

General

- functions + wain
- return
 - always int
 - appear once at end of function

declarations

- Two types: int or int*
 - either as **unsigned integer** or **NULL**
 - must be initialized
 - dcls must precede all statements
- no global variables

wain

- must be last function
- two parameters
 - 1st: int or int*
 - 2nd: int

Statements

- **if**
 - must have an else clause
 - {} needed
- **while**
 - {} needed
- **println**
- **=** (*assignment*)
- **delete []**
- **new**

Conditions

- && and || do not exist

Expressions

May contain only:

- variable names
- integers (without a sign)
- function calls
- **new**
- unary & and *
- +, -, *, /, %, ==, !=, <=, >=, <, >

Arrays

- There are no "arrays" in WLP4, but pointers can be dereferenced to achieve them
 - elements can be accessed only using pointer dereferences

Example

```
int wain(int* begin, int size)
{
    int* array_copy = NULL;
    int i = 0;
    array_copy = new int[size];

    while(i < size)
    {
        *(array_copy + i) = *(begin + i);
        i = i + 1;
    }

    delete [] array_copy;

    return 0;
}
```

Context-free Syntax

CFG

Terminals	INT, DELETE, LPAREN, PLUS, ID, ...
Non-Terminals	procedures, procedure, main, params, paramlist, type, dcl, dcls, statements, lvalue, expr, statement, test, term, factor, arglist
Start Symbol	procedures
Production Rules	procedures → procedure procedures procedures → wain ...

Context-sensitive Syntax

Procedures

A procedure is said to be **declared** from the first occurrence of its name in the string that makes up that procedure.

A procedure cannot be called until it has been **declared**

Can	Can not
<ul style="list-style-type: none">• call itself recursively• call procedures declared before itself	<ul style="list-style-type: none">• call procedures declared after itself.• Two procedures may not have the same name.• wain may not call itself recursively.

ID's

- ID derived from dcl within a procedure p is said to be **declared in p**.
- ID derived from factor or lvalue within p is said to be **used in p**
- ID may be declared at most once within a given procedure.
- The same ID may be declared in different procedures.
- May have the same name as a procedure. If an ID x is declared in a procedure p, all occurrences of x within p refer to the ID x, even if a procedure named x has been declared.

TYPES

Each of these tokens have a 'type':

- NUM
- NULL
- factor
- term
- expr
- lvalue