Section C

Sequence Control

INDRODUCTION

1. CONTROL STRUCTURE

2. TYPES OF SEQUENCE CONTROL

3. SEQUENCING WITH ARITHMETIC EXPRESSION

Control Structure:

in programming language provides the basic frame work within which operations and data are combined in to program and set of program. This involves two aspects:

1. The control of the order of execution of operations , both primitive and user defined , that term as the **"Sequence Control"**
2. The control of transmission of data among the subprograms of a program which we term as "**Data Control"**

Basically Sequence Control may be classified as :

A. Structure used in expression such as precedence rule and parenthesis

B. Structure used between statements or groups of statements such as conditional and iteration statements

C. Structure b/w Subprograms

Types of Sequence Control:

- **Implicit(Default) Sequence Control:**

Are those defined by the language to be effect unless modified by the programmer through some explicit structure. For example there is language defined hierarchy of operation that control the order of the operation in expression when parenthesis is absent

- **Explicit Sequence Control:**

In it the control is defined by the programmer himself like use of **Go To** statements that transfer the control from one location to the another as per requirement

Implicit Control	Explicit Control

int a,b
a=2;
a=3;
C=a+b;
printf(“%d”,c);
	int a,b;
	a=2;
	b = 3;
	go to 1
	c= a+b;
	1:
	printf((“%d: ,c);

1. Expression:

Expression is a basic building block for statements of a program. An expression is a combination of a variable , constant and operator like in C language expression evaluates to a value.

2. Statements:

A program consists of a multiple statement. There are two types of statements:

- **Basic Statements**
- **Compound Statements**

3. Declarative Statements

This is execution model. It covers the logic programming like , Prolog

4. Subprogram:

For structure programming, program is divided in to small section and each section is called a subprogram

1.Sequencing with Arithmetic Expression:

***Expression are evaluated using an assignment statement of form

Variable = expression;

Variable may be any valid variable of a particular language. When the statement is encountered the expression is evaluated first and then replaces the previous value of the left hand side.

For example

x= a * b – c

y = b/c * a

z = a-b/c+d

Precedence in Arithmetic Operator:

An arithmetic expression without parenthesis will be evaluated from left to right using the rules of precedence of operator. There may be two levels:

Sequence Control for expression:

- ✓**High Priority like * / %**
- ✓**Low Priority like + -**

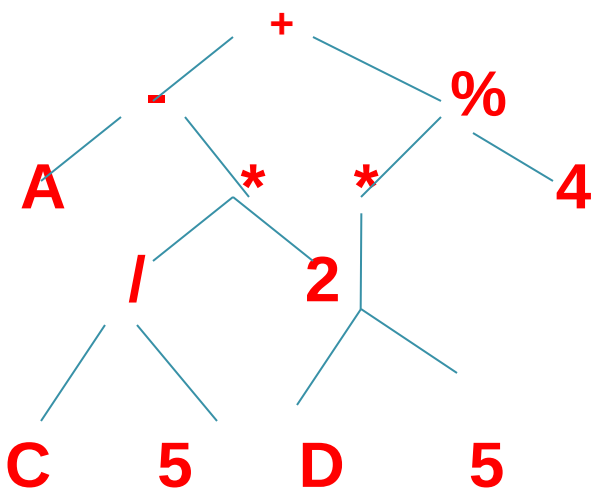
Rules for Evaluation of Expression:

- First parenthesized sub expression left to right are evaluated.
- If parenthesis are nested , the evaluation begins with innermost sub expression
- The precedence rule is applied in determined the order of application of operators in evaluating sub expression
- The associability rule is applied when two or more operator of the same precedence level appear in the sub expression
- Arithmetic expression are evaluated from left to right
- When parenthesis are used, the expression within parenthesis assume highest priority

Arithmetic Expression Tree:

In a arithmetic expression tree, leaves are used to represent operands and root is used for main operation of expreesion.The nodes between root and leaves are used intermediate operation.

For example :the given expression can be represented by tree below: A-(C/5 * 2) + (D * 5 % 4)



Precedence-Table for Operator:

Order	Category	Operator	Operation	Associativity
1.	Highest precedence	( ) [ ] ? : : .	Function call	L → R Left to Right
2.	Unary	! ~ + - ++ - - & * Size of	Logical negation (NOT) Bitwise 1's complement Unary plus Unary minus Pre or post increment Pre or post decrement Address Indirection Size of operant in bytes	R → L Right → Left
3.	Member Access	* →*	Dereference Dereference	L → R
4.	Multiplication	* / %	Multiply Divide Modulus	L → R
5.	Additive	+ -	Binary Plus Binary Minus	L → R
6.	Shift	<< >>	Shift Left Shift Right	L → R
7.	Relational	< <= > >=	Less than Less than or equal to Greater than Greater than or equal to	L → R
8.	Equality	== !=	Equal to Not Equal to	L → R
9	Bitwise AAND	&	Bitwise AND	L → R
10	Bitwise XOR	^	Bitwise XOR	L → R
11	Bitwise OR	|	Bitwise OR	L → R
12	Logical AND	&&	Logical AND	L → R
14	Conditional	? :	Ternary Operator	R → L
15	Assignment	= *= %= /= += -= &= ^= |= <<= >>=	Assignment Assign product Assign reminder Assign quotient Assign sum Assign difference Assign bitwise AND Assign bitwise XOR Assign bitwise OR Assign left shift Assign right shift	R → L
16	Comma	,	Evaluate	L → R

Unit - 4

Associativity: that defines if the sub expression containing the multiple occurrences of operator are grouped from either left to right or right to left

Assignment

- What is sequence control
- Difference between sequence and data control.
- Explain different type of sequence control.
- Define Sequence control in expression.