PLATO

(Programming Language for Abstract Transformation Operations)

Daniel Perlmutter (dap2163) — Project Leader
Yasser Abhoudkhil (ya2282) Joaquín Ruales (jar2262)

Each member of a group must give part of the presentation.

I suggest the following structure:

1) an overview of the language and its motivation,
2) a tutorial introduction to the language including an example or two,
3) a discussion of how the language was implemented, and
4) a summary including lessons learned.

This is a 10-minute presentation, each member doesn't need to say much.

If you use slides, bring them on your own laptop; I don't have a computer for you.

Each time slot is 30 minutes: 10 for presentation, then 20 minutes of questions and a demonstration of your language. Please bring a laptop on which you can do this.

Overview and Motivation

PLATO is a language for manipulating groups, rings, and fields inspired by Matlab, Python, Prolog, Java, and Ocaml

PLATO is

- Strongly typed
- Imperative with Higher Order functions
- Not Object Oriented

Examples

```
main() {
  PRINT primes(100);
}
VECTOR<INTEGER> sieve(VECTOR<INTEGER> v, INTEGER n) {
  if (v[1] ^ 2 <= n) {
    RETURN ([v[1]] @ sieve(v[NOT (v % v[1] = 0)], n));
  } else {
    RETURN v;
  }
}
VECTOR<INTEGER> primes(INTEGER x) {
  RETURN sieve([2 TO x], x);
}
```

```
main() {
  PRINT gcd(18, 12);
}
INTEGER gcd(INTEGER a, INTEGER b) {
  if (b = 0) {
    RETURN a;
  } else {
    RETURN gcd(b, a % b);
  }
}
```

How the Language was Implemented

Used O'Caml to generate Java

Lessons Learned