Day 4 Statements, Expressions, and Operators

statement- complete instruction that directs the computer to carry out some task

C compiler is not sensitive to whitespace

c compiler does care about whitespace in strings

use \ to break literal string over multiple lines

printf(“Hello,\

world¡‘);

place a ; on line by itself creates a null statement

block {} = **compound statement**

expression- anything that evaluates to a numeric value

simplest expression consists of a single item- variable, constant etc.

complex expression- simpler expressions connected by an operator

when an expression contains multiple operators, the evaluation of the expression depends on operator precedence

an assignment statement is itself an expression. assignment statements should not be nested with other expressions

assignment statement- variable = expression;

expression is evaluated and result is assigned to variable

unary mathematical operators

|increment| ++ |increments the operand by one\ |decrement| – | decrements operand by one

++x same as x = x + 1;

prefix mode ++x modify operand before used in enclosing statement

postfix mode x++ modify operand after used in enclosing statement

binary mathematical operators- + - * / %

use () to modify evaluation order

relational operators- == > < >= <= != (not equal. if not equal returns a 1) == and != lower in precedence than others

relaional operators used to construct **program control statements**- modifies order of statement execution

if statement- evaluates an expression and directs program execution depending on the result of that evaluation

`evaluates as true, executed. evaluated as false not executed`

if (expression) { statement1; statement2; else if (expression 2) statement 3; else statement 4; }

only need {} if more than one statement ie creating a block

will return 1 or 0. any nonzero number will be evaluated as true

don't use the != in an if statement containing an else

logical operators- AND &&, OR ||, NOT ! (true if expression is false)

compound assignment operators- combines binary mathematical operation with assignment operaion +=, -=, /=, *=, %=. ex: x += 5 - will increase x by 5 and assign that value to x

conditional operator- only ternary operator. exp1 ? exp2 : exp3 . if exp1 evaluates to be true, then entire expression evaluates to value of exp2. if exp1 evaluates to be false entire expression evaluates to value of exp3.

comma operator- punctuation but also split into 2 sub expressions- both expressions evaluated, left expression evaluated first, entire expression evaluates to value of the right expression

`x = (a++ , b++) - assigns value b to x, then increments a then increments b`

if use single compound if statement expressions evaluated until entire statement evaluates to be false