WLanguage

st119978 Gayatri Tirumalasetty
st120000 Prithvi Raju N
st120055 Varun Reddy Mannem

outline

- Sample Language code
- Tokens
- WLanguage Grammar
- Primitive Types
- Boolean Type
- If-then and While Statements
- Composite Types
- Functions

Features:

Variable declaration
If-then
While
Arrays
Functions with return type

Sample code

```
int d,a := 5;
string s := 'factorial';
d := 1;

while(a>1) do {
    d := d*a;
    a := a-1;
}

see s;
see d;
```

tokens

":"
";"
","
"+"
"-"
"*"
"/"
"%"
"("
")"
":="
"."
"?" if then
"|" else
"?;" end if
"see" print
"while"
"do"
"{"
"}"

"boolean"
"int"
"float"
"string"
"||"
"&&"
"<"
">"
">="
"<="
"=="
"!="
"void"
"return"
"struct"

Grammar

```
program       -> declaration_list statement_list
                 |
                 declaration_list function_list statement_list


functiondecl  -> VOID ID LPAREN RPAREN BEGIN statement_list   END
                 |
                 type ID LPAREN RPAREN BEGIN       statement_list RETURN expr SEMI END
```

```
statement    -> assignment SEMI
              |
                   ifthen
              |
                   print SEMI
              |
                   while
              |
                   BEGIN statement_list END
              |
                   functioncall SEMI
```

```
declaration -> STRUCT ID:i LPAREN struct_values:sv RPAREN SEMI
                |
               type id_list SEMI
               |
              type:t ID COLON INTEGER_LITERAL EQ array_vals SEMI

assignment -> ID EQ expr
              |
              ID EQ ID LPAREN RPAREN
              |
               ID COLON expr:index COLON EQ expr:e
```

```
functioncall  -> ID LPAREN VOID RPAREN

while    -> WHILE LPAREN expr RPAREN DO statement

print    -> PRINT expr

ifthen   -> IF LPAREN expr RPAREN statement ENDIF
                |
            IF LPAREN expr RPAREN statement ELSE  statement ENDIF
```

Composite Types

Arrays:

Initialization:

int arrlist : 4 := 6,7,8,9;

Assignment:

arrlist :3: := 0;
see arrlist ;

Output:

6,7,0,9

Functions

```
int x , a := 5 ;
int d , y;
int func() {
        x := a;
        x := x + 1;
        return x+a;
}

d := f() ;
see d;
```

Output:

11