

grammar Brin;

@header

```
{
    import java.util.*;
}
```

@members

```
{
    /** Map variable name to Integer object holding value */

    HashMap memory = new HashMap();
    Scanner scanner= new Scanner(System.in);
    private boolean variableDefined(String name)
    {
        return memory.containsKey(name);
    }
}
```

prog: statment+ ;

statment

```
: expression NEWLINE {System.out.println($expression.value);}
| 'LET' ID '=' expression NEWLINE
    {memory.put($ID.text, new Integer($expression.value));}
| 'PRINT' LITERAL NEWLINE
    { String s = $LITERAL.text; System.out.print(s.substring(1,s.length()-1));}
| 'PRINT' expression NEWLINE {System.out.print($expression.value);}
| 'INTEGER' var(',')* NEWLINE
| 'INPUT' input_ID(',')* NEWLINE
| 'PRINTLN' LITERAL NEWLINE
    { String s = $LITERAL.text; System.out.println(s.substring(1,s.length()-1));}
| 'PRINTLN' expression NEWLINE {System.out.println($expression.value);}
| COMMENT
| 'END' { System.exit(0);}
;
```

expression returns [int value]

```
: e = multExpression {$value = $e.value;}
( '+' e = multExpression {$value += $e.value;}
| '-' e = multExpression {$value -= $e.value;}
)*
;
```

multExpression returns [int value]

```

:          e=atom {$value = $e.value;}
  ( '/' e=atom {$value /= $e.value; }
  | '*' e=atom {$value *= $e.value;}
)*
;

```

atom returns [int value]

```

: INT {$value = Integer.parseInt($INT.text);}
  | ID
  {
      Integer v = (Integer)memory.get($ID.text);
      if ( v!=null ) $value = v.intValue();
      else System.err.println("undefined variable "+$ID.text);
  }
  | '(' expression ')' {$value = $expression.value;}
;

```

input_ID

```

: ID {
    if(variableDefined($ID.text))
    {
        System.out.println("Value "+$ID.text);
        Integer value=scanner.nextInt();
        memory.put($ID.text, new Integer(value));
    }

    else
    {

        System.err.println("Undefined Global Variable:"+$ID.text);

    }
};

```

var

```

: ID
{
    if(!variableDefined($ID.text))
        memory.put($ID.text, new Integer(0));
    else
        System.err.println("Variable already exists:"+$ID.text);
}

```

```

    }
    ;

ID      : ('a'..'z'|'A'..'Z') ('a'..'z'|'A'..'Z'|'0'..'9'|'_'|'!')*;

INT     : '0'..'9'+;

NEWLINE : '\r'? '\n';

WS      : (' '\t')+ {skip();};

LITERAL : ""(AVOID | ~('\|'|'') ) *"";

AVOID : '\\ ('b'|'t'|'n'|'f'|'r'|'\"'|'\\'|'\\');

//STRING: ""( AVOID | ~('\n'|'r'|'t'|'\\'|'') ). *"";

COMMENT : '/' ~('\n'|'r')* '\r'? '\n' { $channel = HIDDEN; };

/*-----SIL_Grammar.java -----*/

```

```

import java.io.IOException;
import org.antlr.runtime.ANTLRFileStream;
import org.antlr.runtime.CommonTokenStream;
import org.antlr.runtime.RecognitionException;

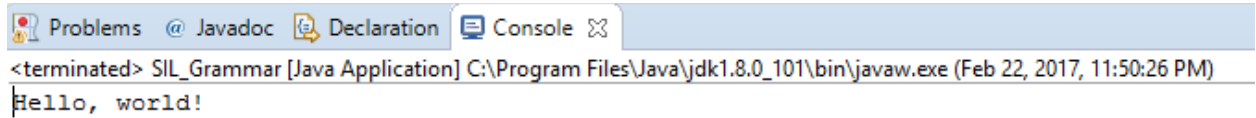
public class SIL_Grammar
{
    public static void main(String[] args) throws IOException, RecognitionException
    {

        BrinLexer lexer = new BrinLexer(new ANTLRFileStream("D:\\CSU LONG BEACH\\Spring
2017\\APL\\Assignment\\Assignment 2\\output.s"));
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        BrinParser parser = new BrinParser(tokens);
        parser.prog();
    }
}

```

```
/* OUTPUT*/
```

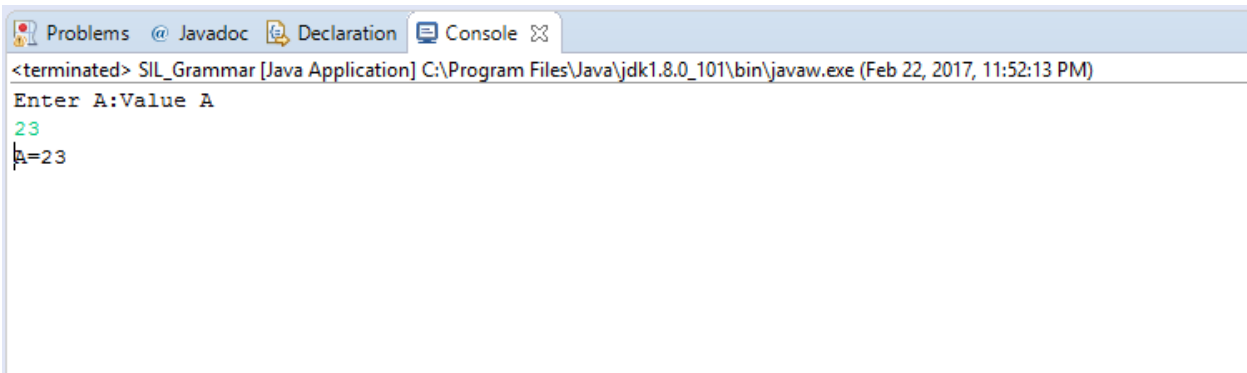
- 1) PRINTLN "Hello, world!"
END



The screenshot shows a Java IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of a Java application. The text in the console is: <terminated> SIL_Grammar [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Feb 22, 2017, 11:50:26 PM) Hello, world! The output "Hello, world!" is on a new line after the header information.

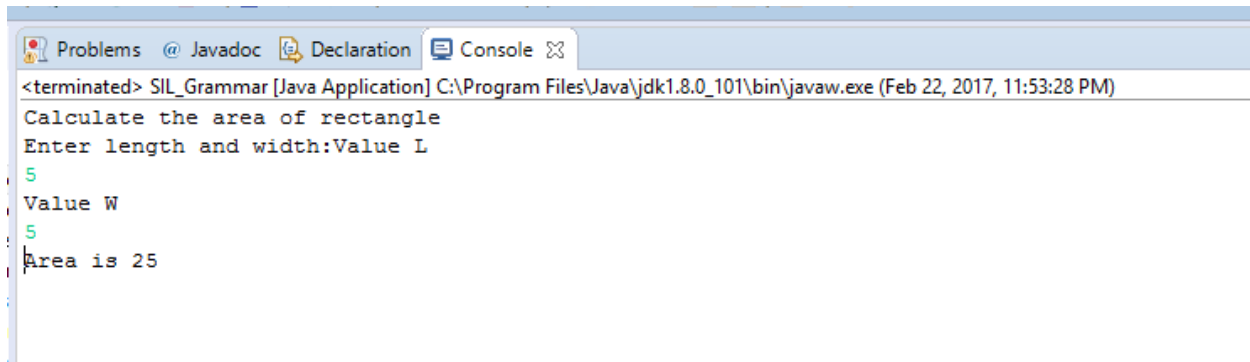
- 2) INTEGER A
PRINT "Enter A:"
INPUT A
PRINT "A="

```
PRINTLN A  
END
```



The screenshot shows a Java IDE window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of a Java application. The text in the console is: <terminated> SIL_Grammar [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Feb 22, 2017, 11:52:13 PM) Enter A:Value A 23 A=23. The input "23" is on a new line after the prompt "Enter A:Value A", and the output "A=23" is on a new line after the prompt "A=".

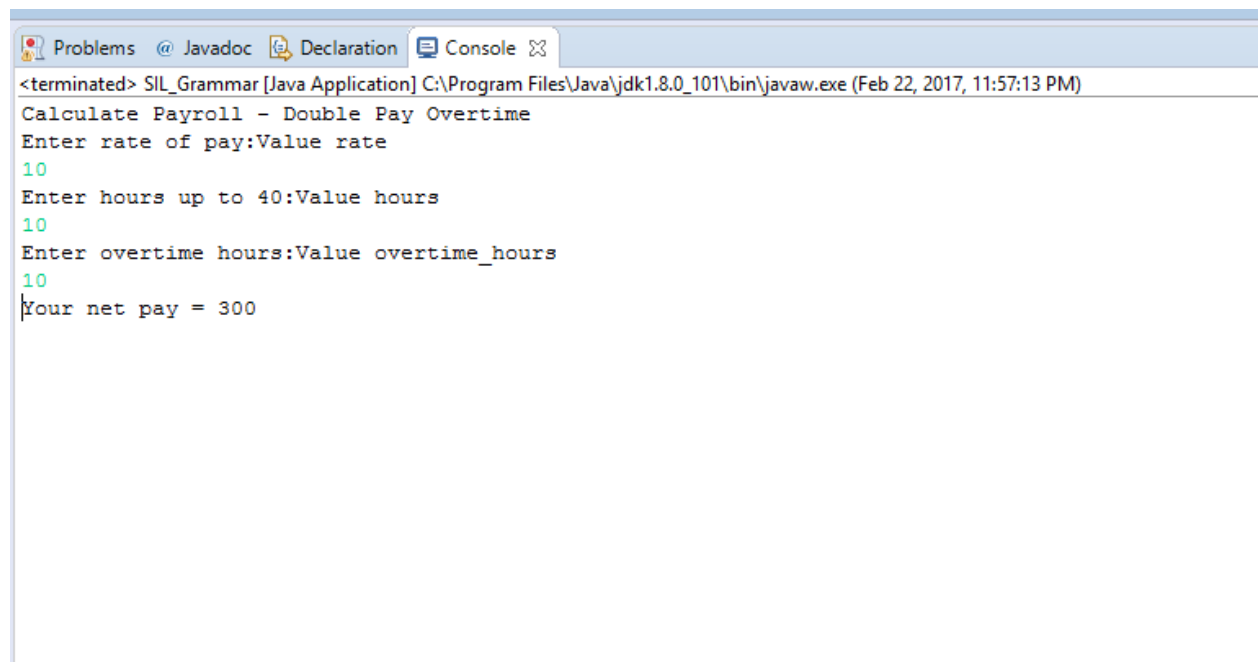
- 3) //This program calculates the area of rectangle
PRINTLN "Calculate the area of rectangle"
INTEGER L, W, AREA
PRINT "Enter length and width:"
INPUT L,W
PRINT "Area is "
LET AREA = L * W
PRINTLN AREA
END



```
<terminated> SIL_Grammar [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Feb 22, 2017, 11:53:28 PM)
Calculate the area of rectangle
Enter length and width:Value L
5
Value W
5
Area is 25
```

4)

```
PRINTLN "Calculate Payroll - Double Pay Overtime"
PRINT "Enter rate of pay:"
INTEGER rate, hours, overtime_hours, netpay
INPUT rate
PRINT "Enter hours up to 40:"
INPUT hours
PRINT "Enter overtime hours:"
INPUT overtime_hours
LET netpay = rate * hours + rate * overtime_hours * 2
PRINT "Your net pay = "
PRINTLN netpay
END
```

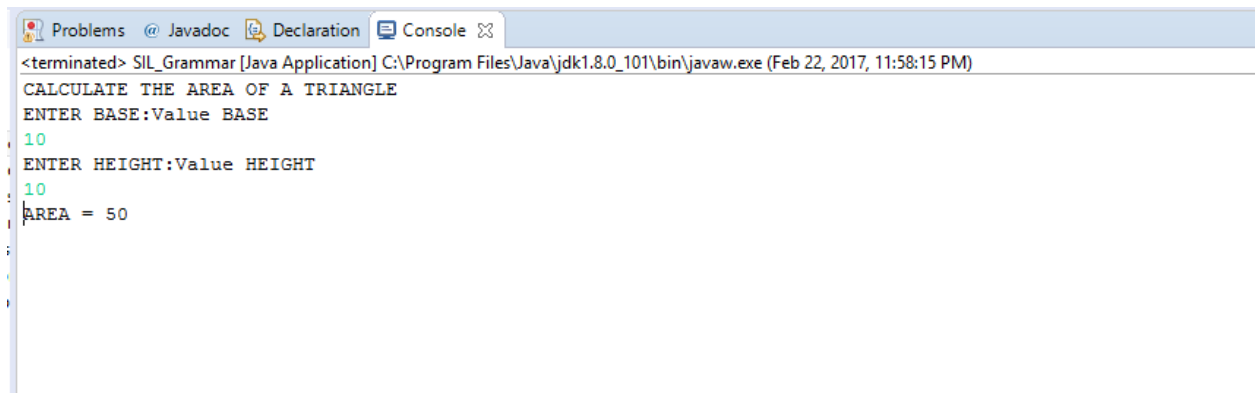


```
<terminated> SIL_Grammar [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Feb 22, 2017, 11:57:13 PM)
Calculate Payroll - Double Pay Overtime
Enter rate of pay:Value rate
10
Enter hours up to 40:Value hours
10
Enter overtime hours:Value overtime_hours
10
Your net pay = 300
```

5)

```
//Calculate the area of a triangle
PRINTLN "CALCULATE THE AREA OF A TRIANGLE"
```

```
PRINT "ENTER BASE:"  
INTEGER BASE, HEIGHT, AREA  
INPUT BASE  
PRINT "ENTER HEIGHT:"  
INPUT HEIGHT  
LET AREA = (BASE * HEIGHT) / 2  
PRINT "AREA = "  
PRINTLN AREA  
END
```



The screenshot shows a Java IDE window with the title bar "Problems @ Javadoc Declaration Console". The console output is as follows:

```
<terminated> SIL_Grammar [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Feb 22, 2017, 11:58:15 PM)  
CALCULATE THE AREA OF A TRIANGLE  
ENTER BASE:Value BASE  
10  
ENTER HEIGHT:Value HEIGHT  
10  
AREA = 50
```