

The purpose of this lab is to familiarize yourself with the CS 20 Osaka submission system and the OCaml style guide posted on the CS 20 home page. It is not intended to be a difficult assignment; if all goes well, it should not take more than 10-15 minutes.

Points for this lab will be awarded as follows:

Normal points: 1 (no credit if your code doesn't compile)  
Style points: 3 (the TAs will be very lenient on this lab)  
Bonus points: 3

For this lab, all we ask is that you implement a function called `factorial` that takes an integer as input and returns an integer. Your code should compile and contain documentation. For bonus points, your `factorial` function should actually compute the factorial of an integer.

You should submit your code in two files: `fact.mli` and `fact.ml`.

`fact.mli` should contain the following line (and perhaps nothing else):

```
val factorial : int -> int
```

`fact.ml` should contain a function that is declared as follows:

```
let rec factorial (n : int) : int =
```

Your job will be to finish defining the function and documenting what it is supposed to do. It *does not* have to compute a factorial necessarily; it may be absolutely trivial and have no relation to "factorial".

The following sequence of commands can be used to test that your code compiles:

```
ocamlc -c fact.mli  
ocamlc -c fact.ml
```

Follow the directions on the CS 20 home page to submit your code. The name of this assignment in Osaka is `lab0`. Send any questions about this lab to `cs20-admin@metapr1.org`.