The main() Method

The `main( )` method is the entry point to any executed Java application. The method is declared within a Java class. The method is declared using a combination of reserved word – `public`, `static`, `void` and `main`, as shown in Figure 17. It may receive none or any number of execution variables.



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


    // Display a greeting message
    System.out.println(x:"Greetings from GreetingApp");
}
```

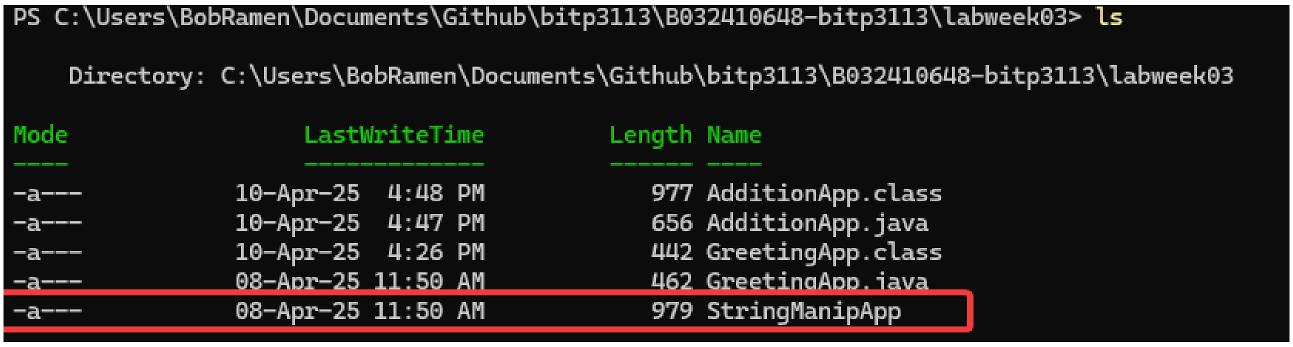
Figure 17: The `main( )` method definition

Exercise 11: Executing a Text Processing App

1. Download **StringManip** file from ulearn.
2. Move the file into folder named **labweek03**.
3. Compile the file.
4. **Using the knowledge comprehended until this point, fix the errors produced from the file.**

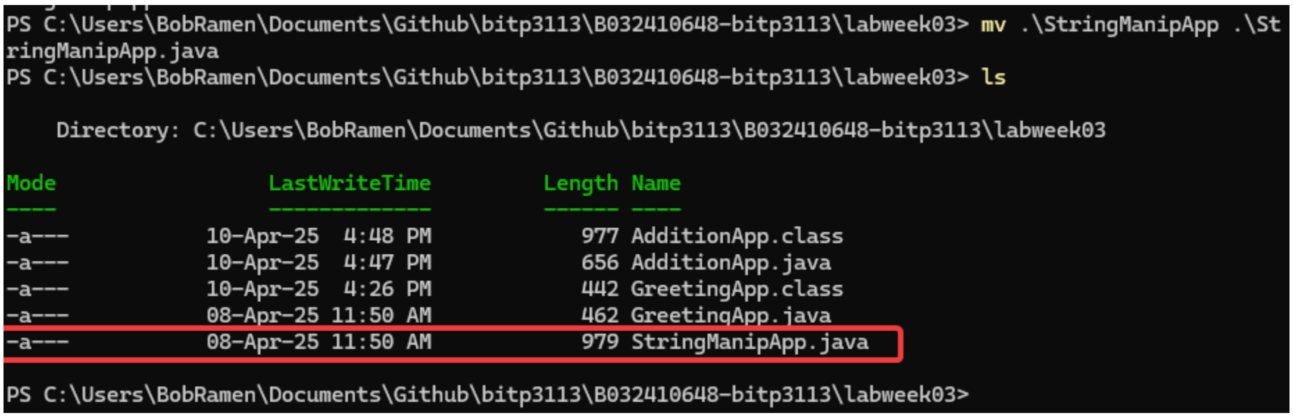
The original file is not saved in .java, so we First, the file needed to be renamed to change the extension.

Before



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          10-Apr-25   4:48 PM            977 AdditionApp.class
-a---          10-Apr-25   4:47 PM            656 AdditionApp.java
-a---          10-Apr-25   4:26 PM            442 GreetingApp.class
-a---          08-Apr-25  11:50 AM            462 GreetingApp.java
-a---          08-Apr-25  11:50 AM            979 StringManipApp
```

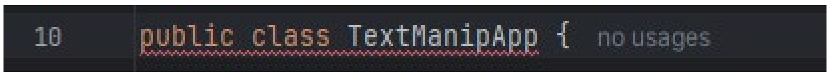
After



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> mv .\StringManipApp .\St
ringManipApp.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          10-Apr-25   4:48 PM            977 AdditionApp.class
-a---          10-Apr-25   4:47 PM            656 AdditionApp.java
-a---          10-Apr-25   4:26 PM            442 GreetingApp.class
-a---          08-Apr-25  11:50 AM            462 GreetingApp.java
-a---          08-Apr-25  11:50 AM            979 StringManipApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

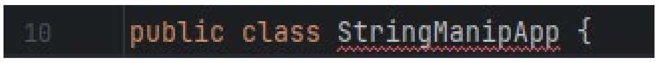
Next, we need to change the class name to be the same as the file name, in this case the file name is StringManipApp but the class name is defined as TextManipApp

Before



```
public class TextManipApp {  no usages
```

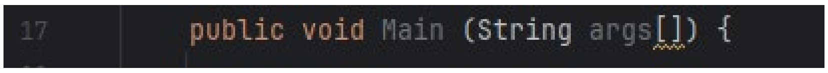
After



```
public class StringManipApp {
```

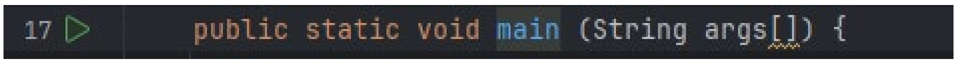
Lastly, the main method is missing a keyword which is static, we need to change that, and the "Main" keyword is case sensitive so change it to "main"

Before



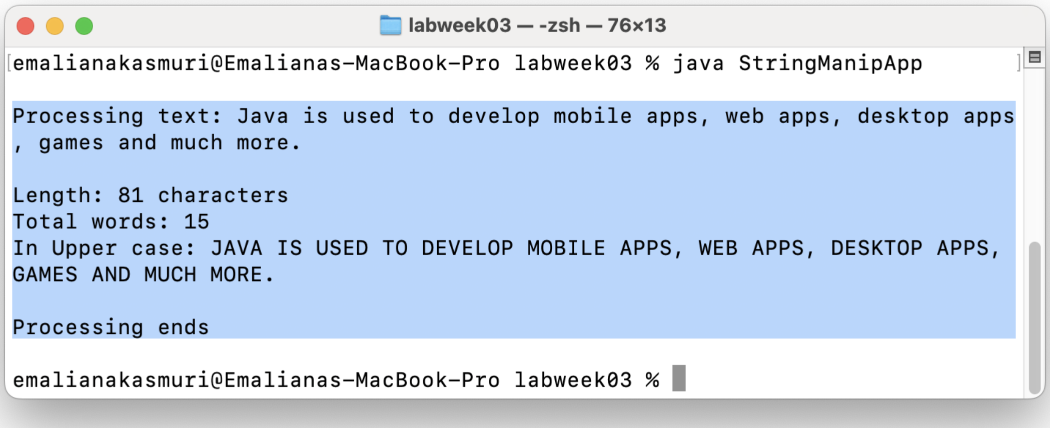
```
    public void Main (String args[]) {
```

After



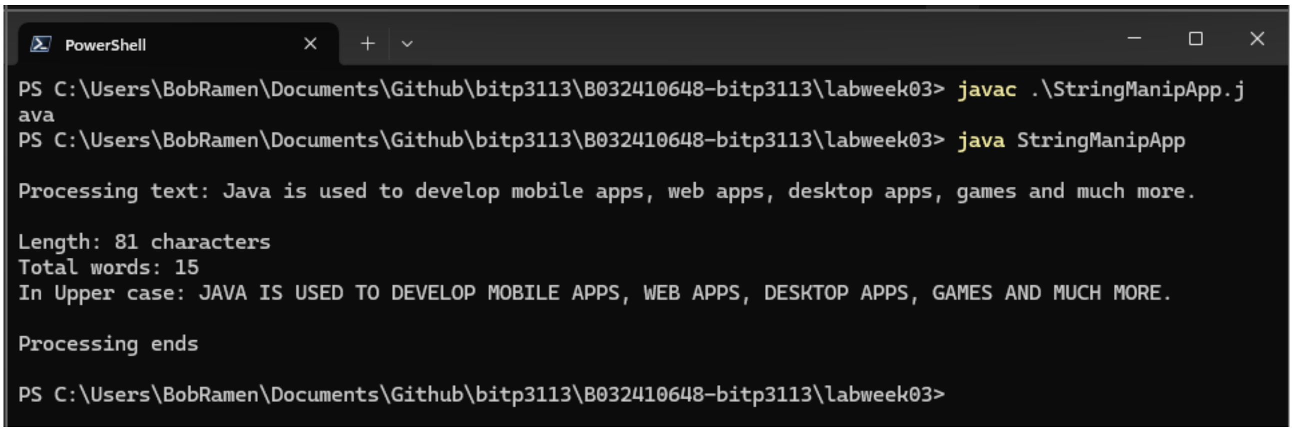
```
    public static void main (String args[]) {
```

5. Execute the application only when the compilation is free from any error. The application shall produce an output similar as shown in Figure 18.



```
emalianakasmuri@Emalianas-MacBook-Pro labweek03 % java StringManipApp

Processing text: Java is used to develop mobile apps, web apps, desktop apps
, games and much more.

Length: 81 characters
Total words: 15
In Upper case: JAVA IS USED TO DEVELOP MOBILE APPS, WEB APPS, DESKTOP APPS,
GAMES AND MUCH MORE.

Processing ends

emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

Figure 18: Output from StringManipApp execution


PowerShell

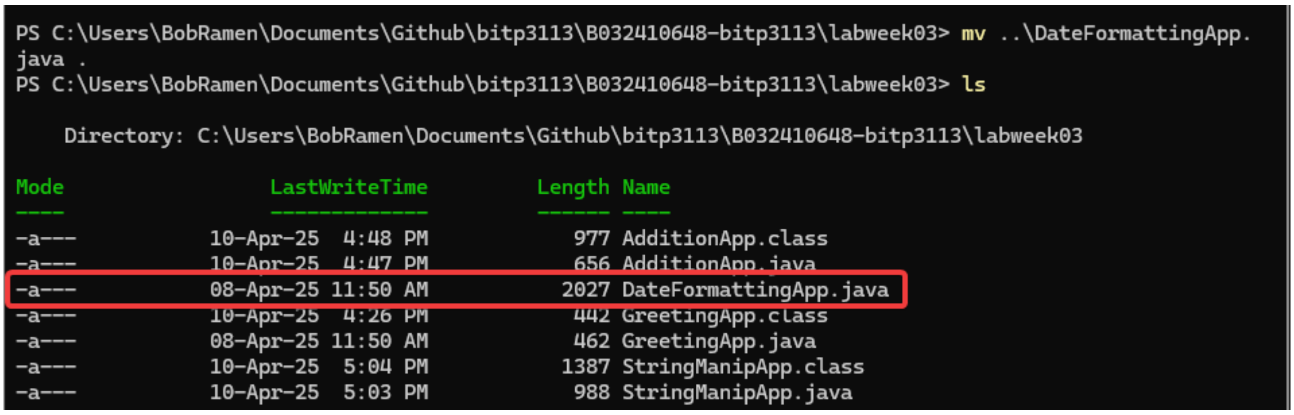
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\StringManipApp.j
ava
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java StringManipApp

Processing text: Java is used to develop mobile apps, web apps, desktop apps, games and much more.

Length: 81 characters
Total words: 15
In Upper case: JAVA IS USED TO DEVELOP MOBILE APPS, WEB APPS, DESKTOP APPS, GAMES AND MUCH MORE.

Processing ends

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

Java Executable Statement

Each Java executable statement must be terminated with a semi-colon `;`. Most Java statements are written within the class or method block. A block is marked with curly brackets.

Exercise 12: Executing a Date Manipulation App

1. Download **DateFormattingApp.java** file from ulearn.
2. Move the file into folder named **labweek03**.

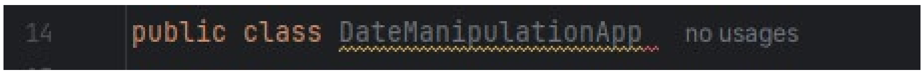


```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> mv ..\DateFormattingApp.
java .
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          10-Apr-25   4:48 PM            977 AdditionApp.class
-a---          10-Apr-25   4:47 PM            656 AdditionApp.java
-a---          08-Apr-25  11:50 AM           2027 DateFormattingApp.java
-a---          10-Apr-25   4:26 PM            442 GreetingApp.class
-a---          08-Apr-25  11:50 AM            462 GreetingApp.java
-a---          10-Apr-25   5:04 PM           1387 StringManipApp.class
-a---          10-Apr-25   5:03 PM            988 StringManipApp.java
```

3. Compile the file.
4. **Using the knowledge comprehended until this point, fix the errors produced from the file.**

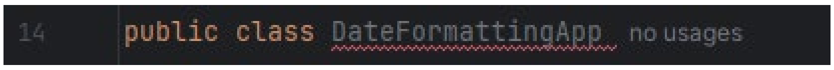
First, we need to change the class name to the file name from DateManipulationApp to DateFormattingApp

Before



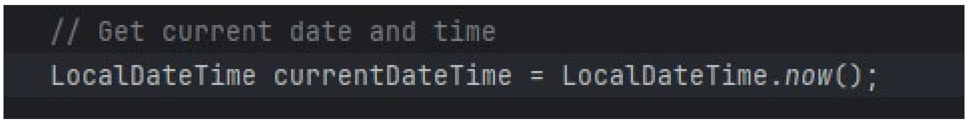
```
public class DateManipulationApp  no usages
```

After

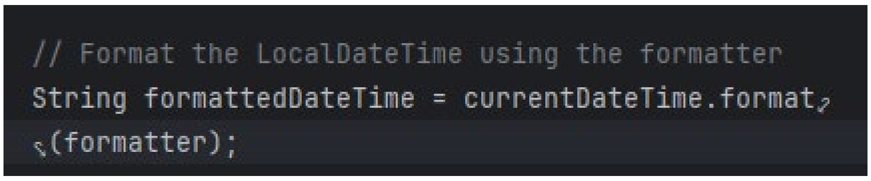


```
public class DateFormattingApp  no usages
```

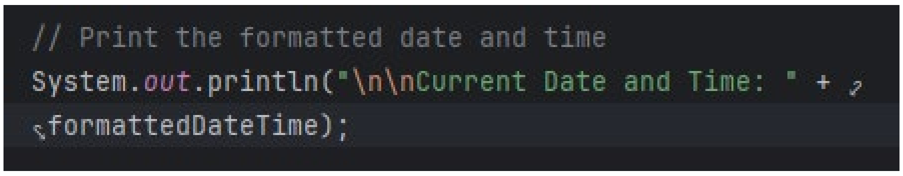
Fix all the semicolon



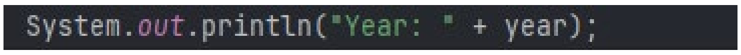
```
// Get current date and time
LocalDateTime currentDateTime = LocalDateTime.now();
```



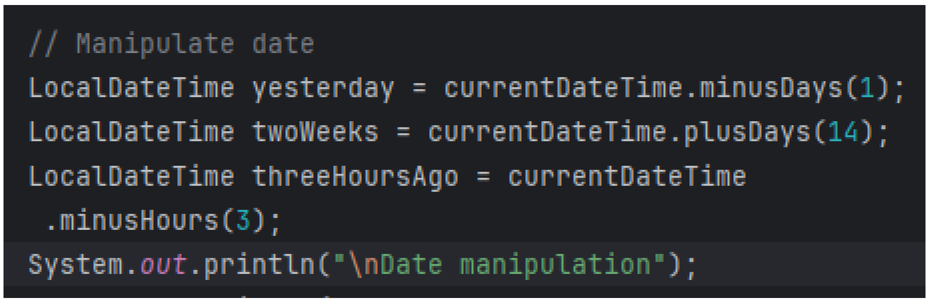
```
// Format the LocalDateTime using the formatter
String formattedDateTime = currentDateTime.format
(formatter);
```



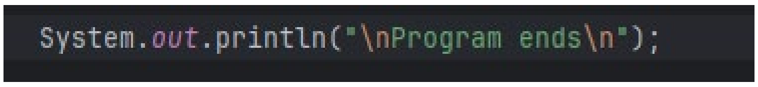
```
// Print the formatted date and time
System.out.println("\n\nCurrent Date and Time: " + 
formattedDateTime);
```



```java
System.out.println("Year: " + year);
```

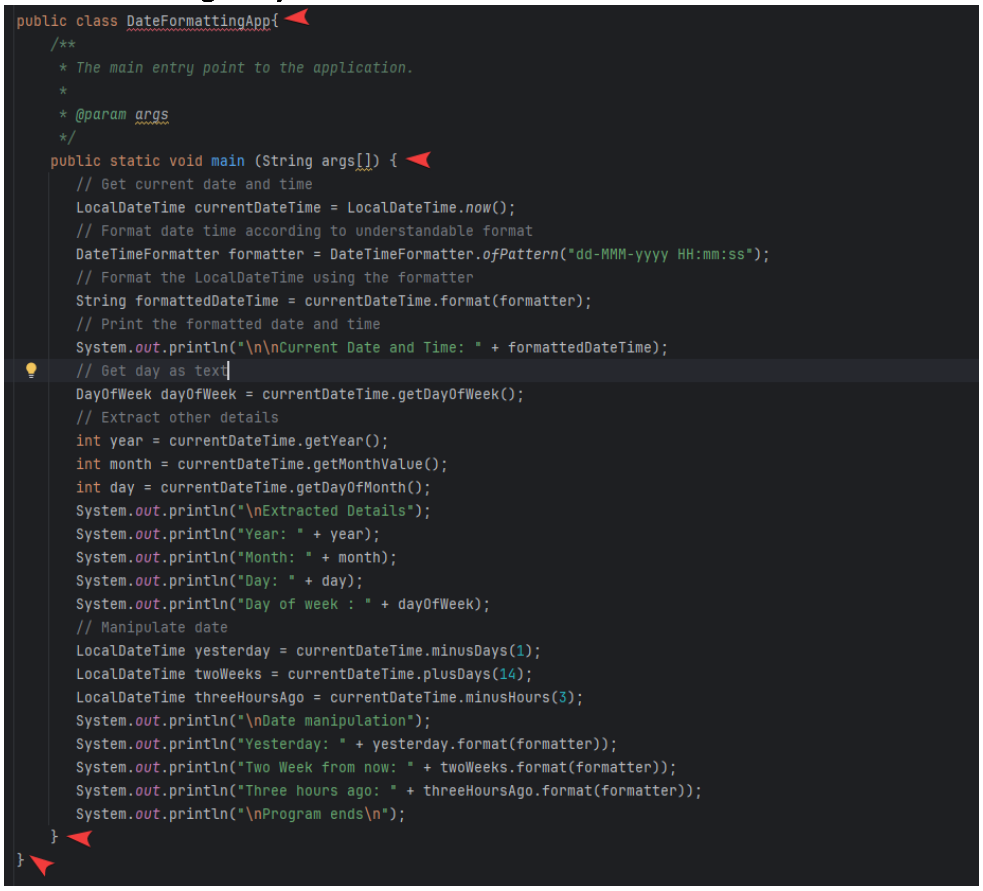


```java
// Manipulate date
LocalDateTime yesterday = currentDateTime.minusDays(1);
LocalDateTime twoWeeks = currentDateTime.plusDays(14);
LocalDateTime threeHoursAgo = currentDateTime
 .minusHours(3);
System.out.println("\nDate manipulation");
```



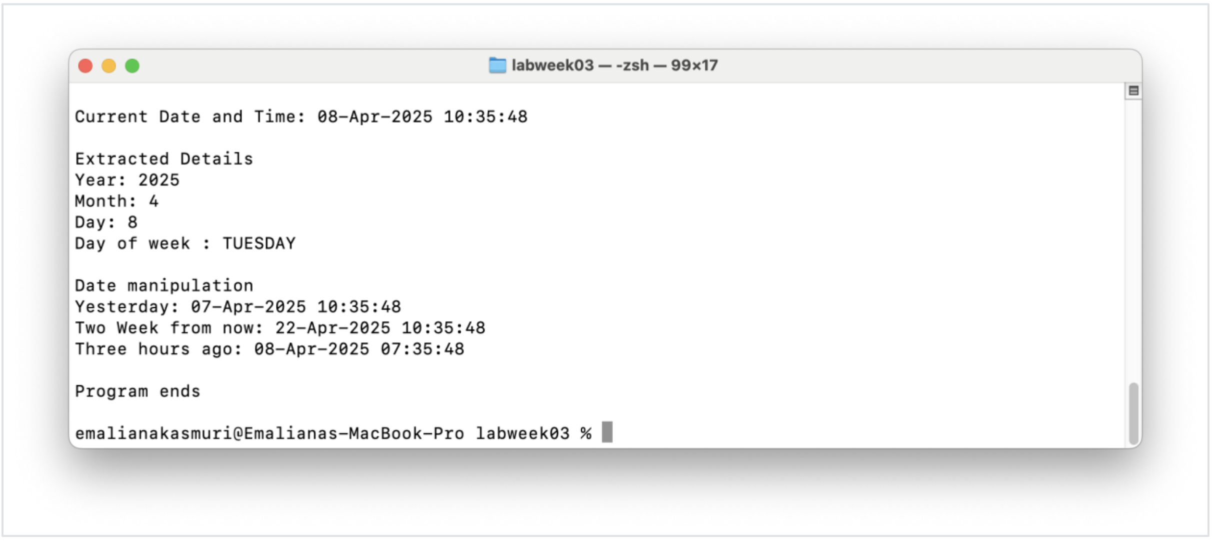
```java
System.out.println("\nProgram ends\n");
```

Add the missing curly bracket



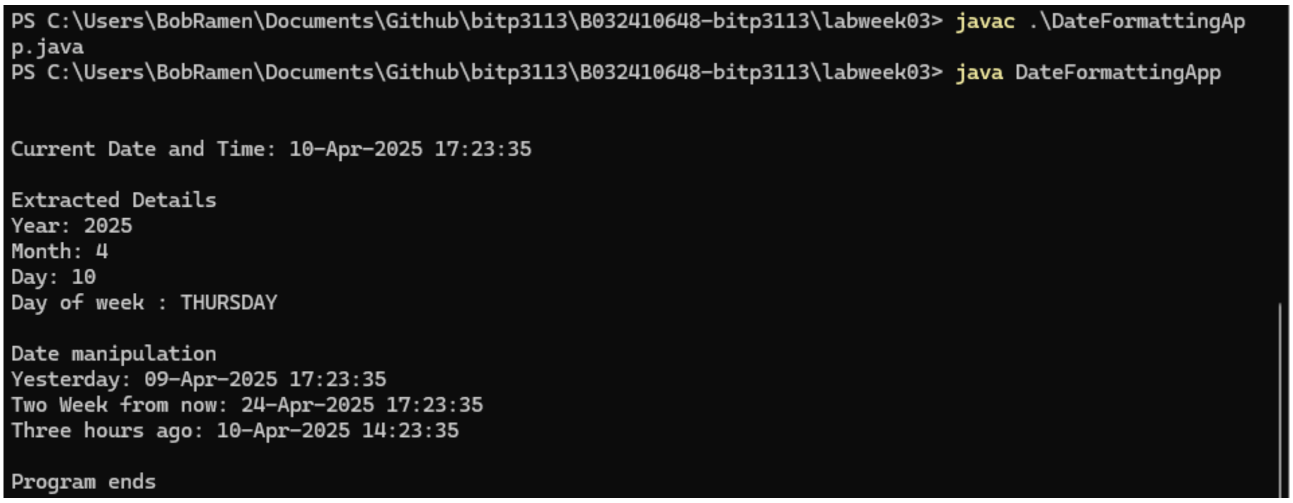
```java
public class DateFormattingApp{
    /**
     * The main entry point to the application.
     *
     * @param args
     */
    public static void main (String args[]) {
        // Get current date and time
        LocalDateTime currentDateTime = LocalDateTime.now();
        // Format date time according to understandable format
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MMM-yyyy HH:mm:ss");
        // Format the LocalDateTime using the formatter
        String formattedDateTime = currentDateTime.format(formatter);
        // Print the formatted date and time
        System.out.println("\n\nCurrent Date and Time: " + formattedDateTime);
        // Get day as text
        DayOfWeek dayOfWeek = currentDateTime.getDayOfWeek();
        // Extract other details
        int year = currentDateTime.getYear();
        int month = currentDateTime.getMonthValue();
        int day = currentDateTime.getDayOfMonth();
        System.out.println("\nExtracted Details");
        System.out.println("Year: " + year);
        System.out.println("Month: " + month);
        System.out.println("Day: " + day);
        System.out.println("Day of week : " + dayOfWeek);
        // Manipulate date
        LocalDateTime yesterday = currentDateTime.minusDays(1);
        LocalDateTime twoWeeks = currentDateTime.plusDays(14);
        LocalDateTime threeHoursAgo = currentDateTime.minusHours(3);
        System.out.println("\nDate manipulation");
        System.out.println("Yesterday: " + yesterday.format(formatter));
        System.out.println("Two Week from now: " + twoWeeks.format(formatter));
        System.out.println("Three hours ago: " + threeHoursAgo.format(formatter));
        System.out.println("\nProgram ends\n");
    }
}
```

5. Execute the application only when the compilation is free from any error. The application shall produce an output similar as shown in Figure 19.



```
Current Date and Time: 08-Apr-2025 10:35:48

Extracted Details
Year: 2025
Month: 4
Day: 8
Day of week : TUESDAY

Date manipulation
Yesterday: 07-Apr-2025 10:35:48
Two Week from now: 22-Apr-2025 10:35:48
Three hours ago: 08-Apr-2025 07:35:48

Program ends

emalianakasmuri@Emalianas-MacBook-Pro labweek03 %
```

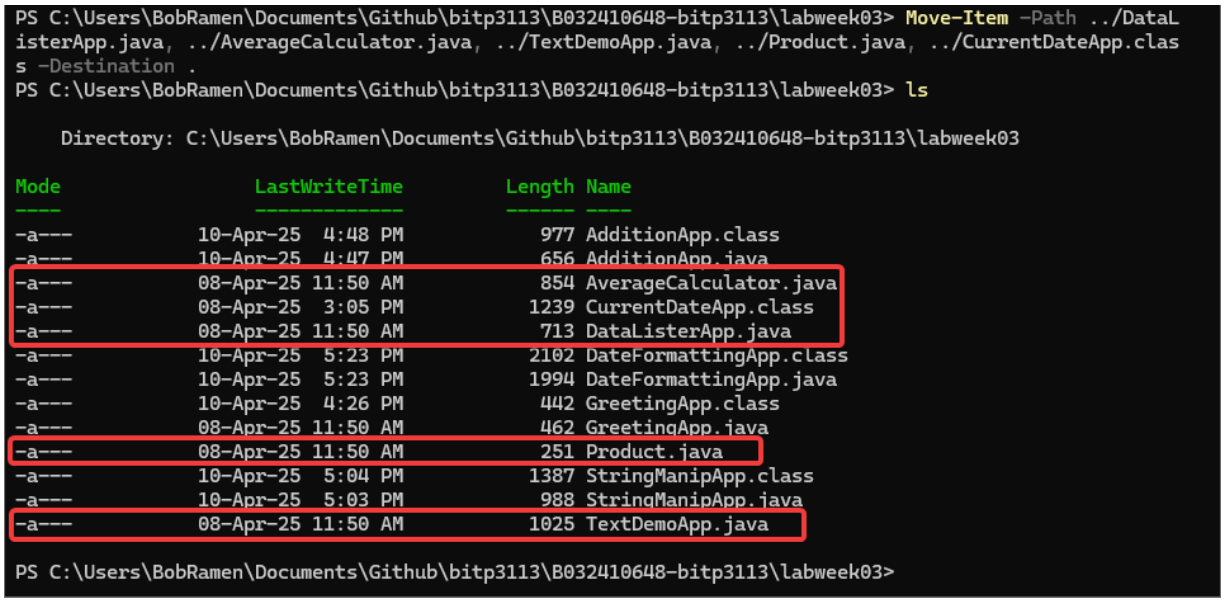
Figure 19: Output from DateManipulationApp execution



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\DateFormattingAp
p.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java DateFormattingApp


Current Date and Time: 10-Apr-2025 17:23:35

Extracted Details
Year: 2025
Month: 4
Day: 10
Day of week : THURSDAY

Date manipulation
Yesterday: 09-Apr-2025 17:23:35
Two Week from now: 24-Apr-2025 17:23:35
Three hours ago: 10-Apr-2025 14:23:35

Program ends
```

Exercise 13: Executing Other Applications

1. Download other the following files from ulearn.
   a. DataListerApp.java
   b. AverageCalculator.java
   c. TextDemoApp.java
   d. Product.java
   e. CurrentDateApp.class
2. Move the file into folder named **labweek03**.



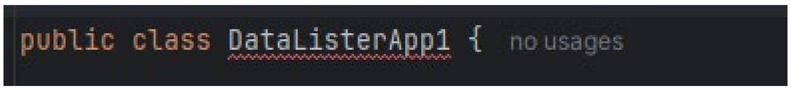
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> Move-Item -Path ../DataL
isterApp.java, ../AverageCalculator.java, ../TextDemoApp.java, ../Product.java, ../CurrentDateApp.clas
s -Destination .
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> ls

    Directory: C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          10-Apr-25   4:48 PM            977 AdditionApp.class
-a---          10-Apr-25   4:47 PM            656 AdditionApp.java
-a---          08-Apr-25  11:50 AM            854 AverageCalculator.java
-a---          08-Apr-25   3:05 PM           1239 CurrentDateApp.class
-a---          08-Apr-25  11:50 AM            713 DataListerApp.java
-a---          10-Apr-25   5:23 PM           2102 DateFormattingApp.class
-a---          10-Apr-25   5:23 PM           1994 DateFormattingApp.java
-a---          10-Apr-25   4:26 PM            442 GreetingApp.class
-a---          08-Apr-25  11:50 AM            462 GreetingApp.java
-a---          08-Apr-25  11:50 AM            251 Product.java
-a---          10-Apr-25   5:04 PM           1387 StringManipApp.class
-a---          10-Apr-25   5:03 PM            988 StringManipApp.java
-a---          08-Apr-25  11:50 AM           1025 TextDemoApp.java

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

3. **Using the comprehended knowledge, compile the files.**
4. **Fix any errors produced from the compilation.**

DataListerApp.java

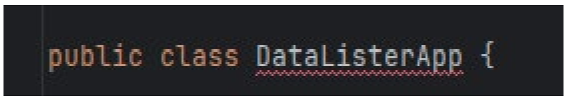
Class name is not the same as filename.

Before



```
public class DataListerApp1 {  no usages
```

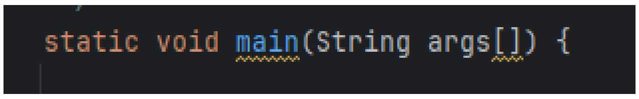
After



```
public class DataListerApp {
```

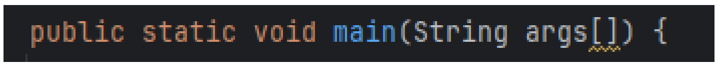
Main method is written incorrectly

Before



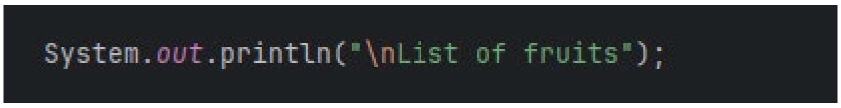
```
static void main(String args[]) {
```

After

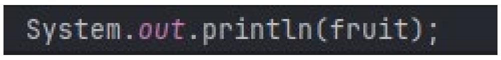


```
public static void main(String args[]) {
```

Semicolon is not added



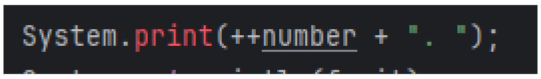
```
System.out.println("\nList of fruits");
```



```
System.out.println(fruit);
```

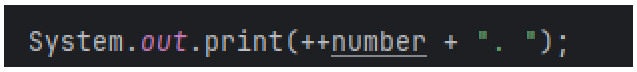
Method to display text is written incorrectly

Before



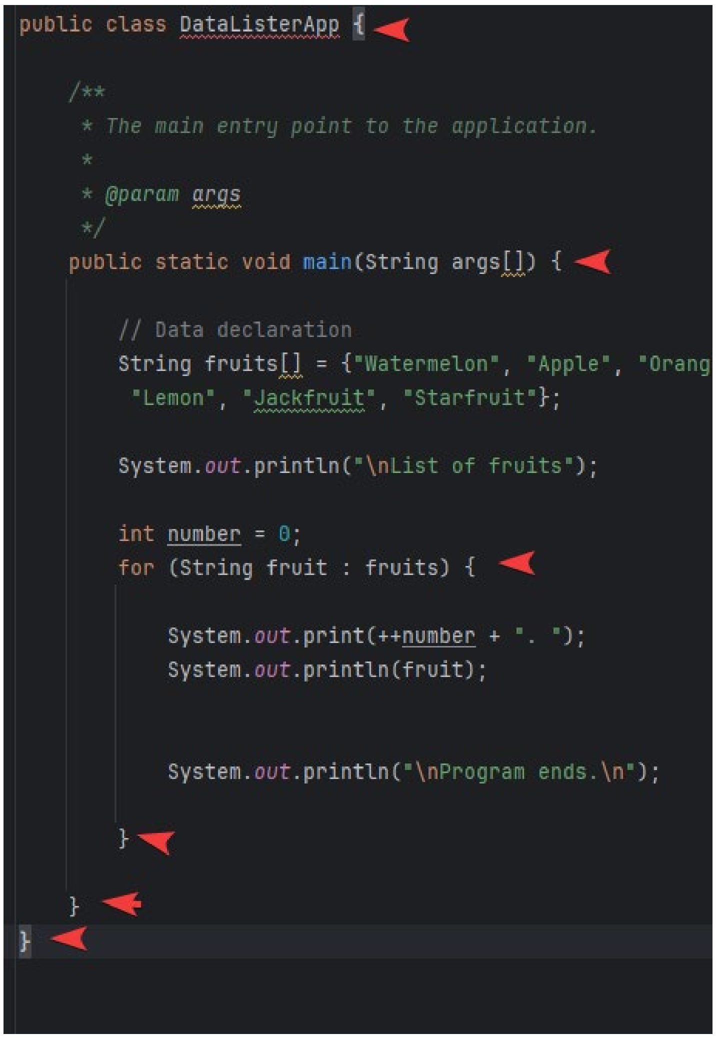
```
System.print(++number + ". ");
```

After



```
System.out.print(++number + ". ");
```

Add all missing brackets

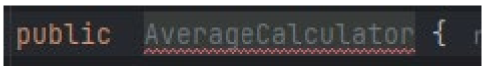


```
public class DataListerApp {

    /**
     * The main entry point to the application.
     *
     * @param args
     */
    public static void main(String args[]) {

        // Data declaration
        String fruits[] = {"Watermelon", "Apple", "Orang
         "Lemon", "Jackfruit", "Starfruit"};

        System.out.println("\nList of fruits");

        int number = 0;
        for (String fruit : fruits) {

            System.out.print(++number + ". ");
            System.out.println(fruit);


            System.out.println("\nProgram ends.\n");

        }

    }
}
```

AverageCalculator.java

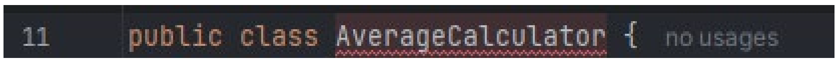
Class is missing

Before



```
public  AverageCalculator {
```

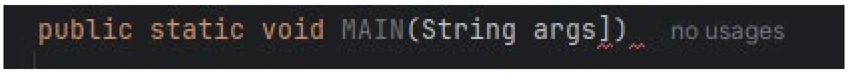
After



```
public class AverageCalculator {  no usages
```

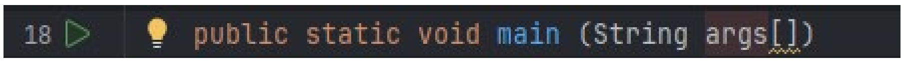
Main method is written incorrectly

Before



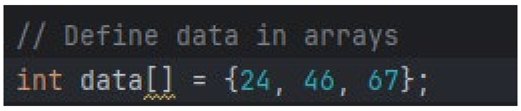
```
public static void MAIN(String args])  no usages
```

After

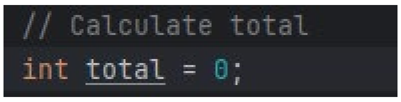


```
public static void main (String args[])
```

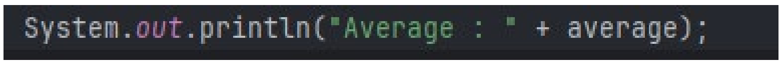
Add all semicolon



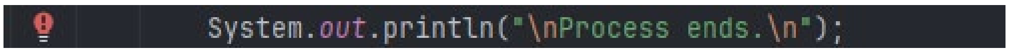
```
// Define data in arrays
int data[] = {24, 46, 67};
```



```
// Calculate total
int total = 0;
```



```
System.out.println("Average : " + average);
```

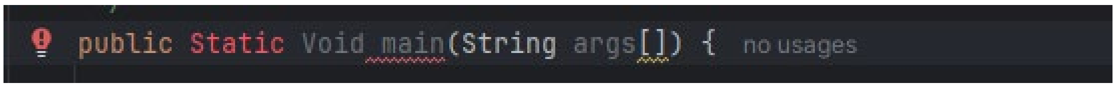


```
        System.out.println("\nProcess ends.\n");
```

TextDemoApp.java

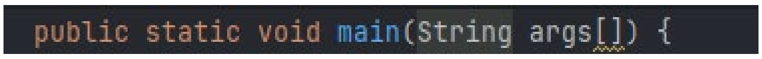
Main method is written incorrectly and there are 2 main method, there can be only one main method

Before



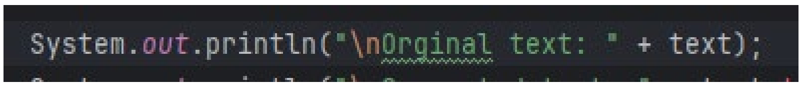
```
public Static Void main(String args[]) {  no usages
```

After

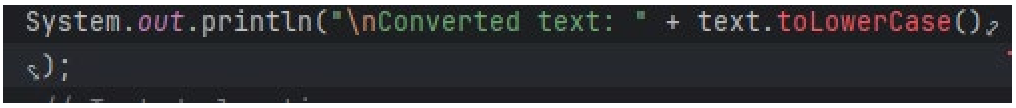


```
public static void main(String args[]) {
```

Insert all missing Semicolon



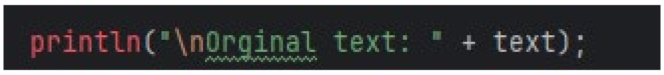
```
System.out.println("\nOrginal text: " + text);
```



```
System.out.println("\nConverted text: " + text.toLowerCase()
);
```

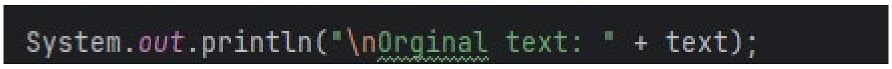
Method to display text is wrongly written

Before



```
println("\nOrginal text: " + text);
```

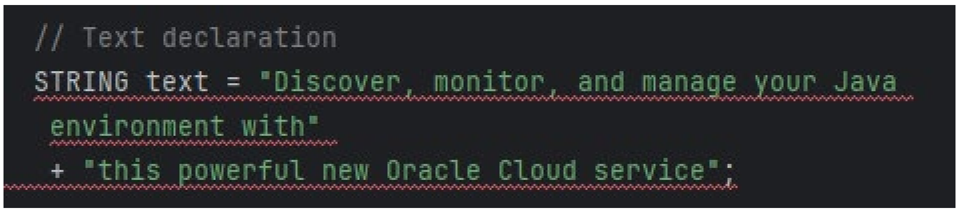
After



```
System.out.println("\nOrginal text: " + text);
```

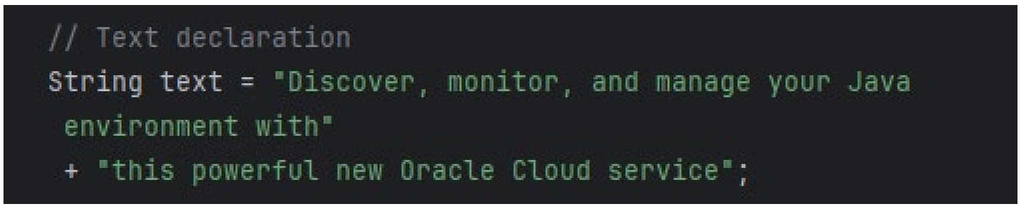
STRING is an incorrect keyword, it is case sensitive

Before



```
// Text declaration
STRING text = "Discover, monitor, and manage your Java
 environment with"
 + "this powerful new Oracle Cloud service";
```

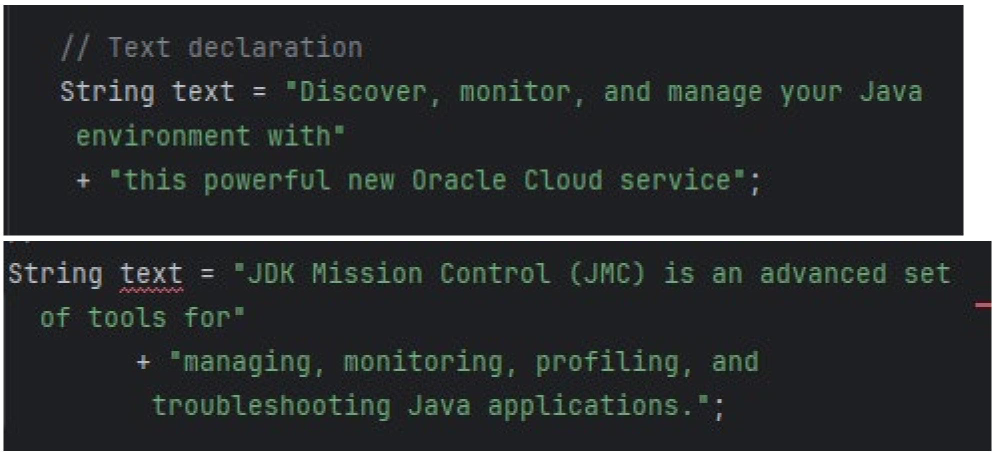
After



```
// Text declaration
String text = "Discover, monitor, and manage your Java
 environment with"
 + "this powerful new Oracle Cloud service";
```

2 variables with the same name are defined, change to other variables

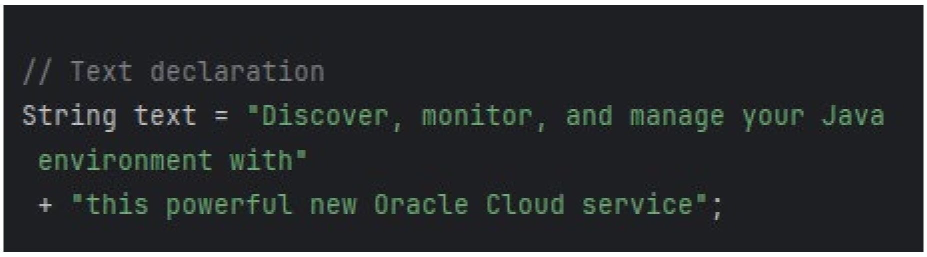
Before



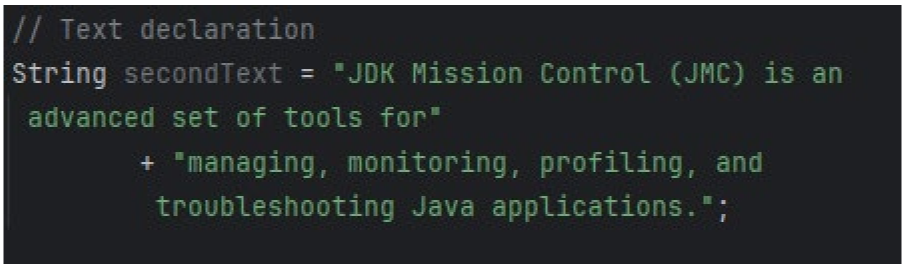
```
// Text declaration
String text = "Discover, monitor, and manage your Java
 environment with"
 + "this powerful new Oracle Cloud service";
```

```
String text = "JDK Mission Control (JMC) is an advanced set
  of tools for"
        + "managing, monitoring, profiling, and
         troubleshooting Java applications.";
```

After



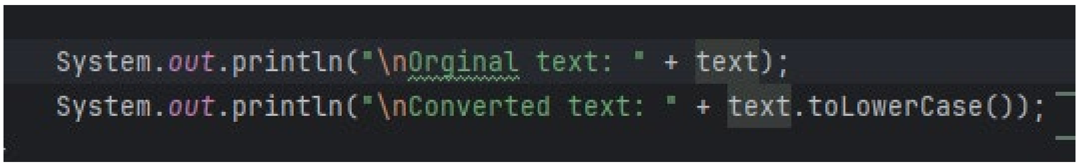
```
// Text declaration
String text = "Discover, monitor, and manage your Java
 environment with"
 + "this powerful new Oracle Cloud service";
```



```
// Text declaration
String secondText = "JDK Mission Control (JMC) is an
 advanced set of tools for"
        + "managing, monitoring, profiling, and
         troubleshooting Java applications.";
```

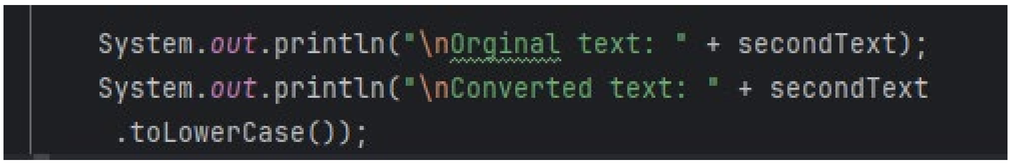
The second print statement is using the wrong variable

Before



```
System.out.println("\nOrginal text: " + text);
System.out.println("\nConverted text: " + text.toLowerCase());
```

After

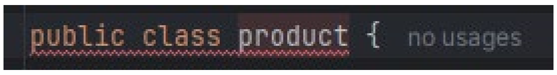


```
System.out.println("\nOrginal text: " + secondText);
System.out.println("\nConverted text: " + secondText
 .toLowerCase());
```

Product.java

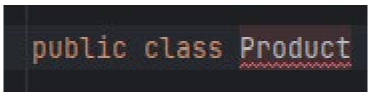
Class name is not the same as the file name

Before



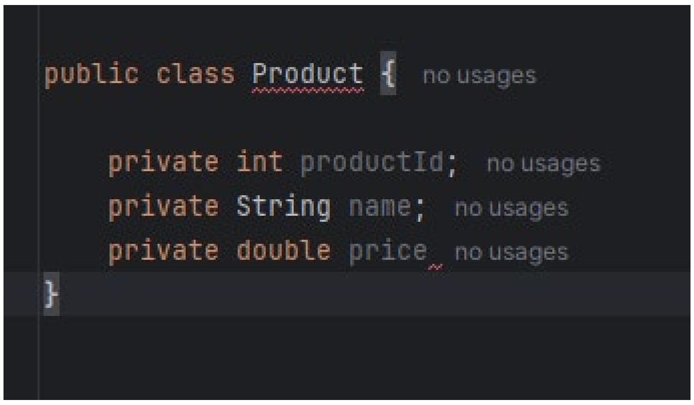
```
public class product {  no usages
```

After



```
public class Product
```

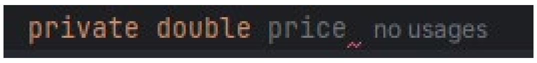
Curly bracket does not tally



```
public class Product {  no usages

    private int productId;  no usages
    private String name;  no usages
    private double price  no usages
}
```

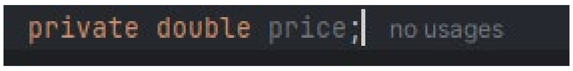
Add missing semicolon

Before



```
private double price  no usages
```

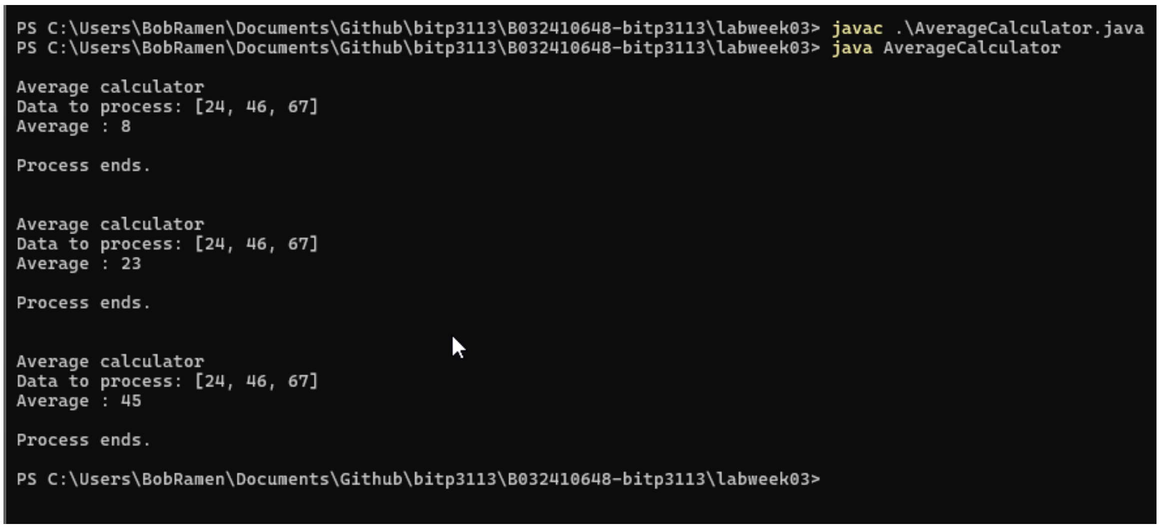
After



```
private double price;  no usages
```

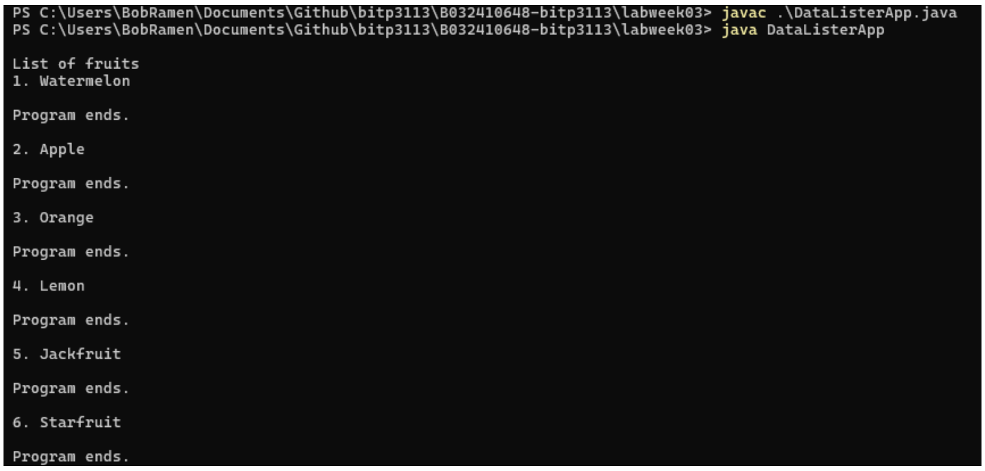
5. Execute the application.

AverageCalculator.java



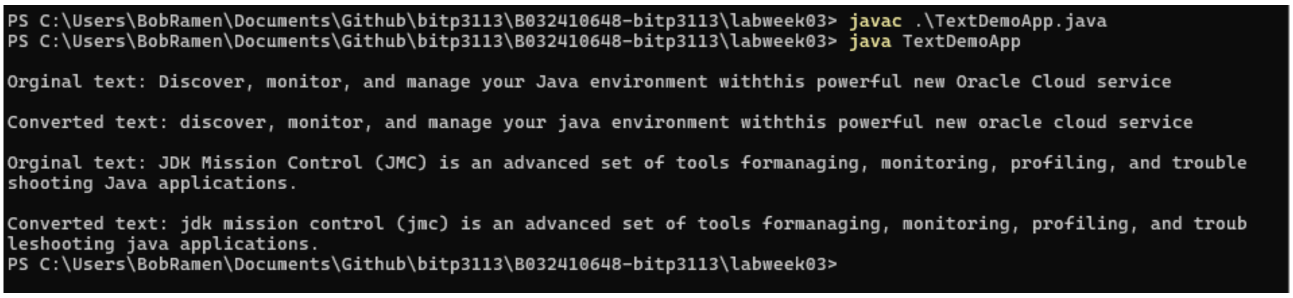
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\AverageCalculator.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java AverageCalculator

Average calculator
Data to process: [24, 46, 67]
Average : 8

Process ends.


Average calculator
Data to process: [24, 46, 67]
Average : 23

Process ends.


Average calculator
Data to process: [24, 46, 67]
Average : 45

Process ends.

PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

DataListerApp.java



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\DataListerApp.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java DataListerApp

List of fruits
1. Watermelon

Program ends.

2. Apple

Program ends.

3. Orange

Program ends.

4. Lemon

Program ends.

5. Jackfruit

Program ends.

6. Starfruit

Program ends.
```

TextDemoApp.java



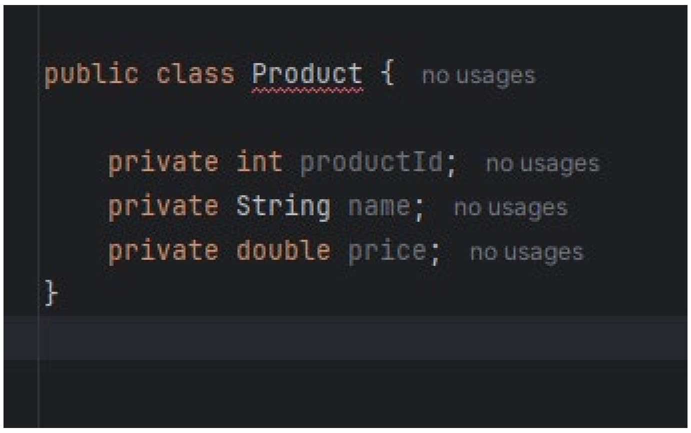
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\TextDemoApp.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java TextDemoApp

Orginal text: Discover, monitor, and manage your Java environment withthis powerful new Oracle Cloud service

Converted text: discover, monitor, and manage your java environment withthis powerful new oracle cloud service

Orginal text: JDK Mission Control (JMC) is an advanced set of tools formanaging, monitoring, profiling, and trouble
shooting Java applications.

Converted text: jdk mission control (jmc) is an advanced set of tools formanaging, monitoring, profiling, and troub
leshooting java applications.
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

6. **One of the files is not executable even though the file passed compilation phase. Using the comprehended knowledge, record your analysis of this failure in ulearn.**

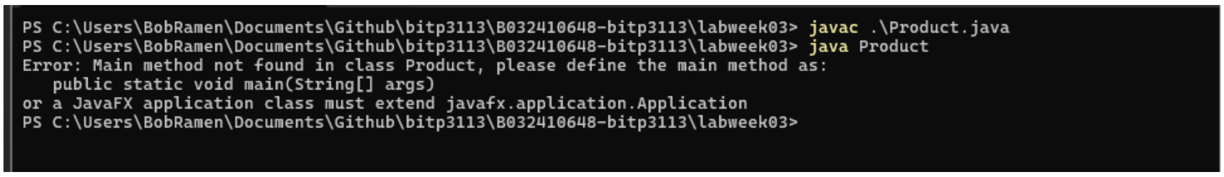
   The file in question is Product.java

   Reason:

   - Java didn't require a main method in each class to pass compilation
   - If the syntax is correct, it can be compiled even though the code is technically not runnable/usable.



```
public class Product {  no usages

    private int productId;  no usages
    private String name;  no usages
    private double price;  no usages
}
```



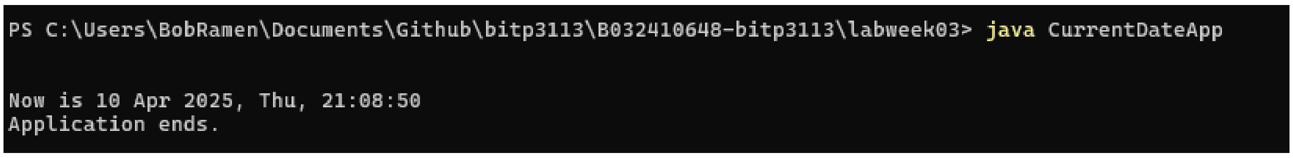
```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> javac .\Product.java
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java Product
Error: Main method not found in class Product, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03>
```

7. **One of the files is executable even without .java. Using the comprehended knowledge, record your analysis on this success in ulearn.**

   The file in question is CurrentDateApp.class

   Reason:

   - The .class file means that the code has already been compiled and has been converted to machine code so any device that can run Java can run it without any problem.



```
PS C:\Users\BobRamen\Documents\Github\bitp3113\B032410648-bitp3113\labweek03> java CurrentDateApp


Now is 10 Apr 2025, Thu, 21:08:50
Application ends.
```