

Lecture 6

Expressions and Assignment Statements

1. Operator precedence

	FORTRAN	PASCAL	C	Ada
highest	**	* / div mod	postfix ++ --	** abs
to	* /	all + -	prefix ++ --	* / mod
lowest	all + -		unary + - * / %	unary + - binary + -

2. Associativity (What happens left to right?)

RULES

FORTRAN	Left * / + - Right **
Pascal	Left all
C	Left post fix ++ -- * / % and binary + - Right prefix ++ -- unary + -
C++	Left * / % binary + - Right ++ -- unary + -
Ada	Left all except ** which is non-associative

3. Parenthesis

4. Conditional expressions

C, C++, and Java: eg. average = (count == 0) ? 0 : sum/count;

5. Side effects

Example: Suppose the function fun changes the value of its parameter: how is the value of $b = a + \text{fun}(a)$ determined?

Example:

```
int a = 5;
int fun1() {
    a = 17;
    return 3;
}
void fun2() {
    a = a + fun1();
}
void main() {
```

```
        fun2();  
    }
```

"order of operations problem"

Pascal and Ada: Any order

Java: left to right;

6. Overloaded Operators

Discuss:

7. Type Conversions

type coercion

type assumption: Java

```
int a, b, c;
```

```
float d;
```

```
a = b*d; forced conversion to integer
```

Explicit Conversion

```
C: a = float(sum)/count;
```

8. Relational and Boolean Expressions

Normal precedence (and before or) except in Ada

C: (a > 7 && c < 10)

Pascal (a > 7) and (c < 10)

9. Short-Circuiting

Define:

C: Always

Ada: Forced (use of **and then** and **or else**)

Pascal: Not standardized

Searching an array for an integer value called **item**:

C++:

```
int arr[10];
```

```
int j = 0;
```

```
while ( j < 10 && arr[j] != item )
```

```
    j++
```

```
if (j < 10)
```

```
    cout << "found";
```

```
else
    cout << "not found";
```

Pascal:

```
arr : array[0..9] of integer;
j : integer;
done : boolean;
j := 0;
done := false;
while not done do
begin
    if j > 9 then
        done := true
    else
        if arr[j] = item then
            done := true
        else
            j = j + 1
    end;
if j < 10 then
    write( 'found' )
else
    write( 'not found' )
```

10. Assignments

multiple assignments `a = b = 0;`

unary assignments: eg. `a += 12;`

as expressions:

```
while (( ch = getchar() ) != EOF) { ... }
```

```
for ( sum = 0; sum < 30; cin >> value, sum += ( value>0 ) ? value : 1 );
```

Note confusion of `=` with `==`