# CS383 Tutorial 1

## Course Project Introduction

Xiao Jia

# Before we start ...

How many of you have used ...

- Java
- Eclipse
- A functional language, e.g. Lisp / ML / Haskell
- lex / yacc / flex / bison / JFlex / Java CUP

# Introduction

- SimPL (pronounced *simple*)
  - simplified dialect of ML
  - both functional and imperative
- Write an *interpreter* in Java

# Phases

1. Lexical analysis
2. Syntactic analysis

} *provided*

3. Type inference
4. Run!

} *write your own*

# Example

```
let
    addFive = fn x => x + 5
in
    addFive 2
end
```

Functions are first-class values

# Example

```
let
    addFive = fn x => x + 5
in
    addFive 2
end
```

f(x) = x + 5
λx.x+5

Functions are first-class values

# Example

```
let
    add = fn x => fn y => x + y
in
    add 1 2
end
```

Only unary functions

# Example

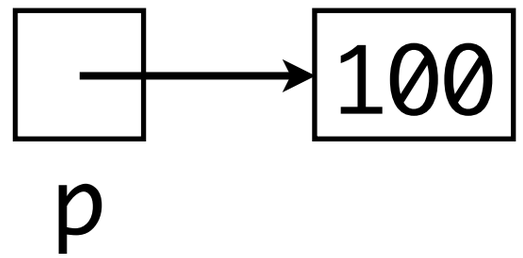
```
let
   factorial = rec f =>
      fn x => if x=1
              then 1
              else x * (f (x - 1))
in
   factorial 4
end
(* this is a comment *)
```

# Example

```
let
    p = ref 100

in
    p := 200;

    !p

end
```

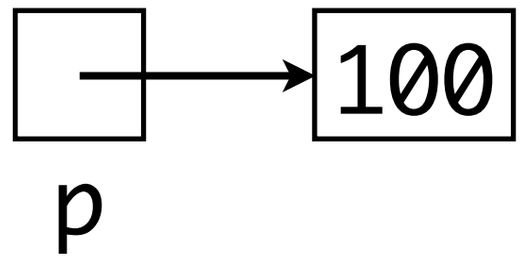
# Example

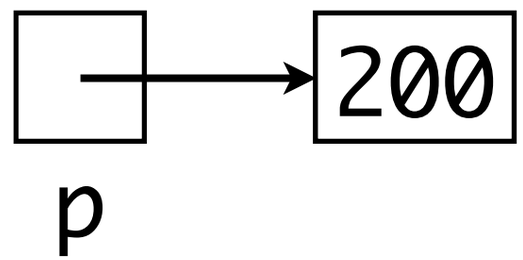
```
let
      p = ref 100
      (*  p = new int(100);  *)
in
      p := 200;

      !p

end
```



# Example

```
let
      p = ref 100
      (*  p = new int(100);  *)
in
      p := 200;
      (*  *p = 200;  *)
      !p
      (*  return *p;  *)
end
```

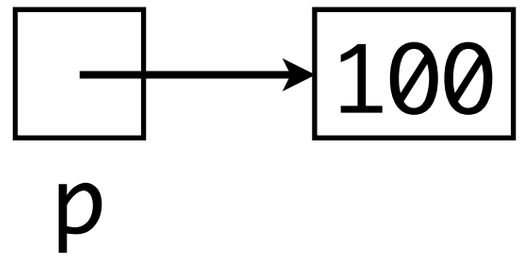


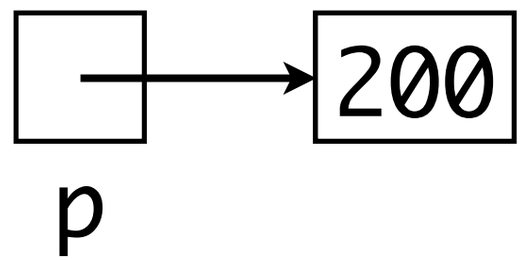
# Example

```
let
      p = ref 100
      (*  p = new int(100);  *)
in
      p := 200;
      (*  *p = 200;  *)
      !p
      (*  return *p;  *)
end
```

reference creation




assignment




dereference

# Example

```
let gcd = fn x => fn y =>
            let a = ref x in
              let b = ref y in
                let c = ref 0 in
                  (while !b <> 0 do
                     c := !a ;
                     a := !b ;
                     b := !c % !b );
                  !a
                end
              end
            end
in  gcd 8 6
end
```

| a | b | c |
|---|---|---|
| 8 | 6 | 0 |
| 8 | 6 | 8 |
| 6 | 6 | 8 |
| 6 | 2 | 8 |
| 6 | 2 | 6 |
| 2 | 2 | 6 |
| 2 | 0 | 6 |

# Example

```
let
   sum = rec sum =>
      fn a => if a = nil
                  then 0
                  else hd a + sum (tl a)
in
   sum (1::2::3::nil)
end
```

# Example

```
(fn p => if (fst p) > (snd p)
          then fst p
          else snd p)
(1,2)
```

# Implementation

- Code in Java 7 and Eclipse
- Submit a runnable JAR file, e.g. SimPL.jar
- java -jar SimPL.jar *program*.spl
- Case time limit: 5000ms
- Your code is executed in a sandbox!

# How to run

```
void run(String filename) {
    try (InputStream inp = new FileInputStream(filename)) {
        Parser parser = new Parser(inp);
        java_cup.runtime.Symbol parseTree = parser.parse();
        Expr program = (Expr) parseTree.value;
        program.typecheck(new DefaultTypeEnv());
        System.out.println(program.eval(new InitialState()));
    }
    catch (SyntaxError e) {
        System.out.println("syntax error");
    }
    catch (TypeError e) {
        System.out.println("type error");
    }
    catch (RuntimeError e) {
        System.out.println("runtime error");
    }
}
```

# Questions?

- Fully understand the specification before you write the code
- Resources are available online at
  http://www.cs.sjtu.edu.cn/~kzhu/cs383/
  http://xiao-jia.com/cs383/
- Send feedback to xjia@cs.sjtu.edu.cn