**Looping**

Syntax

1. Initialize variables
2. Condition check
3. Statements to be executed
4. Counter

There are three types of loops in C language

1. Do while
2. While
3. For

There are two categories of loops.

1. Entry controlled : The statements get executed after checking the condition and if it is true.
2. Exit controlled : The statements gets executed without checking the condition for the first time itself.
   1. Syntax :

do {

}while(cond);

Only in do while the while statement has the semi colon.

Code :

* #include<stdio.h>

int main()

{

char taste = 'g';

int emptyPlate = 10;

int sFull = 0;

do

{

printf("\nHave a bite of food\n");

emptyPlate--;

printf("\nStomach Full (1/0) : ");

scanf("%d", &sFull);

printf("\nHow was the taste (g/b) : ");

scanf(" ");

taste = getchar();

printf("\nEmpty plate value : %d",emptyPlate);

}while(taste=='g') &&( sFull==0) && (emptyPlate<=0);

printf("\nProgram ended\n");

return 0;

}

* #include <stdio.h>

#include <stdlib.o>

#define SUCCESS 0

#define FAILURE -1

#define True 1

#define False 0

int main()

{

int ch;

int flag = 0;

while(True)

{

flag = 0;

switch(displayMenu())

{

case 1 :

printf("\nAddition function Executed\n");

break;

case 2 :

printf("\nSubraction function Executed\n");

break;

case 3 :

printf("\nMul function Executed\n");

break;

case 4 :

printf("\nDivision function Executed\n");

break;

case 5 :

printf("\nExiting from the application\n");

//exit(SUCCESS);

flag = 1;

break;

default :

printf("\nEnter the correct choice\n");

break;

}

if(flag==1)

break;

}

printf("\nProgram Ended\n");

return SUCCESS;

{

int choice;

printf("\nPress,");

printf("\n1. Addition");

printf("\n2. Subraction");

printf("\n3. Mul");

printf("\n4. Division");

printf("\n5. Exit");

printf("\nChoice");

scanf("%d", &choice);

return choice;

}

* As the scanf has not happened correctly it gives a infinite loop.

**FOR LOOP :**

Syntax :

for(initial.section;cond section;counter)

{

sts;

}

* For loop is different from while and do while because :
  1. initialize more than one variable
  2. conditions can be more than one
  3. Counters can be more which run parallelly
  4. We can omit one or more sections
  5. Conditions can also be omitted.

CODE :

#include <stdio.h>

int main()

{

int i,j;

for(i=0,j=2;(i<10)&&(j<