Loops in C++

Loops

- Loops are used to perform iterative task.
- Types of Loops
  - Counter Controlled Loops
  - Sentinel Controlled Loops
  - Entry Controlled Loops
  - Exit Controlled Loops

Loops

- Counter Controlled loops:
  - No of iterations are known before execution.
  - Control variable is used for counting.
  - Also called definite iteration loops.
  - Ex. Any loop with counter.

- Sentinel Controlled loops:
  - No of iterations depends on special value called sentinel value.
  - Sentinel value may be read from user.
  - No of iteration are indefinite or can not be governed before execution.
  - Also called odd loops.
  - Ex. Any loop with depend on user i/p like ch=getchar().

Loops

- Entry Controlled loops:
  - Condition is tested before (at entry time)execution of loop if satisfied body gets executed.
  - Also known as pre-test loops.
  - Ex. Any loop with entry condition.

- Exit Controlled loops:
  - Condition is tested at the end of body means before loop exits.
  - Loop body executes at least once.
  - Also known as post-test loops.
  - Used for menu driven programming.
  - Ex. Only do while loop.

While Loop

- Syntax

```
Initial Expression;
while(Loop condition)
{
      Loop statements.
      Loop expression.
}
```

1> If condition becomes true then loop executes otherwise exits.

2> If loop executes it modifies loop counter and again test condition.

3>Step 2 continues until condition becomes false.

Ex. Print nos, calculate fact ,print series ,print sum of series

While Loop

- Entry controlled loop.
- Empty condition is compile time error.
- A condition can be any expression using logical and relational operator or variable which result to true or false.
- If condition is not satisfied loop never executes.
- Generally it is used as sentinel controlled loop.

for Loop

- Syntax

```
for(Initial Expression ; Loop condition ; Loop expression)
{
      Loop statements.
}
```

1> If condition becomes true then loop executes otherwise exits.

2> If loop executes it modifies loop counter and again test condition.

3>Step 2 continues until condition becomes false.

4>Entry controlled loop.

5>only semicolons are compulsory.

for Loop

- Better than while as monitoring is easy as all the three controllers at one place.
- Semicolon after loop is ends it.
- Can have multiple initializations.
- Can not have multiple conditions.
- Compound condition using comma can be used.
- Can have any valid C statement in place of any of three.
- Generally used for counter controlled loop.
- Ex. Print nos, calculate fact ,print series ,print sum of series, odds in range, Squares in range

Do While Loop

- Syntax

```
Initial Expression;
do
{
        Loop statements.
        Loop expression.
} while(Loop condition);
```

1> Exit controlled loop.

2>Executes at least once.

3>used for menu driven programming.

Ex. Print no in reverse.

Nested Loops

- Loop inside loop.
- Used for 2d data processing.
- ex. Factorial series.
- Patterns using ****
- Used for array sorting.

break and continue statement

- break can be used to stop loop execution at ny point of time.
- break is applied to current loop in nested loops.
- continue can be used to skip special iteration.
- continue continues loop provided condition is true.
- continue is applied to current loop in nested loops.
- Due to continue control jumps to condition test.

Thank You

Use us for repetitive tasks!!!!

……….Loops