

# Kattis Team Guide



2017-04-01

# Contents

<b>1</b>	<b>Using Kattis</b>	<b>2</b>
1.1	Submitting from the command line . . . . .	2
1.2	Submitting through the browser . . . . .	2
1.3	How does Kattis handle a submitted program? . . . . .	2
1.4	Tracking your submissions . . . . .	3
1.5	Clarification requests . . . . .	3
<b>2</b>	<b>Judge replies to submissions</b>	<b>4</b>
2.1	Accepted . . . . .	4
2.2	Compile Error . . . . .	4
2.3	Wrong Answer . . . . .	4
2.4	Run Time Error . . . . .	4
2.5	Time Limit Exceeded . . . . .	4
2.6	Judge Error . . . . .	5
<b>3</b>	<b>Technical information</b>	<b>5</b>
3.1	Input/Output . . . . .	5
3.2	Compilation . . . . .	5
3.3	System libraries . . . . .	5
<b>4</b>	<b>Things to keep in mind</b>	<b>6</b>
4.1	Be careful about your program's exit status . . . . .	6
4.2	Package declarations in Java . . . . .	6
4.3	File name restrictions . . . . .	6

# 1 Using Kattis

You start out by selecting a suitable problem to solve. Then you write code to solve the problem in your favorite language (or some other language, if your favorite one is not supported at the contest). After this, you submit the code to Kattis. Kattis will then compile your code and run it on some secret input. After some careful deliberation, you will get a judgement informing you whether your code behaved as expected or not.

## 1.1 Submitting from the command line

The submit script (available on your machine as the command-line utility `submit`) allows you to easily submit your program from the command line. If you name the file after the problem (e.g. `hello.c`, `hello.cc`, `hello.java` or `hello.py`) it will guess the problem as `hello`. In the cases of Java and Python it will also guess the main class as `hello` and main file as `hello.py` respectively. If the submission script guesses the problem incorrectly, you can specify it using command line flags and its indicated name or its upper case letter (e.g., 'E' for the fifth problem). The submit script also allows you to submit multiple files if you need to.

A typical session using the submit script looks like this:

```
$ submit hello.java
Problem: hello
Language: Java
Files: hello.java
Mainclass: hello
Submit (y/N)?
y
Submission received. Submission ID: 1445
```

## 1.2 Submitting through the browser

Another way to submit your program to Kattis is to log into the web interface and choose submit. There you have a web form where you can upload files or use the embedded editor.

## 1.3 How does Kattis handle a submitted program?

First, Kattis will compile your program. If the compiler fails to compile your program, or does not complete in a reasonable amount of time, Kattis will judge it as Compile Error. Otherwise, Kattis will execute the compiled binary on the first input file (there may be several). If the execution takes too long it will be judged as Time Limit Exceeded. If it crashes, terminates with non-zero exit status, or uses too much memory it will be judged as a Run Time Error. If the execution terminates correctly (exit status 0), Kattis will inspect the output produced to verify that it is correct. If it is incorrect, Kattis will judge the submission as Wrong Answer.

Note that your submission's output is only inspected if it successfully terminates in time, thus a Time Limit Exceeded or Run Time Error does not mean that output was correct up until the point where the program crashed or time ran out. Note that if your program should, for instance, divide by zero after the time limit has been reached, it will be judged as Time Limit Exceeded rather than Run Time Error.

If there are multiple test files, Kattis will continue with the same procedure for the next test file, as long as no error has been found. As soon as an error is detected Kattis will stop and report that error. Each test file is run with a new invocation of your program, so your program does not need to be adapted in any way to handle multiple test files. Each test file will follow the input specification for the problem.

If your program passes all test files successfully, it will be judged as Accepted.

## 1.4 Tracking your submissions

You can track the status of your submission by logging into the web interface and choosing your name from the top right menu. On this page you will see a list of all submissions you have made, in reverse chronological order. As the submission proceeds through the judgement process your submissions page will reflect this. The states a submission will pass through while it is being judged are:

- New
- Waiting for Compile
- Compiling
- Waiting for Run
- Running
- Final Judgement (see Section 2).

## 1.5 Clarification requests

During the contest you can submit requests for clarifications to the judges. To do so, click on Clarifications. The clarifications page has three sections:

- submitted clarification requests from your team that have not yet been answered,
- a form for submitting a clarification request, and
- clarification requests with answers from the judges.

When you submit a clarification request, please select a subject (either one of the problems or "general") and write your request in English. The third section contains answers to your requests and sometimes answers to other teams' requests, in case the judges choose to reveal the question and the answer to all teams. Notifications that there are new clarification replies are displayed on all Kattis web pages, but not on other pages (e.g., API documentation and scoreboards).

## 2 Judge replies to submissions

When Kattis has judged your submission, you will get a reply telling you the status of your submission. You can view this reply in the Kattis web interface. The following replies are possible:

### 2.1 Accepted

Accepted means that your program has terminated successfully within the time limit and produced correct output. Congratulations!

*Accepted does not incur penalty time.*

### 2.2 Compile Error

Compile Error means that we failed to compile your source code. Extra information (with compiler output) can be found on the page for the submission which can help you debug the error. Information about what compilers and flags are used can be found on our technical info page.

Giving an incorrect main class for a Java program will cause a compile error! Compile times over 1 minute will cause a compile error.

*Compile Error does not incur penalty time.*

### 2.3 Wrong Answer

Wrong Answer means that your program finished within the time limit, but that the answer produced was incorrect. Make sure you follow the output specification to ensure that you don't get this error due to simple whitespace errors.

*Wrong Answer incurs penalty time*

### 2.4 Run Time Error

Run Time Error means that your program has crashed during execution with our secret test input (e.g. process exited with exit status  $\neq 0$  or the process was signalled). Make sure you return 0 from your program to not get Run Time Error. In Java uncaught exceptions or termination with a non-zero exit status will be judged as Run Time Errors.

*Run Time Error incurs penalty time.*

### 2.5 Time Limit Exceeded

Time Limit Exceeded means that your program ran for too long. When the time limit is exceeded, the program is terminated. The output produced is not inspected in this case, so getting Time Limit Exceeded does not mean that your program produced correct output, only that it did not exit in time.

*Time Limit Exceeded incurs penalty time.*

## 2.6 Judge Error

Judge Error means that you've found a bug (or at least misconfiguration) in the judge. This should never happen during the competition.

*Judge Error does not incur penalty time.*

## 3 Technical information

This is a technical description of how Kattis works, read this carefully.

### 3.1 Input/Output

Your program should read its input from standard input (file descriptor 0 / `STDIN_FILENO` / `stdin` in C, `std::cin` in C++, `System.in` in Java, `sys.stdin` in Python) and produce output on standard output (file descriptor 1 / `STDOUT_FILENO` / `stdout` in C, `std::cout` in C++, `System.out` in Java, `sys.stdout` in Python). Anything written on standard error (file descriptor 2 / `STDERR_FILENO` / `stderr` in C, `std::cerr` in C++, `System.err` in Java, `sys.stderr` in Python) will be ignored, this can be used for debugging your program during development (i.e., you do not have to remove debug output before submitting if you use standard error for debug output). Of course, writing to standard error will take some run-time. You can mix reading and writing freely (all input in an input file will be available when your program starts).

When executing the program `hello` with the input file `in.txt` and output file `out.txt` the execution Kattis does will be identical to the following shell command

```
$ ./hello < in.txt > out.txt
```

After execution has finished Kattis will examine `out.txt` and compare it to the correct solution.

### 3.2 Compilation

For all languages, a maximum compile time of 1 minute is enforced. If compilation takes longer than that, it will cause a compile error.

### 3.3 System libraries

You are allowed to use all standard libraries included with the language you are working in. This includes the STL (Standard Template Library) for C++. The STL version available is the version which shipped with the installed GCC version.

## 4 Things to keep in mind

### 4.1 Be careful about your program's exit status

If you exit with anything except for exit status 0, your submission will be judged as a Run Time Error. So make sure you return 0 at the end of your program.

### 4.2 Package declarations in Java

You can declare your submitted Java files to be part of a package. However, when you do so, you must give the full name of the class, including the package name, as the “main class” parameter of your submission (cf. Section 1.1). For instance, if you submit the following program, you should specify “mypackage.Hello” as the main class.

```
package mypackage;

public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

### 4.3 File name restrictions

You must use valid file suffixes for the language selected (see notes under compilers). File names may only consist of alphanumeric ASCII characters, periods (“.”), underscores (“\_”), and plus (“+”, only for the suffix “.c++”). The first character of the filename must be alphanumeric. The last character must also be alphanumeric, unless the filename suffix is “.c++”. We strongly recommend naming the files after the names of the problems to simplify submissions using the submit script.