

CS4215 Programming Language Implementation

Lab task for Week 02/03

OCaml Refresher Exercise

1. Familiarize yourself with one of the following two environments (i) emacs (ii) Eclipse IDE for OCaml. Both can be accessed by logging to `loris-88.ddns` using an account that we will be providing for you during the lab session. For the benefits of those who may be new to OCaml, we will be starting the lab session in Week 2.

2. For emacs, please take the following steps to set up your environment.

```
hg clone ~chinwn/emacs
cd emacs
sh install.sh
```

You are free to use a different editor if you wishes.

3. Please download the following two files from IVLE workbin

```
lab03.ml
lab03-extra.ml
```

The first file contains a series of small exercises and you have to hand in by 6pm 26Jan2017 on IVLE workbin. To encourage you to submit early The second file contains two optional exercises. You may do it for more practice.

4. Let us first look at three different ways of executing OCaml code. After you have unzipped the files, you will see the execution/compilation commands in `comp.sh`. This can be executed using `sh comp.sh`. The commands are:

- For OCaml interpreter in interactive mode. First execute `ocaml`. After that load the file using `# use "lec03.ml";;`. You will see the definitions and commands executed. You may then test the individual methods written using `# mn e1 .. en;;`. To quit, just type `# quit;;`.
- For OCaml interpreter in batch mode, execute it using `ocaml lec03.ml`. Take note that only commands, such as `print_endline`, to print to the console will cause some output to be displayed.

- For OCaml bytecode compiler, you may generate bytecode by using `ocamlc lec03.ml -o ex3.byte`. This will produce a bytecode that can be invoked using `ocamlrun ex3.byte` or `./ex3.byte`
 - For OCaml native compiler, you may generate machine code executable by using `ocamlopt lec03.ml -o ex3`. This will produce an executable that can be invoked using `./ex3`
5. Please make sure you complete your lab exercises on time. You are welcomed to discuss with the TA/lecturer in case you have problems.
 6. Have fun with OCaml!