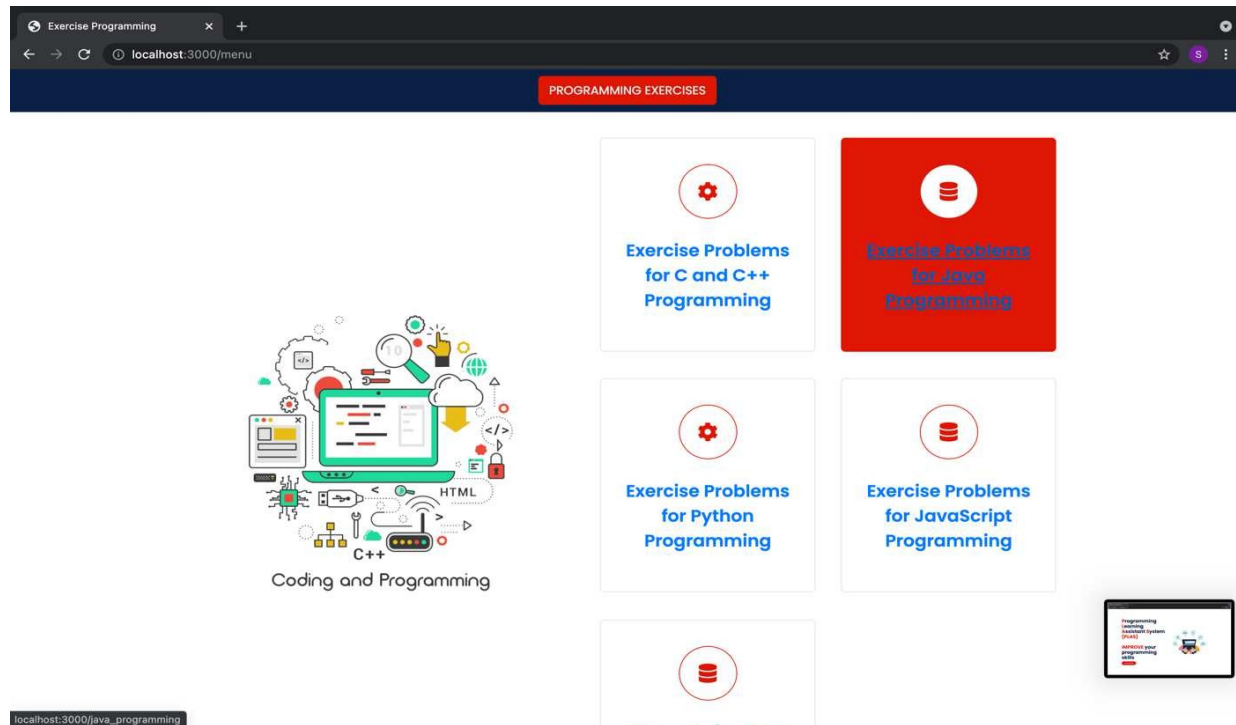


# Code Writing Problem (CWP)

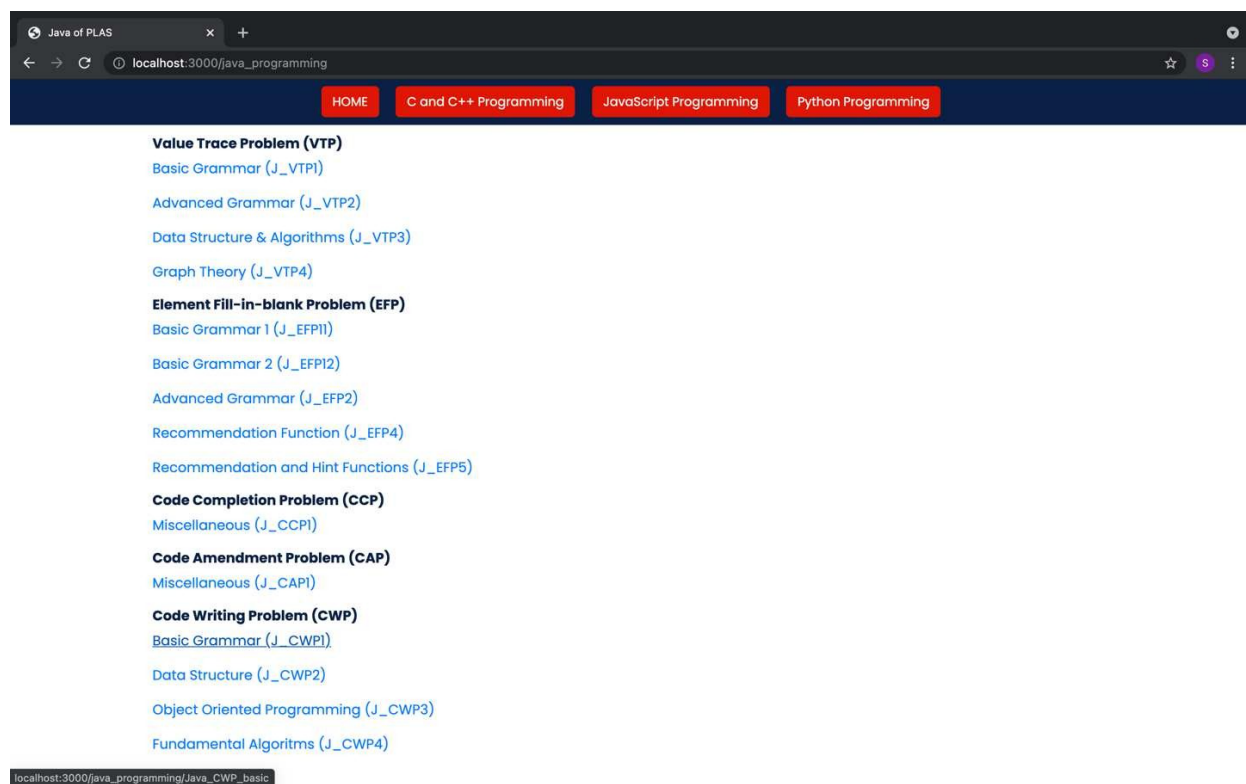
1. Choose Java programming.



2. To solve CWP problems, start from “Basic Grammar(J\_CWP1)”.

## About CWP

In this problem type, CWP, we gave the test code and source code for each problem. You need to write the source code by reading the test code.



3. Solve the problems by clicking each problem. To go back home page, click “**Top**” button.

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/java\_programming/Java\_CWP\_basic'. The page title is 'Offline JPLAS'. Below the title are two buttons: 'TOP' (green) and 'End' (blue). On the left, there is a sidebar menu with 'Problems' selected. The main content area is titled 'Problems' and contains a table with 11 rows. The first row is highlighted in blue.

| Problem No | ProblemName     | Remark |
|------------|-----------------|--------|
| 1          | HelloWorld      |        |
| 2          | MessageDisplay  |        |
| 3          | CodeCorrection1 |        |
| 4          | CodeCorrection2 |        |
| 5          | IfAndSwitch     |        |
| 6          | EscapeUsage     |        |
| 7          | ReturnAndBreak  |        |
| 8          | OctalNumber     |        |
| 9          | Hexadecimal     |        |
| 10         | MaxItem         |        |
| 11         | MinItem         |        |

Below the table is a 'Submission' section with a small icon.

4. Write the answer code under source code by reading the test code on the right side. Then, click “**Submit**” button to check whether your answer is correct or not. To continue another problem, click “**Next**” button. To go back all basic problems page, click the title “**CWP Basic**”.

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/java\_programming/Java\_CWP\_basic/p1'. The page title is 'CWP Basic'. Below the title are two buttons: 'Next' (red) and 'Submit' (blue). On the left, there is a sidebar menu with 'Problems' selected. The main content area is titled 'CWP Basic' and contains two code editors: 'Source Code' and 'Test Code'. The 'Source Code' editor shows the following code:

```
1 package p1;
2 public class HelloWorld{
3
4 }
```

The 'Test Code' editor shows the following code:

```
1 package p1;
2 import static org.junit.Assert.assertEquals;
3 import java.io.ByteArrayOutputStream;
4 import java.io.PrintStream;
5 import org.junit.Test;
6
7 import report.Report;
8
9 public class HelloWorldTest {
10
11     @Test
12     public void testMain() {
13         ByteArrayOutputStream bos = new ByteArrayOutputStream();
14         PrintStream bout = new PrintStream(bos);
15         System.setOut(bout);
16         HelloWorld.main(null);
17         String[] outs = bos.toString().split(System.lineSeparator);
18         //System.out.println(outs.length);
19         assertEquals("Hello World!", outs[outs.length-1]);
20     }
21
22     @Test
23     public void jplasStocker() {
24         Report.record(this.getClass().getName());
25     }
26 }
```

5. If your answer is incorrect, you can modify your source code again by reading the error message in the box. Then, you can submit again until the test is successful.

The screenshot shows a web browser window with a URL bar indicating a local development environment. The main area is split into two panes. The left pane contains the following Java code:

```
4 private String name;
5
6 public void setName(String name) {
7     this.name = name;
8 }
9
10 }
```

The right pane contains the following Java code:

```
9 public class HelloWorldTest {
10
11     @Test
12     public void testMain() {
13         ByteArrayOutputStream bos = new ByteArrayOutputStream()
14         PrintStream bout = new PrintStream(bos);
15         System.setOut(bout);
16         HelloWorld.main(null);
17         String[] outs = bos.toString().split(System.lineSeparator);
18         //System.out.println(outs.length);
19         assertEquals("Hello World!", outs[outs.length-1]);
20     }
21
22     @Test
23     public void jplasStocker() {
24         Report.record(this.getClass().getName());
25     }
26 }
27
28
29
30
```

Below the code panes is a dark grey box with a red dashed border containing a compilation error message:

```
COMPILING ERROR!! p1/HelloWorldTest.java:16: error: cannot find symbol
    HelloWorld.main(null);
    symbol:   method main(<null>)
    location: class HelloWorld
1 error
```

6. You will see this result if your answer is correct.

The screenshot shows the same IDE interface as before, but with different code and a successful test result. The left pane contains:

```
14 }
15 }
```

The right pane contains:

```
19 assertEquals("Hello World!", outs[outs.length-1]);
20
21 }
22
23 @Test
24 public void jplasStocker() {
25     Report.record(this.getClass().getName());
26 }
27
28
29
30
```

Below the code panes is a dark grey box containing the following text:

```
JUnit version 4.3
..
OK (2 tests)
```

7. For the problem you have already solved, and the answer is corrected, “**Finished**” message will show in the Remark. If the answer is not corrected, you will see “**Trying**” message. There is no message for the problem you haven’t solved yet.

The screenshot shows the 'Offline JPLAS' web application running in a browser. The browser address bar shows 'localhost:3000/java\_programming/Java\_CWP\_basic'. The application has a dark header with the title 'Offline JPLAS' and two buttons: 'TOP' (green) and 'End' (blue). Below the header, there is a sidebar menu with links for 'Sidemenu', 'Problems', and 'Submission'. The main content area is titled 'Problems' and contains a table with 11 rows. The first two rows have remarks: 'Finished' (green) and 'Trying' (yellow). Below the table, there is a section titled 'Submission' with a small input field.

| Problem No | ProblemName     | Remark   |
|------------|-----------------|----------|
| 1          | HelloWorld      | Finished |
| 2          | MessageDisplay  | Trying   |
| 3          | CodeCorrection1 |          |
| 4          | CodeCorrection2 |          |
| 5          | IfAndSwitch     |          |
| 6          | EscapeUsage     |          |
| 7          | ReturnAndBreak  |          |
| 8          | OctalNumber     |          |
| 9          | Hexadecimal     |          |
| 10         | MaxItem         |          |
| 11         | MinItem         |          |

Submission

8. Fill your **studentID** or **name** to submit your answer file. Click “**Show Record**” button.

The screenshot shows the 'Submission' section of the 'Offline JPLAS' web application. It features a table with 11 rows, similar to the one in the previous screenshot. Below the table, there is a section titled 'Submission' with a small input field containing the text '43M21510'. Below this input field is a blue button labeled 'Show Record'. Below the button is a large, empty text area labeled 'Your Record'. At the bottom, there is a green button labeled 'Save'.

|    |                 |  |
|----|-----------------|--|
| 3  | CodeCorrection1 |  |
| 4  | CodeCorrection2 |  |
| 5  | IfAndSwitch     |  |
| 6  | EscapeUsage     |  |
| 7  | ReturnAndBreak  |  |
| 8  | OctalNumber     |  |
| 9  | Hexadecimal     |  |
| 10 | MaxItem         |  |
| 11 | MinItem         |  |

Submission

1

43M21510

2

Show Record

Your Record

3

Save

9. Click “**Save**” button when you see your answers in the Record box and click “**Ok**”.

Offline JPLAS x +

localhost:3000/java\_programming/Java\_CWP\_basic

|    |                 |  |
|----|-----------------|--|
| 3  | CodeCorrection1 |  |
| 4  | CodeCorrection2 |  |
| 5  | IfAndSwitch     |  |
| 6  | EscapeUsage     |  |
| 7  | ReturnAndBreak  |  |
| 8  | OctalNumber     |  |
| 9  | Hexadecimal     |  |
| 10 | MaxItem         |  |
| 11 | MinItem         |  |

Submission

1

43M21510

2

Show Record

2021-06-10 3:07:31 PM =====  
2021-06-10 3:07:31 PM 1. Preparing to test topic: Java\_CWP\_basic  
2021-06-10 3:07:31 PM =====

3

Save

Offline JPLAS x +

localhost:3000/java\_programming/Java\_CWP\_basic

|    |                 |  |
|----|-----------------|--|
| 3  | CodeCorrection1 |  |
| 4  | CodeCorrection2 |  |
| 5  | IfAndSwitch     |  |
| 6  | EscapeUsage     |  |
| 7  | ReturnAndBreak  |  |
| 8  | OctalNumber     |  |
| 9  |                 |  |
| 10 |                 |  |
| 11 |                 |  |

Submission

1

43M21510


2

Show Record

2021-06-10 3:07:31 PM =====  
2021-06-10 3:07:31 PM 1. Preparing to test topic: Java\_CWP\_basic  
2021-06-10 3:07:31 PM =====


3

Save

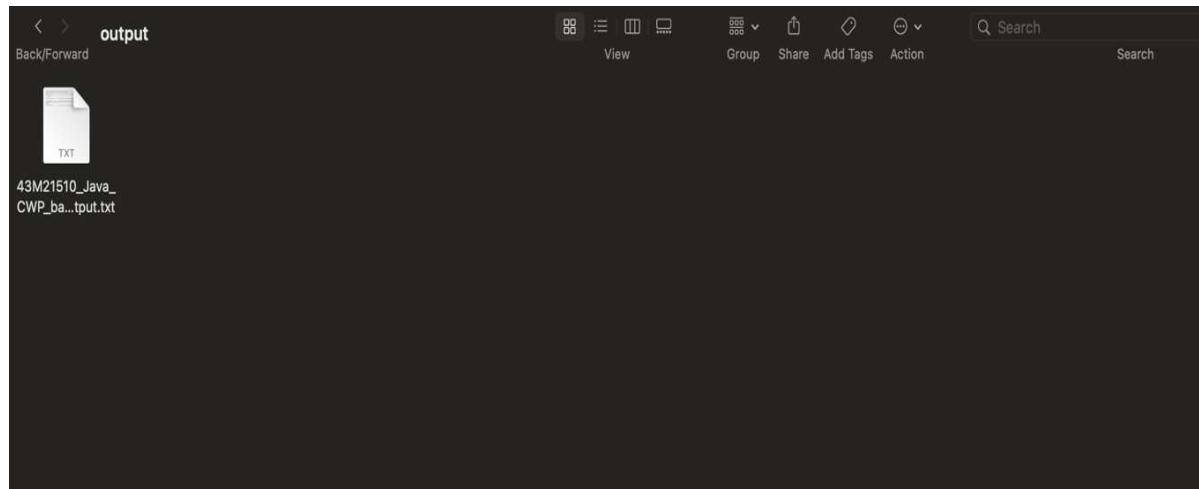


**File saved under output directory.**

OK



10. Your answer file “xxxxxx\_ProblemName\_output.txt” will be saved under your “**output**” directory of your project. Then, send the output file to the teacher via email or pcloud.



```
TextEdit File Edit Format View Window Help
43M21510_Java_CWP_basic_output.txt
2021-06-10 3:07:31 PM -----1.1. Start testing : Java_CWP_basic/p1/HelloWorldTest-----
JUnit version 4.3
...
OK (2 tests)
2021-06-10 3:07:31 PM -----1.1. End testing : Java_CWP_basic/p1/HelloWorldTest-----

2021-06-10 3:07:31 PM -----1.2. Start testing : Java_CWP_basic/p2/MessageDisplayTest-----
COMPILING ERROR!! p2/MessageDisplayTest.java:1: error: class, interface, or enum expected
package p2;
1 error
2021-06-10 3:07:31 PM -----1.2. End testing : Java_CWP_basic/p2/MessageDisplayTest-----
```