THE GRAMMAR

```
<program>        ::= <exp>
<exp>:LitExp     ::= <LIT>
<exp>:VarExp     ::= <VAR>
<exp>:PrimAppExp ::= <prim> LPAREN <operands> RPAREN
<exp>:IfExp      ::= IF <exp>test THEN <exp>thenPart ELSE <exp>elsePart
<operands>       **= <exp> +COMMA
<prim>:AddPrim   ::= ADDOP
<prim>:SubPrim   ::= SUBOP
<prim>:Add1Prim  ::= ADD1OP
<prim>:Sub1Prim  ::= SUB1OP

IfExp
%%%
    public String toString() {
        return test + " ? " + thenPart + " : " + elsePart;
    }
    public Val eval( Env env ) {
        return test.eval( env ).isTrue() ?
                                thenPart.eval( env ) : elsePart.eval( env );
    }
%%%

Val
%%%
/**
 * The run-time value of a variable
 */
public class Val {

    /**
     * The actual int value
     */
    public final int value;

    /**
     * Create an int Val.
     */
    public Val( int value ) {
        this.value = value;
    }

    /**
     * Return false if this Val's integer is 0; true otherwise.
     * Useful when an integer is used as a test in a conditional expression.
     */
    public boolean isTrue() {
        return value != 0;
    }

    /**
     * Return the int value as a string.
     */
    @Override
    public String toString() {
        return Integer.toString( value );
    }
}
%%%
```

Feb 7 20:56 2019 V2session.txt Page 1

```
$ rep-t
-->
   0: <program>
if 13 then 6 else 5
   1: | <exp>:IfExp
   1: | | IF "IF"
   1: | | <exp>:LitExp
   1: | | | LIT "13"
   1: | | THEN "THEN"
   1: | | <exp>:LitExp
   1: | | | LIT "6"
   1: | | ELSE "ELSE"
   1: | | <exp>:LitExp
   1: | | | LIT "5"
6
-->
   1: <program>
if -(x,10) then 100 else +(i,v)
   2: | <exp>:IfExp
   2: | | IF "IF"
   2: | | <exp>:PrimAppExp
   2: | | | <prim>:SubPrim
   2: | | | | SUBOP "-"
   2: | | | LPAREN "("
   2: | | | <rands>
   2: | | | | <exp>:VarExp
   2: | | | | | VAR "x"
   2: | | | | COMMA ","
   2: | | | | <exp>:LitExp
   2: | | | | | LIT "10"
   2: | | | RPAREN ")"
   2: | | THEN "THEN"
   2: | | <exp>:LitExp
   2: | | | LIT "100"
   2: | | ELSE "ELSE"
   2: | | <exp>:PrimAppExp
   2: | | | <prim>:AddPrim
   2: | | | | ADDOP "+"
   2: | | | LPAREN "("
   2: | | | <rands>
   2: | | | | <exp>:VarExp
   2: | | | | | VAR "i"
   2: | | | | COMMA ","
   2: | | | | <exp>:VarExp
   2: | | | | | VAR "v"
   2: | | | RPAREN ")"
6
-->
   2: <program>
```