

# Coding Test Generic Instructions

---

## Language

You are free to use whatever language you are most comfortable working in. We will be running your submission on a unix/linux system (see next section).

## Command Line applications

Unless otherwise stated in the description of the problem, programs must be command-line ("console" or "terminal") applications. Under Linux, MacOS or other UNIX-like systems this means that that will be run from a bash shell within a terminal window. This applies even if the more normal mode of using the language would be with another kind of interface (such as a web application for JavaScript or PHP).

Unless otherwise stated, programs will take a number of command-line arguments and write their results to standard output (the console/terminal). So they will be run something like this:

```
MyProgram input.txt foo bar
```

In this case, the program MyProgram will be invoked with three command-line arguments, "input.txt", "foo" and "bar".

Your program should print only the specified output (or an error message where appropriate), and no additional text, blank lines etc. To be completely clear, A Java program with its main class compiled into MyProgram.class will be run as:

```
java MyProgram input.txt foo bar
```

JavaScript in MyProgram.js will be run as:

```
node MyProgram.js input.txt foo bar
```

Note: MyProgram is just an example – you should give your program a more meaningful name!

## Submitting your code

Your submitted program code may consist of multiple source files. In general it is fairly obvious how programs should be built or run, but you should include a prominent README file that gives instructions if it may be unclear how to do so. You should also include with your code any other data files, tests and so on to demonstrate that it works – but do not include any pre-compiled files (e.g. .jar, .class, .obj, .dll etc.).

Please send a link to your submission in a **publicly-accessible** code repository (e.g. GitHub or Gitlab).

## Other requirements

Your program must not rely on any libraries that do not form part of a normal base installation of the language, but you may use anything from within the language's standard libraries. So for example, unless explicitly asked to do so, we wouldn't expect you to implement your own sort routine if there is already one available in the standard libraries. The restriction on additional libraries does not apply to any tests or other collateral that you may wish to submit. So, for example, it's fine to use JUnit for Java, RSpec for Ruby and so on.

We will test your code, and expect you to have done so prior to submitting it. You will not automatically "fail" if we uncover bugs or limitations in your program. However, please be prepared to be challenged over

improvements and possible fixes as a part of the interview process. We will test your code with input sets other than those supplied in the problem description, so please expect that.

You may find that the specification of the problem seems ambiguous, or not as complete as you would like. That is, of course, part of the reality of software development! If you find that you have to make certain assumptions in order to implement your program then please do so, but make it clear what those assumptions are. We will not provide you with further examples or test cases – although if you find something in the problem specification that seems contradictory or just plain wrong then please let us know.

It should go without saying that the program must be your own work. Please do not ask an LLM to do your work for you - we are looking for developers not prompt engineers. If you take inspiration or small code snippets from other sources, you must acknowledge this with appropriate comments. In any event you will be expected to be able to explain in detail how every part of your program works and why you designed it like you did.

We're looking for evidence of clear, well-structured code and a good approach to development, so do include any tests or other relevant artefacts with your source code.

As part of the interview we will ask you to make a small enhancement to your program, so please bring a laptop with the necessary development tools etc installed.