

Comp 6481 Labs User Manual

Revisions & authors.....	1
Purpose	2
Login.....	2
Home Page	2
Submitting a lab solution	5
Submission Results	6
Appendix Code examples:	8
Java solution	8
“C” solution.....	9
C++ solution	9
Python Solution.....	9
Appendix: optional reading	10
Clarifications	10
How a submission is being Judged?	10
Output Evaluation	10
Restrictions	10
Contributors.....	11

Revisions & authors

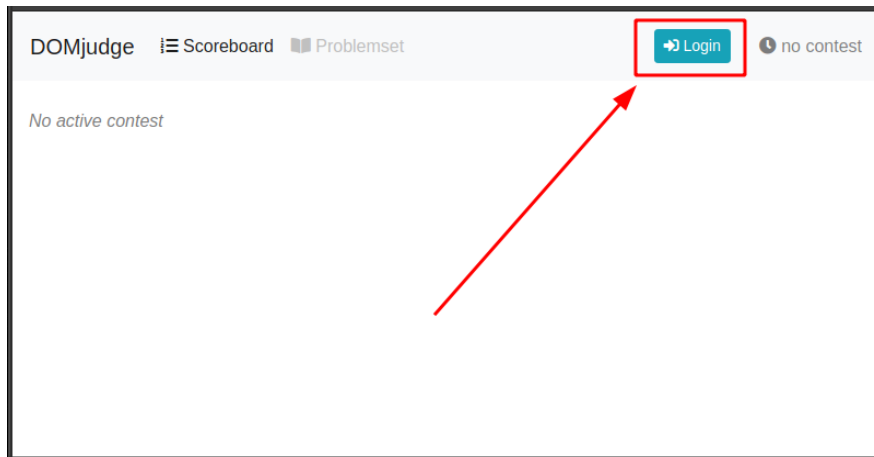
Jul 2020	Alexander Aric	V 2.0	
Jan 2022	Alexander Aric	V 2.3	

Purpose

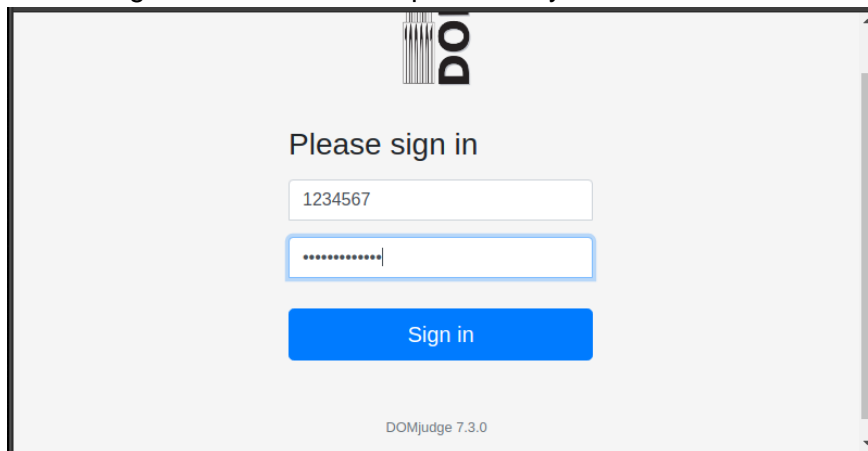
The manual is meant as an introduction to COMP6481 lab's auto judging system GUI. It is based on the [DomJudge](#) (DJ) platform. By reading the manual students should be able to successfully complete their COMP6481 labs. Additional material is listed in section "Optional reading". Full documentation is available on the developers site [here](#). The DJ works entirely through a web interface.

Login

- Start by going to the URL <https://labs6481.encs.concordia.ca>
- and click on "Login" button



- Sign in with credentials provided by tutor or instructor



Upon successful login you will be placed in "Home" page. The section below discusses the main areas of the home page.

Home Page

The top bar menu of the Home page remains visible throughout browsing the entire system. The menus highlighted by red rectangles below are:

- “Home” : default page where user is placed upon successful login
- “Problems”: page where problem description is
- “Scoreboard”: page showing labs status (aka result),
- “Submit” button to be used to submit solutions.
- “Logout” - self-explanatory.
- “Lab0” - the name of the currently active lab.
- Timer - showing how much time left before the active lab expires.

The screenshot shows the DOMjudge web interface. The top navigation bar contains the following elements, each highlighted with a red box:

- DOMjudge logo
- Home button (with a house icon)
- Problemset button (with a list icon)
- Scoreboard button (with a bar chart icon)
- Submit button (green with a cloud icon)
- Logout button (blue with a door icon)
- Lab0 button (with a trophy icon and a dropdown arrow)
- Timer showing 11d 22:45:50

Below the navigation bar, there is a table summarizing a student's status:

RANK	TEAM	SCORE	HELLO WORLD
1	DOMjudge	0	0

Below the table, there are two main sections:

- Submissions:** A yellow box indicating "No submissions".
- Clarifications:** A section with a blue header, containing the text "No clarifications." and a sub-section "Clarification Requests" with the text "No clarification request." and a "request clarification" button.

Red arrows point from a central point below the navigation bar to each of the eight highlighted elements.

Note: **No submissions are accepted after lab-time expires.**

Under the top bar menu there is a table summarizing a student's status.

The screenshot shows the DOMjudge interface. At the top, there is a navigation bar with links: Home, Problemset, Scoreboard, and Jury. There are also buttons for Submit and Logout, and a Lab0 dropdown menu. The main content area features a scoreboard table with the following data:

RANK	TEAM	SCORE	HELLO WORLD
1	DOMjudge	0 0	

Below the table, there are two sections: Submissions and Clarifications. The Submissions section shows "No submissions". The Clarifications section shows "No clarifications." and "No clarification request." with a "request clarification" button.

- **RANK** - is not used for COMP6481 labs (for programming contest only)
- **TEAM** - shows StudentID
- **SCORE** - is not used for COMP6481 labs (for programming contest only)
- **The last column** - shows the problem name (for example "Hello World")

Note: problem names are hyperlinks. By clicking it the problem description will be opened in a new tab.

Finally under the table showing the student's status there are Submission and Clarifications menus.

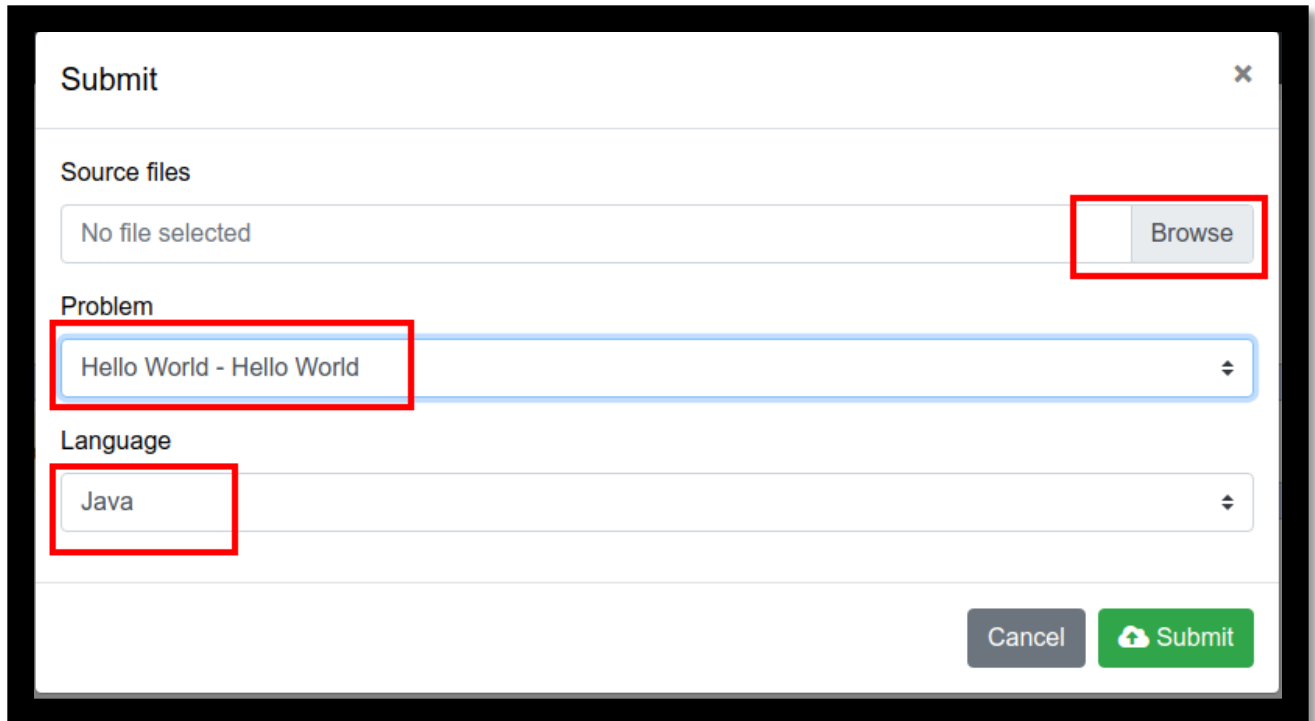
This screenshot is similar to the first one, but with red arrows pointing to the Submissions and Clarifications sections. The Submissions section is highlighted with a red box, and the Clarifications section is also highlighted with a red box. The scoreboard table is the same as in the first screenshot.

Clarifications are not used for COMP 6481 labs.

Submissions show results, time and status of student's submission(s).

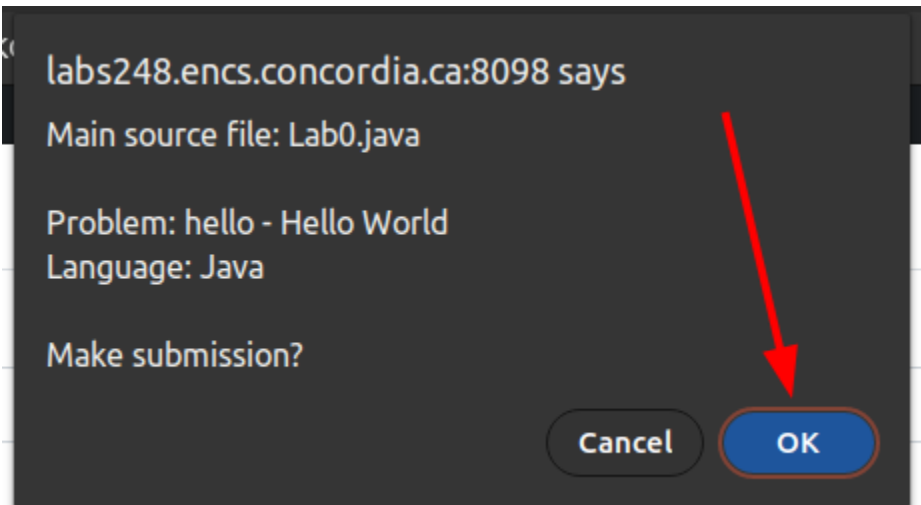
Submitting a lab solution

To submit a solution to the lab problem, on “**Home**” page find and click the green “**Submit**” button. A pop up will appear with submission menu (as below)



- **Source files:** is to navigate to the .java file stored on your local system.
- **Problem:** is to select the lab's problem
- **Language:** should be always “Java”
- To process solution evaluation click the green “Submit” button.

A popup might appear like the one below, to confirm submission. Click on “OK” button.



The Submissions section will show the summary of submission. Note the status will be “PENDING” until the system processes the submitted solution. Within 10 to 15 seconds the

status will change to correct or failure.

The screenshot shows the DOMjudge interface. At the top, there's a navigation bar with 'DOMjudge', 'Home', 'Problemset', 'Scoreboard', and 'Jury'. On the right, there are buttons for 'Submit' and 'Logout', and a user profile 'AdminTestLab' with a clock showing '80d 17:32:13'. Below the navigation bar, there's a scoreboard table:

RANK	TEAM	SCORE	HELLO
1	DOMjudge	0	0

Below the scoreboard, there's a green notification bar that says 'Submission done! Watch for the verdict in the list below.' Below that, there's a 'Submissions' table:

time	problem	lang	result
15:27	HELLO	JAVA	PENDING

A red box highlights the 'Submissions' table, and a red arrow points to the 'PENDING' result. To the right of the 'Submissions' table, there's a 'Clarifications' section with 'No clarifications.' and a 'Clarification Requests' section with 'No clarification request.' and a 'request clarification' button.

If the your submission failed like on the screenshot below, correct java code and resubmit it again until the result is “**CORRECT**”

The screenshot shows the DOMjudge interface with the same navigation bar as the previous screenshot. The scoreboard table now shows:

RANK	TEAM	SCORE	HELLO
1	DOMjudge	1	13553

Below the scoreboard, there's a 'Submissions' table:

time	problem	lang	result
15:33	HELLO	JAVA	CORRECT
15:27	HELLO	JAVA	WRONG-ANSWER

A red arrow points to the 'CORRECT' result. To the right of the 'Submissions' table, there's a 'Clarifications' section with 'No clarifications.' and a 'Clarification Requests' section with 'No clarification request.' and a 'request clarification' button.

Submission Results

Please pay attention to the table below. It shows possible results of your submission.

Make sure you understand these results. It will save you time during the lab, instead of waiting on your lab instructor explanation on what can be wrong with your submission.

A submission can have the following results:

CORRECT	The submission passed all tests: Problem is solved.
COMPILE-ERROR	There was an error when compiling your program. Note that when compilation takes more than 30 seconds, it is aborted and this counts as a compilation error.
TIMELIMIT	Your program run took longer than the maximum allowed time for this problem. This might indicate that your program hangs in a loop or the solution is not efficient enough.
RUN-ERROR	There was an error during the execution of the program. This can have a lot of different causes like division by zero, incorrectly addressing memory (wrong array indexing), etc. Check that your program exits with exit code 0!
NO-OUTPUT	Your program did not generate any output. Check that you write to standard out.
OUTPUT-LIMIT	Your program generated more output than the allowed limit. The output was truncated and considered incorrect.
WRONG-ANSWER	The output of your program was incorrect. This can happen simply because your solution is not correct, but remember that your output must comply exactly with the specifications of the problem. See section "How a Submission is Being Judged" below for more details.
TOO-LATE	Bummer, you submitted after the contest ended! Your submission is stored but will not be processed anymore

Appendix Code examples:

The examples below are solutions for the following problem:

- The first line of the input contains the number of test cases.
- Then each test case consists of a line containing a name (a single word) of at most 99 characters.
- For each test case output the string "Hello !" on a separate line.

Sample input and output for this problem:

Input	Output
3 world Jan SantaClaus	Hello world! Hello Jan! Hello SantaClaus!

Note that the number 3 on the first line indicates that 3 test cases follow

Java solution

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int nTests = scanner.nextInt();

        for (int i = 0; i < nTests; i++) {
            String name = scanner.next();
            System.out.println("Hello " + name + "!");
        }
    }
}
```


"C" solution

```
#include <stdio.h>

int main() {
    int i, ntests;
    char name[100];

    scanf("%d\n", &ntests);

    for (i = 0; i < ntests; i++) {
        scanf("%s\n", name);
        printf("Hello %s!\n", name);
    }
}
```

C++ solution

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int ntests;
    string name;

    cin >> ntests;
    for (int i = 0; i < ntests; i++) {
        cin >> name;
        cout << "Hello " << name << "!" << endl;
    }
}
```

Python Solution

```
import sys

n = int(input())
for i in range(n):
    name = sys.stdin.readline().rstrip('\n')
    print('Hello %s!' % (name))
```

Appendix: optional reading

Information described in the appendix is not for COMP6481 labs, but can be valuable for other courses with more advanced usage of the DomJudge platform.

Clarifications

Clarifications might not be used for the labs. Please consult with your LDs and instructor. In general Clarifications are to communicate with the LDs during a lab to clarify an uncertainty. There is a button to submit a new clarification request to an LD. You can associate a specific problem or a general category to a request. Clarification requests are only readable for the LDs. An LD can answer specifically to a student or send a reply to everyone if it is relevant for all.

How a submission is being Judged?

Output Evaluation

After your program has compiled successfully it will be executed and its output compared to the output provided by the lab instructor. Before comparing the output, the exit status of your program is checked: if your program exits with a non-zero exit code, the result will be a run-error even if the output of the program is correct! There are some restrictions during execution. If your program violates these it will also be aborted with a run-error, see section "[Restrictions](#)" below. When comparing program output, it has to exactly match the output of the instructor, except that some extra whitespace may be ignored (this depends on the system configuration of the problems). **So make sure that you follow the output specifications.**

Restrictions

To prevent abuse, keep the jury system stable and give everyone clear and equal environments, there are some restrictions to which all submissions are subjected:

Compile time	Compilation of your program may take no longer than 30 seconds. After that, compilation will be aborted and the result will be a compile error. In practice this should never give rise to a problem.
Restrictions below are NOT used for COMP 6481 course	
Source Size	The total amount of source code in a single submission may not exceed 256 kilobytes, otherwise your submission will be rejected.
Memory	The judges will specify how much memory you have available during execution of your program. Note that this may vary per problem. This is the total amount of memory (including program code, statically and dynamically defined variables, stack, Java VM, . . .)! If your program tries to use more

	memory, it will most likely abort, resulting in a run error.
Number of processes	You are not supposed to explicitly create multiple processes (threads). This is to no avail anyway, because your program has exactly 1 processor core fully at its disposal. To increase stability of the system, DOMjudge executes submissions in a sandbox where a maximum of 15 processes can be run simultaneously (including processes that started your program). People who have never programmed with multiple processes (or have never heard of “threads”) do not have to worry: a normal program runs in one process.

Contributors

Concordia is using the DomJudge Programming Contest System (domjudge.org) to implement several COMP courses labs and quizzes. The Manual is based on the original DomJudge Teams Manual adjusted for COMP6481 course needs.