Introduction

Operators

Operators

- We have to operate the variables and constants.
- Many language has operators in the form of keywords.
- Operators in c are mostly made of signs that are not part of alphabet but sign is available on keyboard. That why operators are short.
- Ex a=5;
- Left of assignment operator is called LVALUE and right of assignment operator is called right value.

- Lavlue is a variable whereas rvalue can be either a constant or a variable.
- Most important rule of assignation is right to left value.
- A=b;
- We are assigning the value of b to A at the moment only, later change of b will not effect the new value of A.

- Arithmetic operators
- +,-,*,/%(modulo)
- A=11%3
- A=2
- Compound assignment(+=,-=, *=,/=,%=,<<=,|=)
- A+=5
- A=A+5

- Increase and decrease operators
- C++ can be read as c=c+1;
- Or c+=1

- Realational or equality operator
- ==
- !=
- >
- <
- >=
- <=

Logical operators

- !(5==5) is a false
- Logical&& and || are used when evaluating two expressions to obtain a single expression.

Conditional operator?

- Condition ? Result1 :result2
- 7==5 ? 4:3 ans 3

Bitwise operator(&|^~<<>>

- &and
- | or
- ^ Xor
- ~not
- << SHL
- >> SHR

Explicit type casting operator.

- Type casting operator allow you to convert a data type to another data type.
- Int i;
- Float f =3.14;
- i=(int)f;
- (int) is the type casting operator.
- Another method
- i= int(f);
- Int(f) is the functional notation.
- Sizeof()
- It returns the size of data type.

Declarations (mdu 08)

- Purpose of declaration
- 1. Choice of storage representation.
- 2.Storage management.
- Polymorphism
- 4. type checking.

Type checking(mdu 06,07,08,05)

- Type checking means that each operation executed by a program recieves the proper number of arguments of proper data type.
- For unary operation one arg is mandatory.
- For b
- Binary operation two args are mandatory.
- Type checking can be run time(dynamic)
- Or compile time.

- Advantages of dynamic time binding is flexiblity
- Disadvantage of dynamic time binding is difficult to debug.
- Extra storage is required.

Strongly typed language.

If we detect all types of errors statically in a program than language

is strongly typed.

Type conversion and coercion

- If during type checking , a mismatch occurs between actual type of args and expected
- Then it may flag as an error.
- A coercion (implicit type conversion)can be applied to change the type of actual argument to correct type.
- Language provides type conversion in
- Built in function. Ex round(3.14) converts to integer. (Int)x converts to in integer type.
- As coercion;- if the args for an arithmetic operation such as + are of int,real int is automatically converted to float.

Examples in Pascal:

```
var A: real;
B: integer;
```

A := B - Implicit, called a coersion - an automatic conversion from one type to another

A := B is called a widening since the type of A has more values than B.

B := A (if it were allowed) would be called a narrowing since B has fewer values than A. Information could be lost in this case.

In most languages widening coersions are usually allowed;

narrowing coersions must be explicit:

```
B := round(A); Go to integer nearest A
B := trunc(A); Delete fractional part of A
```