C Programming

Trainer : Smita Kadam

Email ID : smita@sunbeaminfo.com

Loops - Iteration

- Helps to give execution control repeatedly to a specific block

1. On Entry Check Loop
    1. While
    2. For

2. On Exit Check Loop
    1. do…while

- **Please note in all loops available in C language execution control will enter inside loop block only when given entry/exit given expression will result true.**
- While writing loop program should focus on
    - Initial state
    - Expression Check
    - Modification Statement – is a statement which helps to result expression to false state
    - Absence of modification statement will result into infinite loop

On Entry Check - while

```
while (<expression>) // on entry check
{
    <statements>
     ……
}//calls block repeatedly
```

On Entry Check - for

```
for (<initial statement>;<expression>;<modification statement>)
{
        <statements>
    ……
}//calls block repeatedly
```

On Exit Check – do….while

```
do
{
        <statements>
     ……
}while(<expression>);//calls block repeatedly
```

On Entry Check Vs On Exit Check

On Entry Check

```
int n = 5;
While(n<=3)
{
        printf("%d",n);
}
```

Execution control will never entered inside loop as initial state is not related to on entry expression check.

On Exit Check

```
int n = 5;
do
{
        printf("%d",n);
}while(n<=3);
```

At least one execution is fixed no matter what is initial state of expression.

Jump Statements – break, continue, return, goto

break

Can be used inside switch/loop.

Helps to move execution control forcefully outside switch/loop

continue

Can be used only inside loop.

Helps to move execution control forcefully to next iteration.

Skips the execution of statements below continue.

return

Can be used inside function.

Helps to move execution control forcefully back to calling function.

goto

<label>:

    <statements>

goto <label> ;

Helps to move execution control forcefully to a specific label definition.

Thank you!