

University of Puerto Rico  
Mayagüez Campus  
College of Engineering  
Department of Electrical and Computer Engineering

**ICOM4029 – Compilers**  
Professor: Bienvenido Vélez  
Technical Assistant: René D. Badía

## **Lab 1 – Introduction to Cool**

*Cool*, or Class Object-Oriented Language, is a simple language specifically designed for use in a compilers course. It contains enough necessary properties and features for learning how a compiler works and to be used as a basis for developing your own compiler.

The following steps will guide you on creating your first *Cool* program, compiling it, and running it using the *spim* MIPS emulator.

### **\*Side note: Accounts\***

Your class account's username is: 4029\_aXX

where XX represents your student number as it appears on the daily assistance sheet

The initial password is: ChangeMe (it's case-sensitive)

You must/should change your password by typing: *yppasswd* at the command line / terminal and following the instructions onscreen.

### **I. Preparation**

After logging on, open a terminal window and enter the following commands in the order they appear:

1. `export PATH=$PATH:~icom4029/cool/bin:~`
2. `cp ~rbadia/compfiles/myspim ~`
3. `cp ~rbadia/compfiles/.bash_profile ~`

The first command will include *Cool*'s binaries directory in the path environment variable so you can use its executables from any folder.

The 2<sup>nd</sup> and 3<sup>rd</sup> commands copy 2 files into your home directory:

`.bash_profile`: Contains the "export" command above so it executes automatically every time you log on.

`myspim`: a small script for running a MIPS assembly program (*Cool*'s compiler output) using the *spim* emulator.

### **II. Writing the Program**

Now, we are going to write a simple *Cool* program that displays "Hello World!". Open up emacs or any other text editor and write the following code:

```
class Main {
  out : IO <- new IO;

  main(): Object {
    out.out_string("Hello World!\n")
  };
};
```

Save your file as *hello.cl* when finished.

### **III. Compiling it**

To compile your program, go to the folder where you saved it and enter:

```
coolc -o hello.s hello.cl
```

(the “-o hello.s” can be omitted). This will create a file named *hello.s* which is the MIPS assembly code that resulted from the compilation.

#### IV. Running it

To actually run the program and see its output we are going to use a MIPS emulator called *spim* since the lab’s computers have a different architecture (x86).

Here’s the *myspim* script’s usage:

```
myspim <assembly file>
```

So, for running your compiled Hello World program (*hello.s*), enter the following:

```
myspim hello.s
```

The screen will display *spim*’s initialization messages and then run the program, which will output “Hello World!”

#### V. Sample Program 2

Write the following *Cool* program (*stat.cl*):

```
class Main inherits IO {
  i : Int <- 0;
  number : Int;
  max : Int <- 0;
  sum : Int <- 0;
  maxStr : String;
  avgStr : String;
  conv : A2I <- new A2I;

  main() : Object {
  {
    while (i < 4) loop {
      out_string("Enter an integer: ");
      number <- in_int();
      if (max < number)
        then max <- number
      else 0
      fi;
      i <- i + 1;
      sum <- sum + number;
    } pool;
    maxStr <- conv.i2a(max);
    out_string(("The greatest # was: ".concat(maxStr)).concat("\n" ) );
    avgStr <- conv.i2a(sum / 4);
    out_string(("The average is: ").concat(avgStr)).concat("\n"));
  }
};
};
```

Copy the *atoi.cl* sample program from the *Cool* examples directory:

```
cp ~icom4029/cool/examples/atoi.cl .
```

Compile the program: `coolc -o stat.s atoi.cl stat.cl`

Run it: `myspim stat.s`

#### VI. Closing Notes

There are some sample cool programs at `~icom4029/cool/examples`. Your first programming assignment (PA1) will have you writing a stack machine in cool so you should take a look at the

examples, read the cool manual at least up to Section 11 and get familiar with cool by writing some sample programs of your own.