CONTROL STRUCTURES IN C

ITERATIONS

- If we want to execute a set of statements repeatedly, we use iterative structures also called as loops.
- There are three types of loops in C:
  - for
  - while
  - do..while

while AND do..while LOOP

- Syntax of while:

```
while(condition)
{
//statements
}
```

- Syntax of do...while:

```
do
{
//statements
} while(condition);
```

WHILE LOOP EXAMPLE

```
#include <stdio.h>
int main()
{
int count=1;
while (count <= 4)
{
printf("%d ", count);
count++;
}
return 0;
}
```

Output:

```
1 2 3 4
```

do.. WHILE LOOP EXAMPLE

```
#include <stdio.h>
int main()
{
int count=1;
do
{
printf("%d ", count);
count++;
} while (count <= 4);
return 0;
}
```

Output:

```
1 2 3 4
```

COMPARISON OF ALL THREE LOOPS

for	while	do..while

Syntax:
for(initialization;condition;updation)
{
}
	Syntax:
	while(condition)
	{
	}
		Syntax:
		do
		{
		} while(condition);

It is a pre tested loop	It is a pre tested loop	It is a post tested loop
If the condition is false at the entry level, the loop won’t execute even once.	If the condition is false at the entry level, the loop won’t execute even once.	If the condition is false at the entry level, the loop WILL execute ATLEAST once.
Used for problems where we know the *count of how many times* loop has to execute.	Used for problems where we DO NOT know the *count of how many times* loop has to execute.	Used for problems where we DO NOT know the *count of how many times* loop has to execute.
Example: Calculate sum of first “n” numbers	**Example:** Count the number of digits in a given number	**Example:** ATM transactions

OPERATING WITH DIGITS OF A NUMBER

1. Input number
2. Check if number is 0
3. Extract the last digit i.e unit’s place by doing number % 10.
4. Update the number to store only the remaining digits by doing number/10
5. Go to step 2.

Can you guess which loop it is????

QUICK EXERCISE

- Count the number of digits in a user entered number.
- Given a number, create and print another number which is the reverse.
- Input digits from 0-9 and print them in words. Do the above continuously until the user presses a 0

NESTED LOOPS

- Any loop can be placed inside another loop.
- For each cycle/iteration of the outer loop, inner loop executes once completely.
- Total iterations = Number of iterations of outer loop * Number of iterations of inner loop

QUICK EXERCISE

- Predict the output:

```
for( i=1;i<=5;i++)
{
for(j=4;j>=1; j--)
{
printf(“i = %d  j = %d\n”,i,j);
}
}
```

<u>Output</u>

```
i = 1  j = 4
i = 1  j = 3
i = 1  j = 2
i = 1  j = 1
i = 2  j = 4
i = 2  j = 3
.....
i = 5  j = 4
i = 5  j = 3
i = 5  j = 2
i = 5  j = 1
```

QUICK EXERCISE

- Predict the output:

```
for( i=1;i<=4;i++)
{
for(j=1;j<i; j++)
{
printf(“i = %d  j = %d\n”,i,j);
}
}
```

Output:

```
i = 2  j = 1
i = 3  j = 1
i = 3  j = 2
i = 4  j = 1
i = 4  j = 2
i = 4  j = 3
```