How to Use the System

When Getting in Trouble with the System

Raise your hand to call a staff member and tell them about your trouble. Here are a few phrases that may be useful in typical situations:

- "We printed ____ (source code, debug output, …) about ____ minutes ago but have not received the printout yet."
- "____ (browser, editor, …) does not start."
- "Our ____ (computer, keyboard, mouse, …) stopped working."

Contest staff will help you resolve the issue.

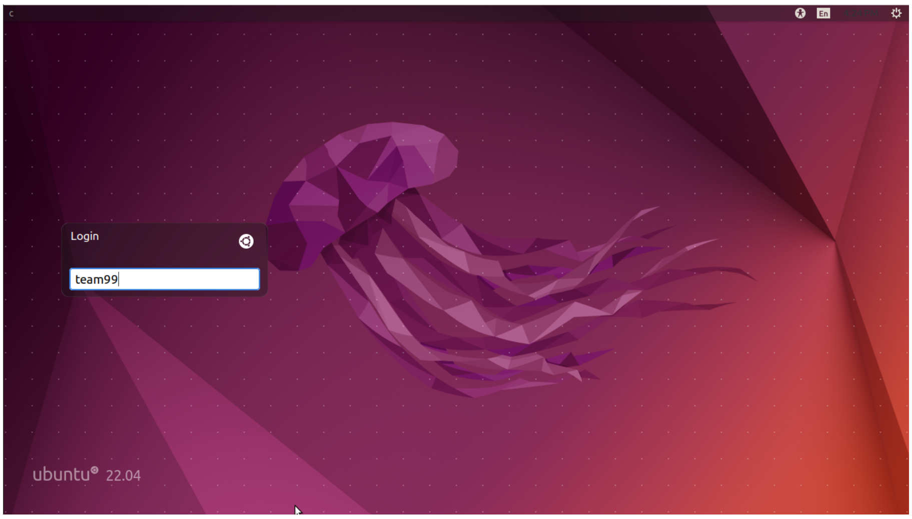
Do not use the Clarification in DOMjudge for system troubles. Use the Clarification only for questions about contest problems.

Logging In to Your Computer

Your team ID and password are provided in a document in an envelope.

Note: You will use *the same team ID* but *different passwords* in the practice session (Nov. 25) and the main contest (Nov. 26). You will receive a document of login information each day.

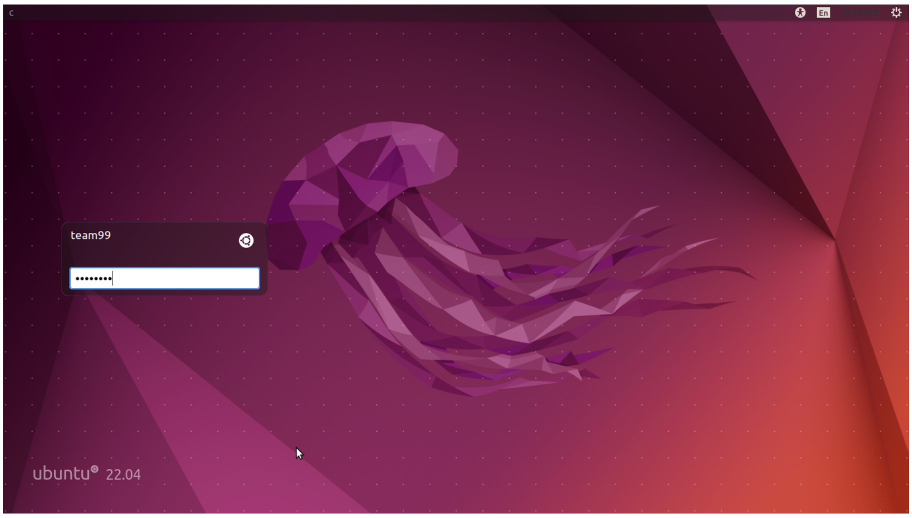
1. Type your team ID (`team01` or `team02` or …) and press an `Enter` key.



Login

team99

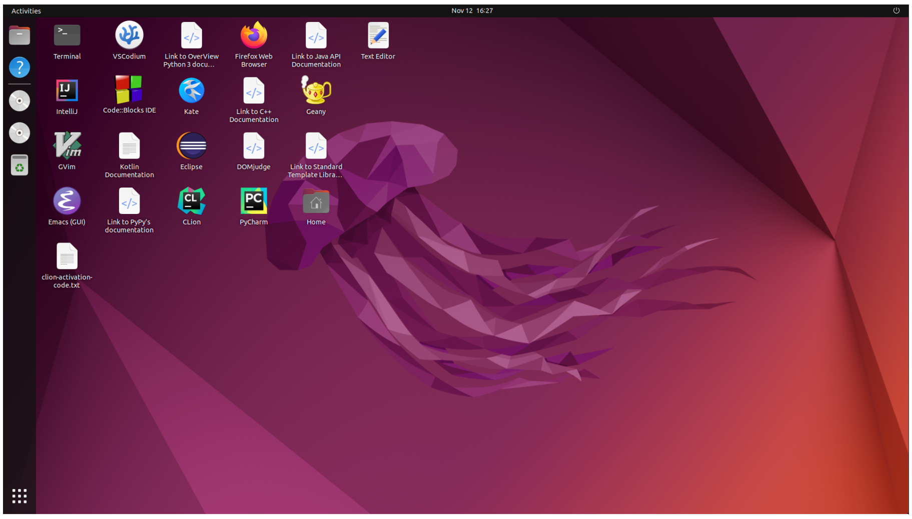
2. Type your password and press an `Enter` key.



team99

••••••••

3. You should see a screen like this.

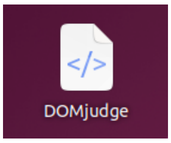


Activities
Nov 12 16:27
Terminal
VSCodium
Link to OverView Python 3 docu...
Firefox Web Browser
Link to Java API Documentation
Text Editor
IntelliJ
Code::Blocks IDE
Kate
Link to C++ Documentation
Geany
GVim
Kotlin Documentation
Eclipse
DOMjudge
Link to Standard Template Libra...
Emacs (GUI)
Link to PyPy's documentation
CLion
PyCharm
Home
clion-activation-code.txt

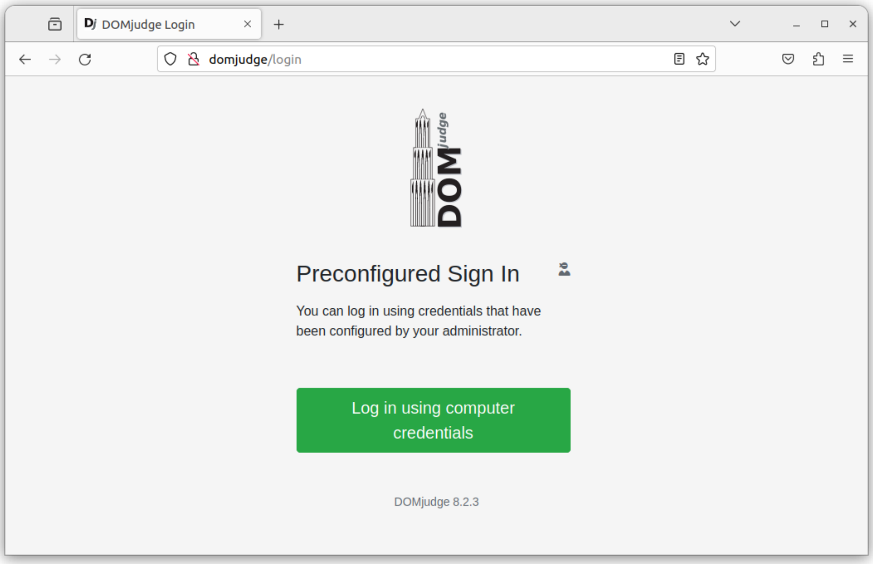
Logging In to the Contest System

We use DOMjudge 8.2.3 for the contest system this year.

1. Double-click the `DOMjudge` shortcut on the desktop. It will open a new Firefox tab.



2. Click the `[Log in using computer credentials]` button. It will automatically log in as your team.

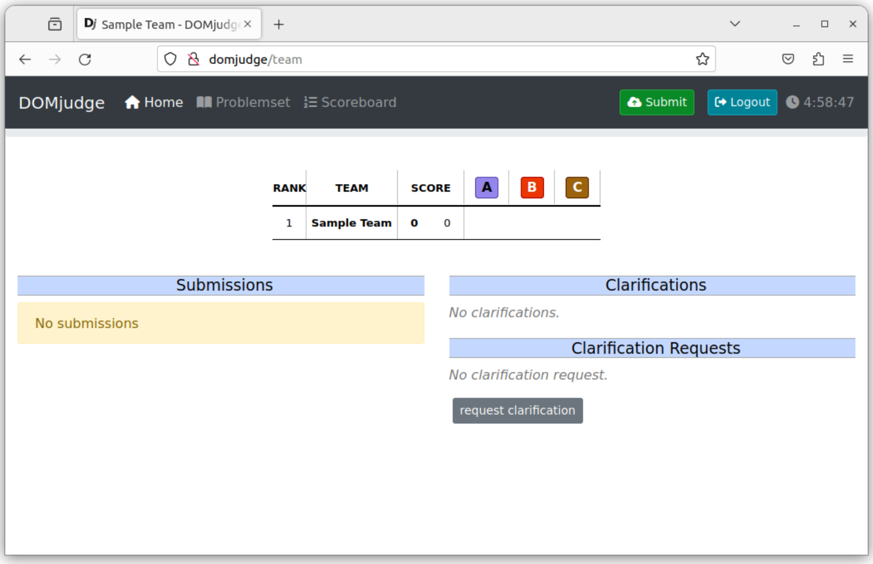


Preconfigured Sign In

You can log in using credentials that have been configured by your administrator.

DOMjudge 8.2.3

3. Once you get logged in, you will see a page like this.



RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

Submissions

No submissions

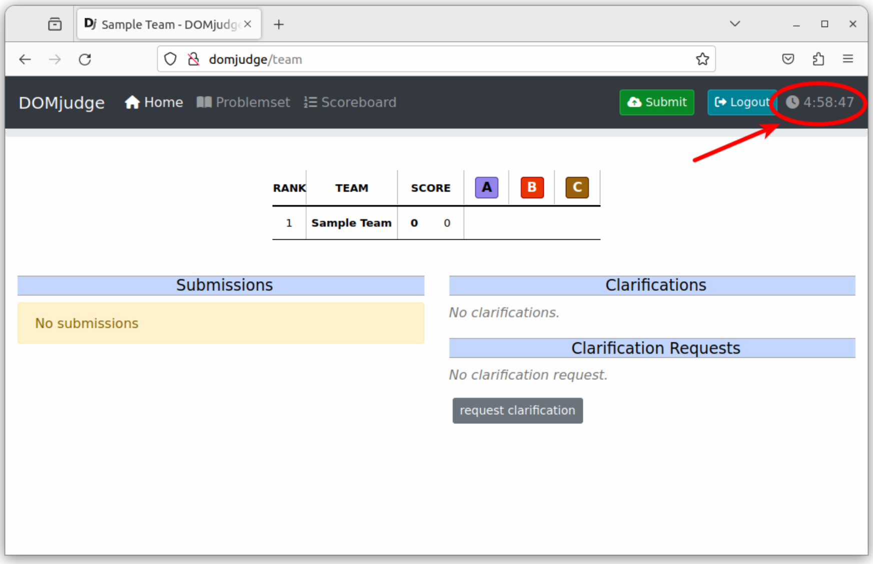
Clarifications

No clarifications.

Clarification Requests

No clarification request.

4. You can check the time to end at the upper-right corner.



RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

Submissions

No submissions

Clarifications

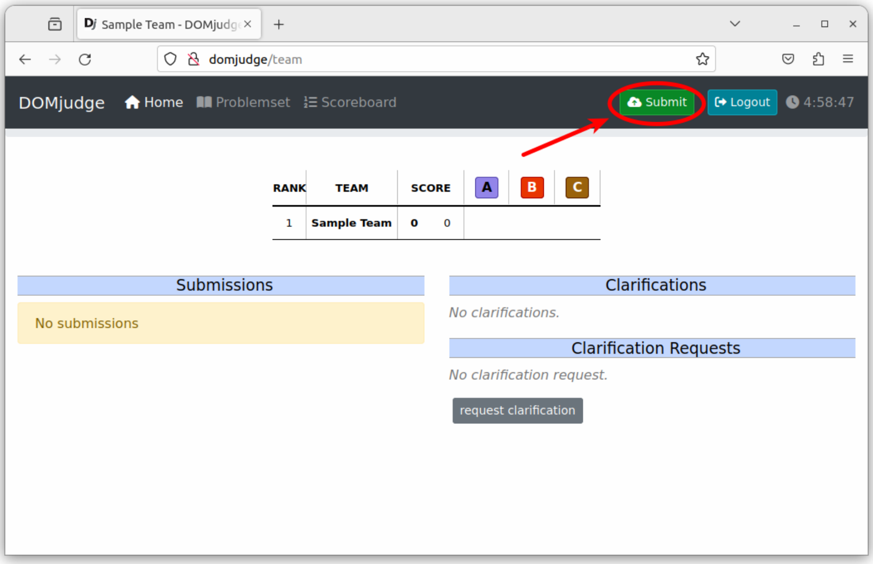
No clarifications.

Clarification Requests

No clarification request.

Submitting Your Code

1. Click the `[Submit]` button near the upper-right corner. It will open the Submit dialog.



RANK	TEAM	SCORE	A	B	C
1	Sample Team	0 0			

Submissions

No submissions

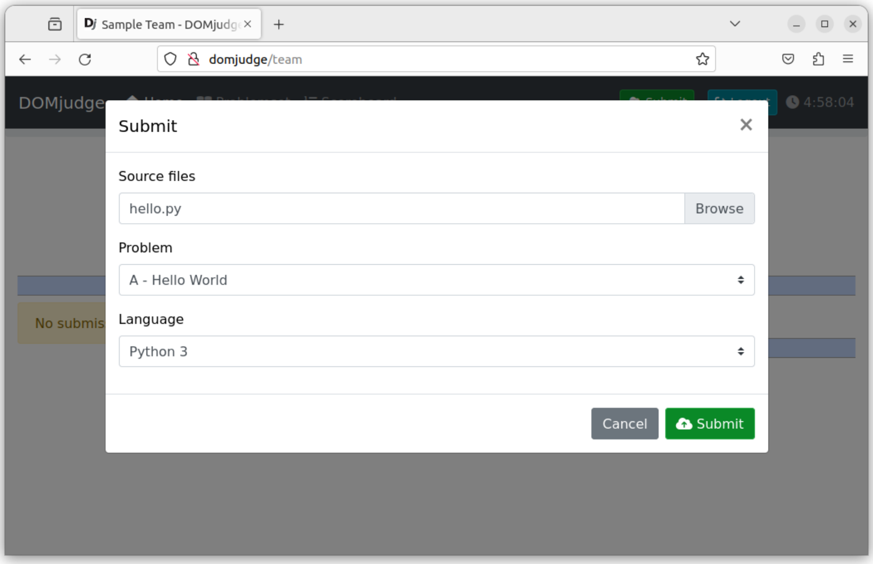
Clarifications

No clarifications.

Clarification Requests

No clarification request.

2. Choose your source files, problem, and language, then click the `[Submit]` button in the dialog.



Source files: hello.py

Problem: A - Hello World

Language: Python 3

Your submission results show up in the Submission section. You can see the details of each submission by clicking the corresponding row.

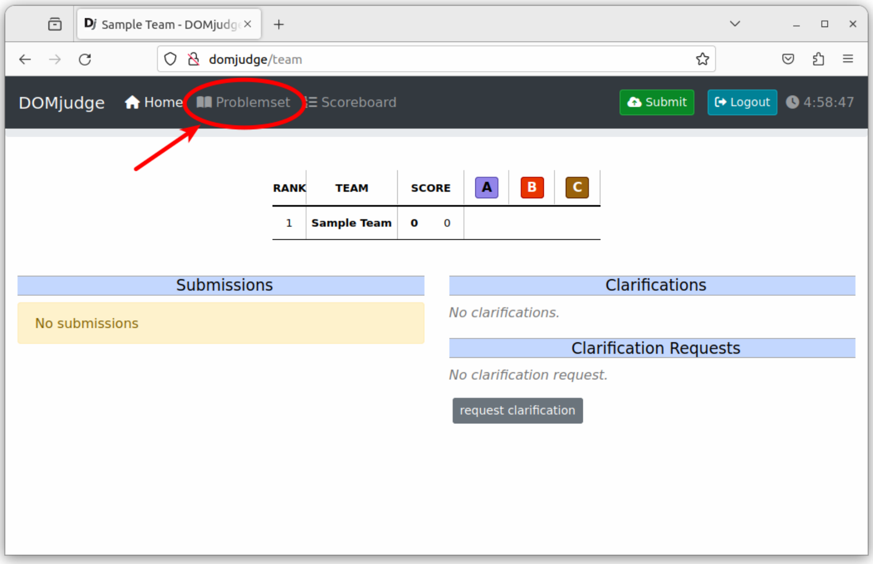
The Submission section also shows the submissions from your teammates. You can download the submitted code as a zip file there.

You can submit multiple source files: hold a `Ctrl` key down when choosing files.

Filenames may contain only alphabets (`A-Z`, `a-z`), digits (`0-9`), and a few symbols (`+._-`). They must start with an alphabet or a digit. In other words, they may not start with a symbol.

Downloading Sample Input/Output

1. Click `[Problemset]` in the top navigation. It will open the Contest problems page.



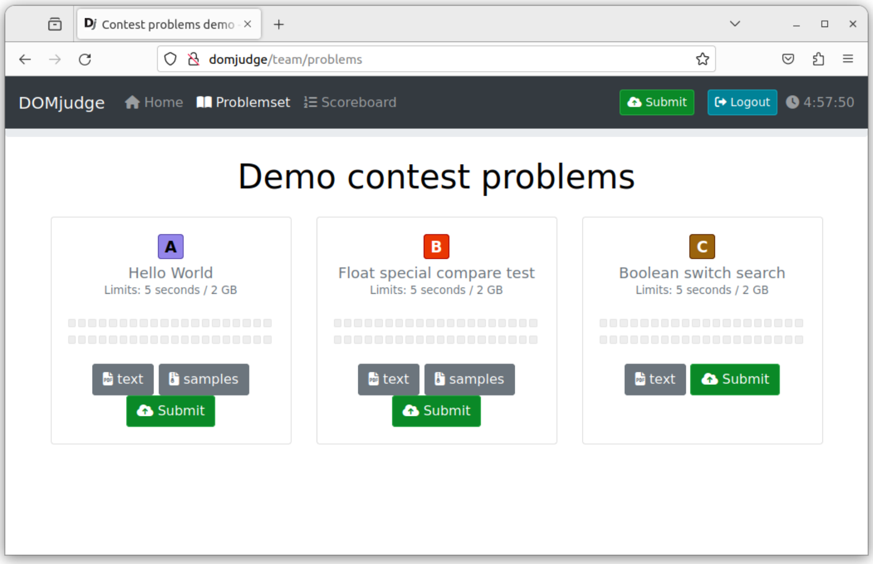
RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

No submissions

No clarifications.

No clarification request.

2. Click the button of what you want to download.



Demo contest problems

A
Hello World
Limits: 5 seconds / 2 GB

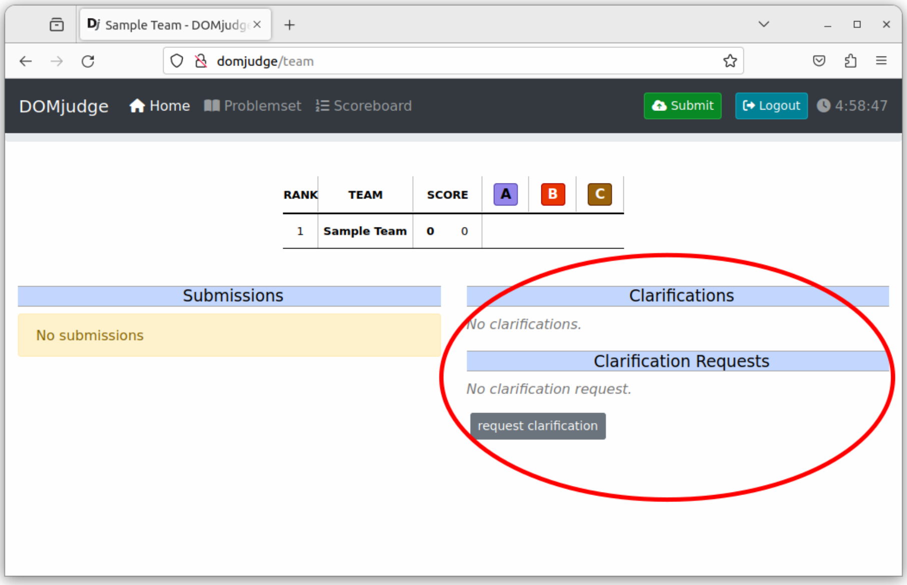
B
Float special compare test
Limits: 5 seconds / 2 GB

C
Boolean switch search
Limits: 5 seconds / 2 GB

Communicating with the Judge

On the right side is the only communication channel between you and the judge.

The Clarifications section shows messages from the judge. When the judge makes any announcements, they will show up in this section.



RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

Submissions

No submissions

Clarifications

No clarifications.

Clarification Requests

No clarification request.

To ask a question to the judge:

1. Click the `[request clarification]` button. It will open the Send clarification request dialog.
2. Choose the Subject, write your question in the Message field, and click the `[Send]` button.

Your question will show up in the `Clarification Requests` section until it is answered. After you receive an answer, it will appear in the Clarifications section.

Be sure to write your question in English when you talk to the judge! Also, remember the judge **does not** answer system-related questions, such as a usage of DOMjudge or a trouble with your computer. If you have such questions, raise your hand and talk to the contest staff.

Printing Your Code and Debug Output

To print a text file, such as your code and debug output, open a terminal and run:

```
printfile $FILENAME
```

This command outputs "`Print request was sent.`" on success.

Your printouts will be delivered to your seat by staff. **Do not** attempt to pick them up by yourself; the printers are located in the staff area, which you cannot enter.

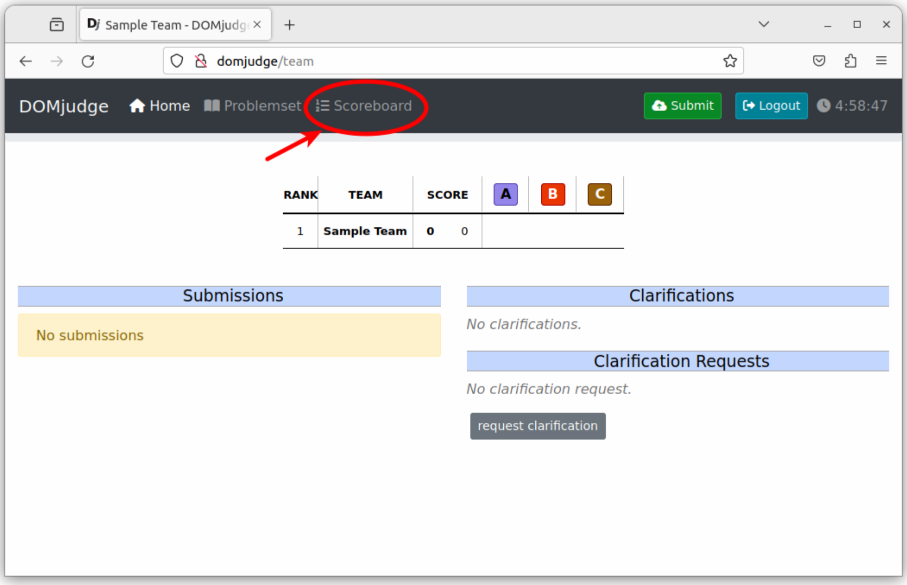
Do not use other methods, especially printing features in applications.

Only plain text is supported. **Do not** print anything else (e.g. PDF, HTML).

We do not have many printers. **Do not** attempt to print many pages. Printing the whole content of an STL document or a Java API document is a bad idea, for example.

Viewing the Scoreboard

Click `[Scoreboard]` in the top navigation to show the current standings. Note the scoreboard will stop being updated (be "frozen") 60 minutes before the contest ends.



RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

Submissions

No submissions

Clarifications

No clarifications.

Clarification Requests

No clarification request.

Writing and Compiling Your Code

The system is set up with a number of IDEs and editors. Use your favorite. There is a shortcut on the desktop for each of them.

You can test your programs with the following commands. Just pass your source file(s) or Java class file(s) as the arguments. See Appendix A for more detailed instructions.

- C: `compilegcc` (then run `./a.out`)
- C++: `compileg++` (then run `./a.out`)
- Java: `compilejava` / `runjava`
- Python 3: `compilepython3` / `runpython3`
- Kotlin: `compilekotlin` / `runkotlin`

Customizing the System

We strongly discourage you from changing system settings of the computer. If you choose to do so, you are responsible for all troubles it could cause, i.e. you will be doing it at your own risk.

Also be advised the system will be reset to the original state between the practice and the main contest. All files and customizations made during the practice will be lost.

Taking Out Your Files after the Contest

If you would like to take your files out of the computer after the contest, please put them in:

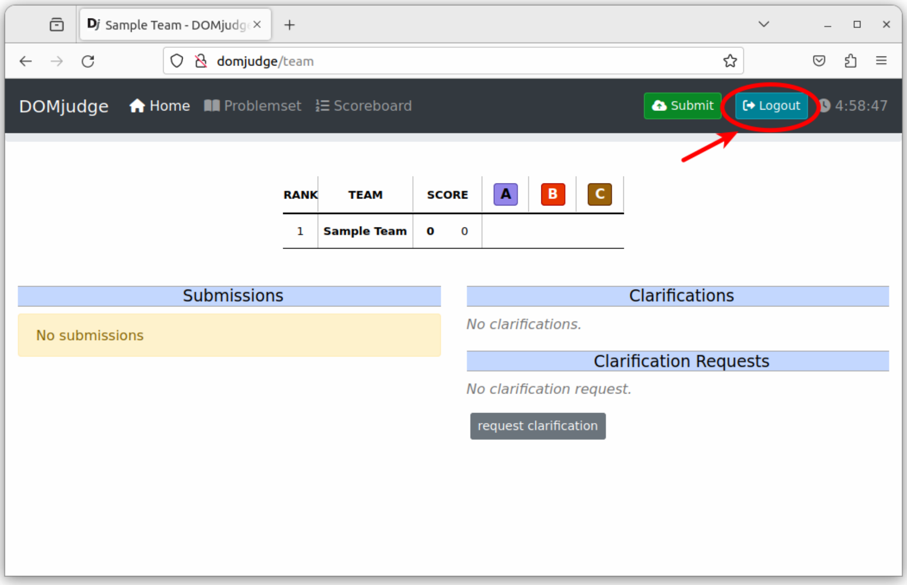
```
~/takeout
```

We will pick up the files there and send them to you by email. We will give you some time to copy the files after the contest is over, so that you do not have to care about it during the contest.

See Appendix B if you need step-by-step instructions.

Logging Out from the Contest System

Click the `[Logout]` button at the upper-right corner of the page.



RANK	TEAM	SCORE		A	B	C
1	Sample Team	0	0			

Submissions

No submissions

Clarifications

No clarifications.

Clarification Requests

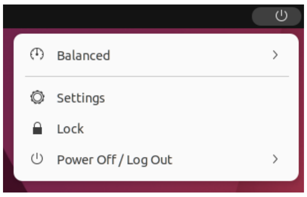
No clarification request.

request clarification

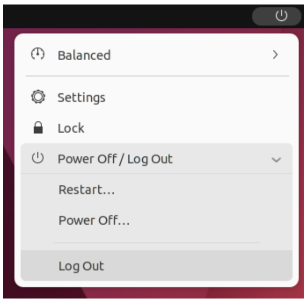
Logging Out from Your Computer

1. Click a power-icon button at the upper-right corner of the screen.

2. Click your team name to open the submenu.



3. Select `[Log Out]` from the shown menu.



Appendix A: How to Compile and Run Your Code

C/C++

We assume that you save your source file as `A.c` or `A.cpp` and an input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.

2. Type as follows in the Terminal window (`↵` denotes an `Enter` key), depending on whether you are using C or C++. It compiles your code and generates a binary named `a.out` .

   - C:

   ```
   compilegcc A.c↵
   ```

   - C++:

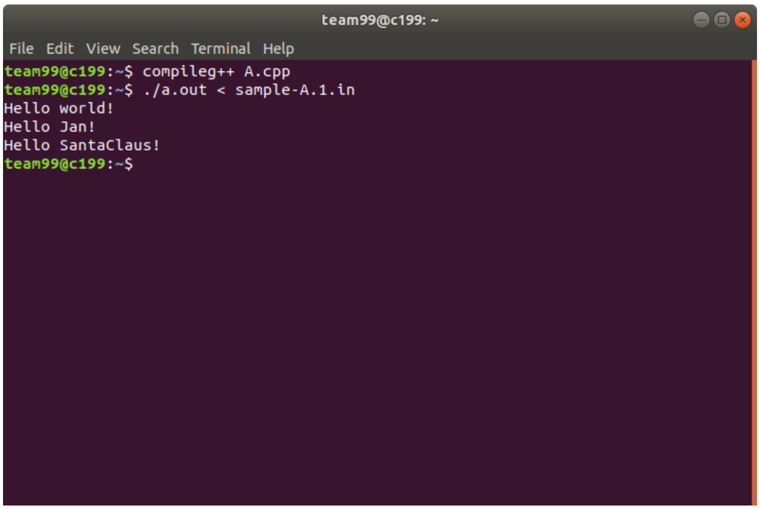
   ```
   compileg++ A.cpp↵
   ```

3. Type as follows to run your program:

   ```
   ./a.out < sample-A.1.in↵
   ```

   If you would like to save the output into a file named `my-A.1.out` :

   ```
   ./a.out < sample-A.1.in > my-A.1.out↵
   ```

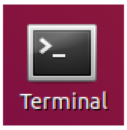


```
team99@c199:~$ compileg++ A.cpp
team99@c199:~$ ./a.out < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

Java

We assume that you save your source file as `A.java` and an input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.



2. Type as follows in the Terminal window (↵ denotes an `Enter` key). It compiles your code into Java class files (e.g. `A.class`).

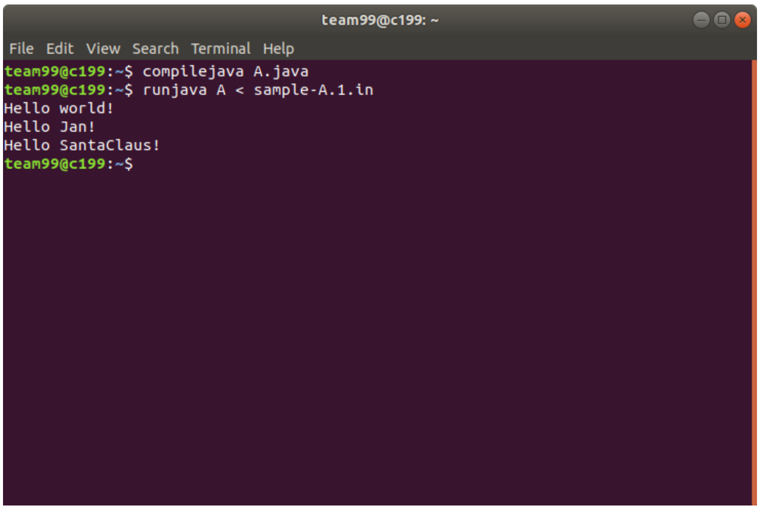
```
compilejava A.java↵
```

3. Type as follows to run your program:

```
runjava A < sample-A.1.in↵
```

If you would like to save the output into a file named `my-A.1.out` :

```
runjava A < sample-A.1.in > my-A.1.out↵
```



```
team99@c199:~$ compilejava A.java
team99@c199:~$ runjava A < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

Python

We assume that you save your source file as `A.py` and an input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

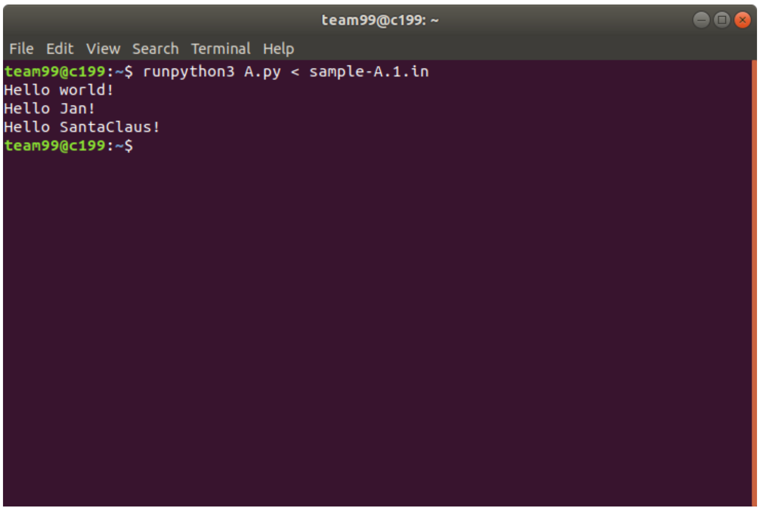
1. Double-click the Terminal shortcut on the desktop.

2. Type as follows in the Terminal window to run your program (↵ denotes an `Enter` key):

```
runpython3 A.py < sample-A.1.in↵
```

If you would like to save the output into a file named `my-A.1.out`:

```
runpython3 A.py < sample-A.1.in > my-A.1.out↵
```



```
team99@c199:~$ runpython3 A.py < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

Kotlin

We assume that you save your source file as `A.kt` and an input file as `sample-A.1.in` in your home folder (`teamXX`). Do not create them on the desktop; we do not expect them to be saved there.

1. Double-click the Terminal shortcut on the desktop.

2. Type as follows in the Terminal window (`↵` denotes an `Enter` key). It compiles your code into Java class files (e.g. `AKt.class`).

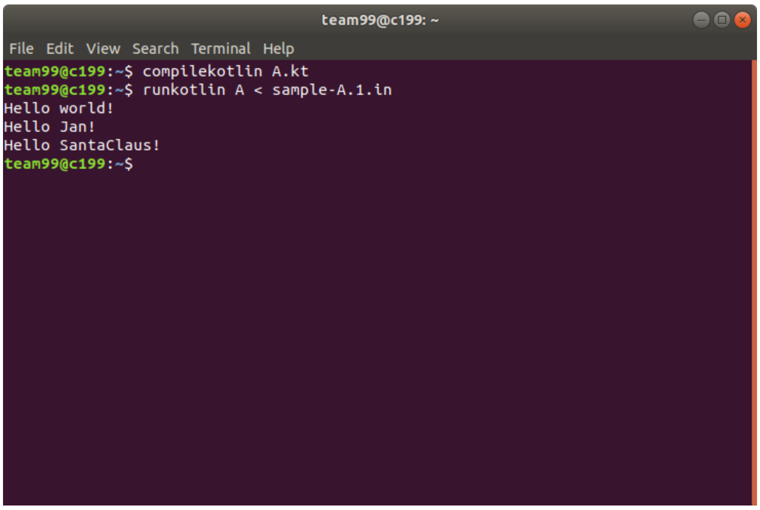
```
compilekotlin A.kt↵
```

3. Type as follows to run your program:

```
runkotlin A < sample-A.1.in↵
```

If you would like to save the output into a file named `my-A.1.out`:

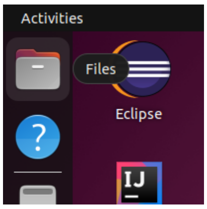
```
runkotlin A < sample-A.1.in > my-A.1.out↵
```



```
team99@c199:~$ compilekotlin A.kt
team99@c199:~$ runkotlin A < sample-A.1.in
Hello world!
Hello Jan!
Hello SantaClaus!
team99@c199:~$
```

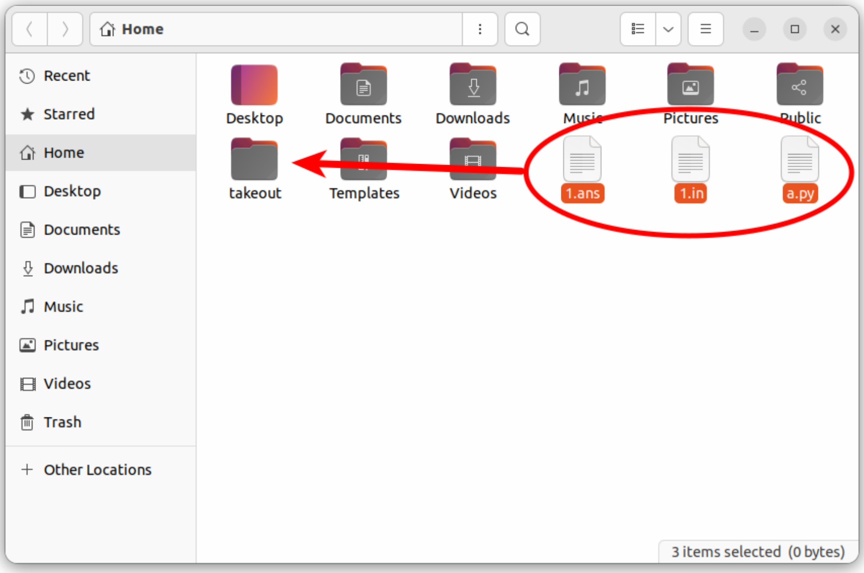
Appendix B: How to Move Your Files to Take Out

1. Click `[Files]` in the launcher (on the left side of the screen) to open your home folder.



2. Then, select your files to copy. You can use `Shift` and `Ctrl` keys as you do on Windows/Mac. [*1]

3. Drag and drop them into the takeout folder.



3 items selected (0 bytes)

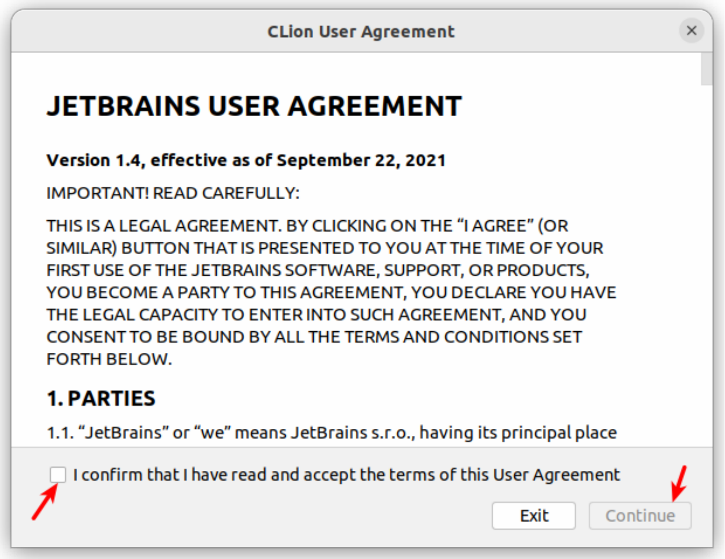
Note: We will give you some time to copy the files after the contest is over, so that you do not have to care about it during the contest.

[1] For Mac users, `Ctrl` behaves like `Command` here.

Appendix C: How to activate CLion

To use CLion IDE, you need to activate it. Follow the steps below. Note that you have to do this activation twice, in the practice session and in the main contest, because your system is reset before the main contest.

1. When you run CLion for the first time, you will see the CLion User Agreement dialog. Read the content carefully. Once you are done, check `[I confirm ...]` and click `[Continue]`.



CLion User Agreement

JETBRAINS USER AGREEMENT

Version 1.4, effective as of September 22, 2021

IMPORTANT! READ CAREFULLY:

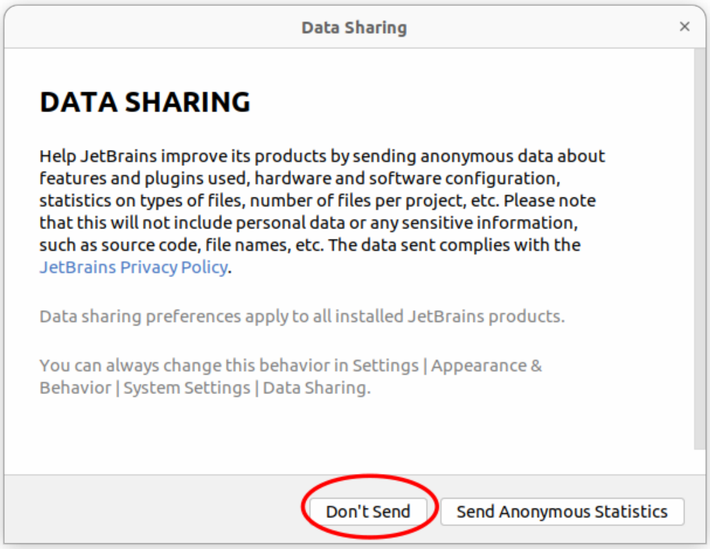
THIS IS A LEGAL AGREEMENT. BY CLICKING ON THE "I AGREE" (OR SIMILAR) BUTTON THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE JETBRAINS SOFTWARE, SUPPORT, OR PRODUCTS, YOU BECOME A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL CAPACITY TO ENTER INTO SUCH AGREEMENT, AND YOU CONSENT TO BE BOUND BY ALL THE TERMS AND CONDITIONS SET FORTH BELOW.

1. PARTIES

1.1. "JetBrains" or "we" means JetBrains s.r.o., having its principal place

□ I confirm that I have read and accept the terms of this User Agreement

2. On the Data Sharing dialog, click `[Don't Send]`.[2]



Data Sharing

DATA SHARING

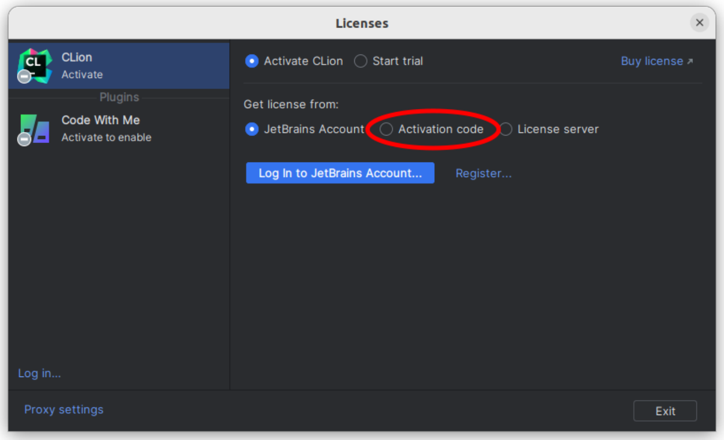
Help JetBrains improve its products by sending anonymous data about features and plugins used, hardware and software configuration, statistics on types of files, number of files per project, etc. Please note that this will not include personal data or any sensitive information, such as source code, file names, etc. The data sent complies with the JetBrains Privacy Policy.

Data sharing preferences apply to all installed JetBrains products.

You can always change this behavior in Settings | Appearance & Behavior | System Settings | Data Sharing.

[2] In our use case, both buttons work in the same way. It is okay to choose `[Send anonymous statistics]`, but the statistics will not be sent to JetBrains. Remember that your computer can access only to DOMjudge, nowhere else on the internet.

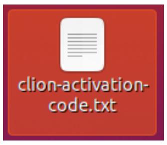
3. On the Licenses dialog, choose `[Activation code]`.



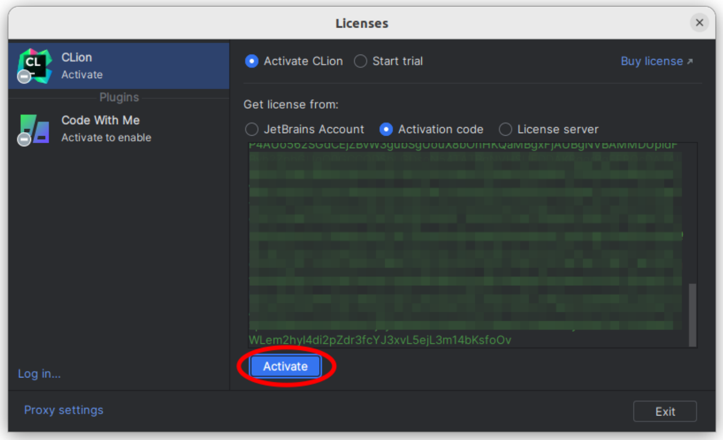
● Activate CLion

● JetBrains Account

4. On the desktop, find a file named `clion-activation-code.txt`. Double click to open it.



5. Copy-and-paste the file content into the CLion's dialog. Then click `[Activate]`.

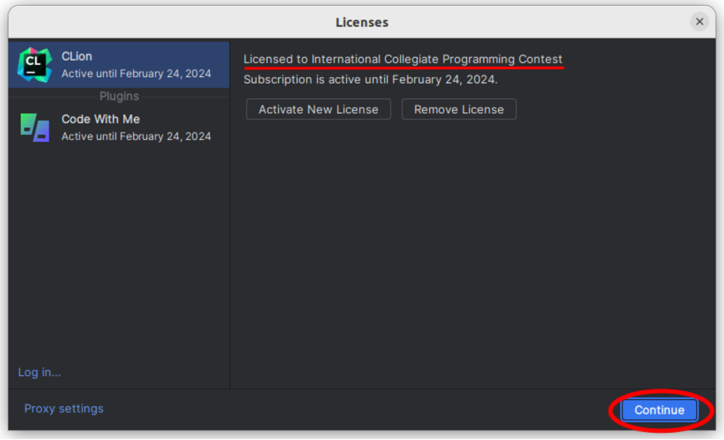


● Activate CLion

● Activation code

[redacted]

6. Now you should see "`Licensed to International Collegiate Programming Contest`" in the dialog. Click `[Continue]` to start using CLion.



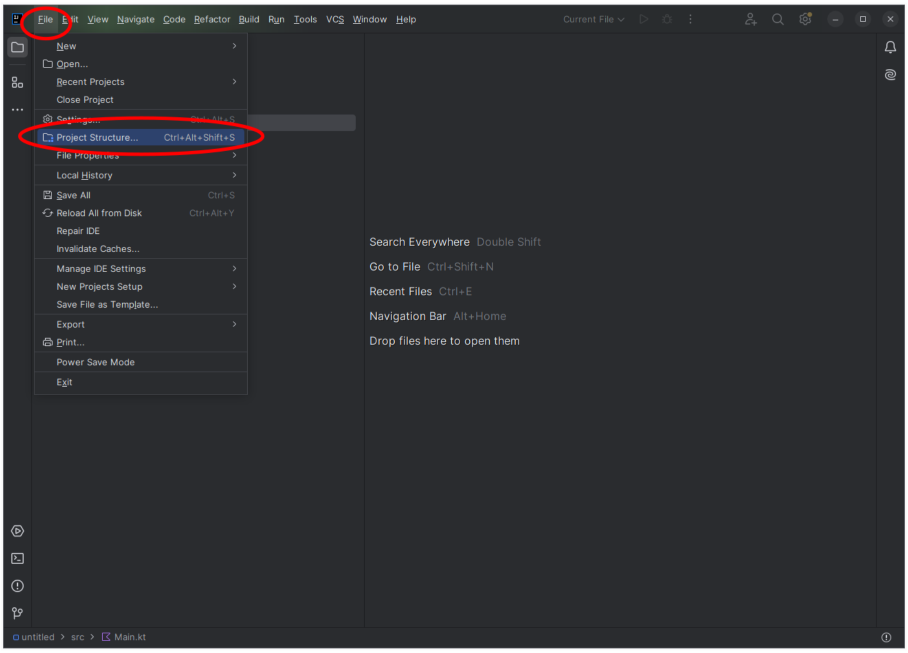
Licensed to International Collegiate Programming Contest

Subscription is active until February 24, 2024.

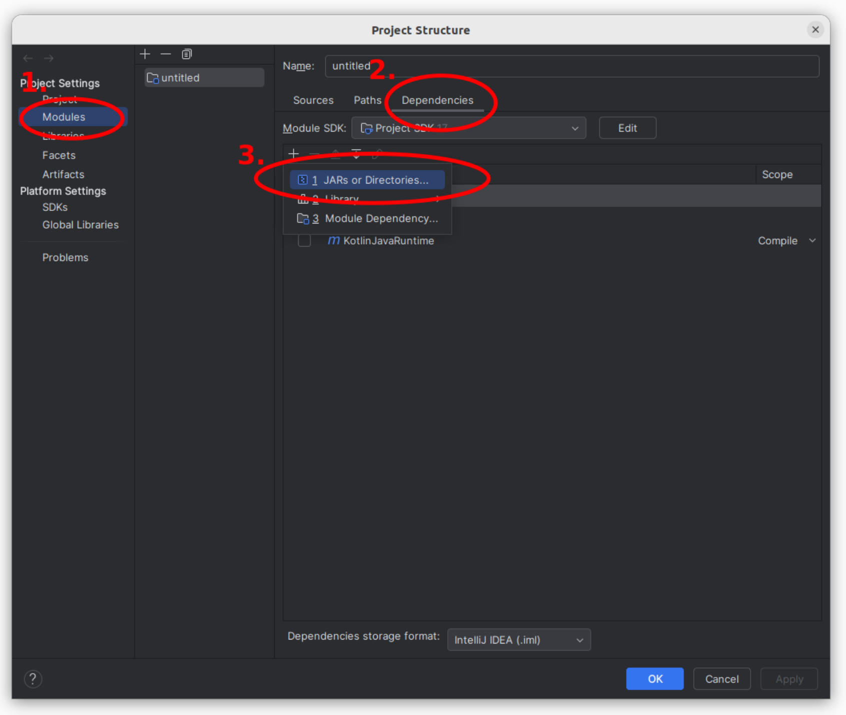
Appendix D: How to use Kotlin in IntelliJ IDEA

To use Kotlin's language features in IntelliJ IDEA (such as compiling, autocomplete, debugging, etc), you need to follow the steps below.

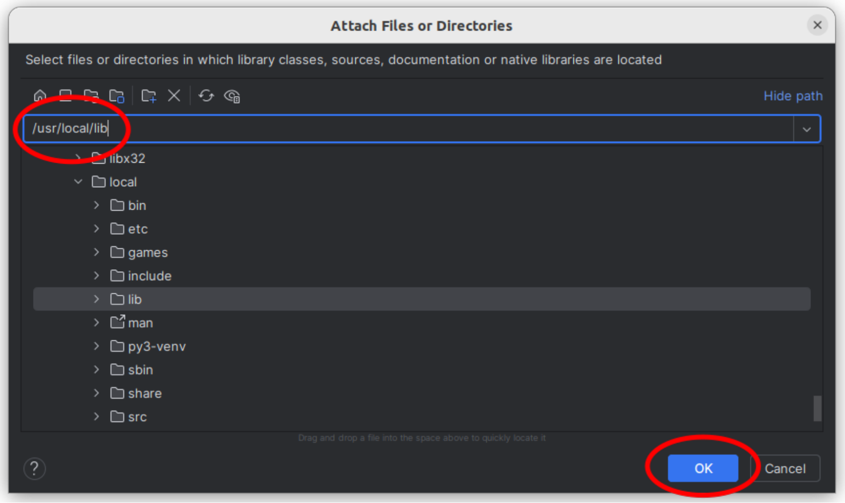
1. Open IntelliJ IDEA from the desktop icon, and create a new Kotlin project.
2. From the top right "Main menu" (hamburger menu), click the `[Project Structure]` option. You can alternatively open it by pressing `Ctrl`, `Shift`, `Alt`, and `S`.



3. Open `Modules` pane. From `Dependencies` tab, click `[+]` icon and click `[1. JARs or Directories...]`.



4. Add `/usr/local/lib` to the dependencies, and click `[OK]`.



Attach Files or Directories

Select files or directories in which library classes, sources, documentation or native libraries are located