Iteration Statements in Programming

Iteration statements, commonly known as loops, are statements in programming used to execute part of code repeatedly based on condition or set of conditions. These constructs are important for performing repetitive tasks efficiently. In this article, we will discuss various types of iteration statements and their use in different programming languages.

Types of Iteration Statements in programming:

There are mainly three types of iteration statements:

- For Loop
- While Loop
- Do-While Loop

1. For Loop:

For loops iterate over a range of values or elements in a collection. These are mainly used where the number of iterations is known beforehand like iterating through elements in an array or predefined range.

Syntax

```
for i in range(5):
    print(i)
```

2. While Loops

While loops execute as long as specifies condition evaluates to true. These are suitable for situations where the number of iterations is not known and depends on dynamic conditions.

Syntax

```
count = 0
while count < 5:
    print(count)
    count += 1
```

3. Do-While Loops

Do while loops are similar to while loop but guarantee at least one execution of the loop body before checking the condition. These are useful when loop must execute at least once regardless of the condition.

Syntax

```
int count = 0;
do {
   cout << count << endl;
   count++;
} while (count < 5);
```

Example program for looping statements:

```
#include <stdio.h>

int main()
{
   // For Loop
   for (int i = 0; i < 5; i++) {
      printf("%d ", i);
   }
   printf("\n");

   // While Loop
   int count = 0;
   while (count < 5) {
      printf("%d ", count);
      count++;
   }
   printf("\n");
```

```
    // Do-While Loop
    count = 0;
    do {
        printf("%d ", count);
        count++;
    } while (count < 5);
    printf("\n");

    return 0;
}
```

Output

```
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
```

For example, look at the following code which prints first 10 numbers using a while loop.

```
#include <stdio.h>
int main()
{
int i = 1; // initialize loop variable
while (i<=10) // test the condition
{ // execute the loop statements
printf("%d", i);
i = i + 1; // condition updated
}
getch();
return 0;
}
```

Output

```
1 2 3 4 5 6 7 8 9 10
```

Write a program to calculate the sum of first 10 numbers.

```
#include <stdio.h>
int main()
{
int i = 1, sum = 0;
while (i<=10)
{
sum = sum + i;
i = i + 1; // condition updated
}
printf("\n SUM = %d", sum);
return 0;
}
```

Output

```
SUM = 55
```

Write a program to print 20 horizontal asterisks(*).

```
#include <stdio.h>
int main()
{
```

```
int i=1;
while (i<=20)
{
printf("*") ;
i++;
}
return 0;
}
```

Output

```
****************************
```

Write a program to calculate the sum of numbers from m to n.

```
#include <stdio.h>
int main()
{
int n, m, sum =0;
clrscr();
printf("\n Enter the value of m: ");
scanf("%d", &m);
printf("\n Enter the value of n: ");
scanf("%d", &n);
while (m<=n)
{
sum = sum + m;
m = m + 1;
printf("\n SUM = %d", sum);
return 0;
}
```

Output

```
Enter if- value of m: 7
Enter the value of n: 11
SUM = 45
```

Do-while: **Programming Tip:** ***Do not forget to place a semicolon at the end of the do-while statement.***

```
#include <stdio.h>
int main()
{
int i = 1;
do
{
printf("\n %d", i);
i =  i + 1;
} while (i<=10);
return 0;
}
```

Write a program to calculate the average of first n numbers.

```
#include <stdio.h>
int main()
{
int n, i = 1, sum = 0;
float avg 0.0;
printf("\n Enter the value of n: ");
scanf("%d", &n);
do
{
sum = sum + i;
i =  i + 1;
} while (i<=n);
avg = (float) sum/n;
printf("\n The sum of first %d numbers = %d", n, sum);
printf("\n The average of first %d numbers %f", n, avg);
return 0;
}
```

Output

```
Enter the value of n: 18
The sum of first 18 numbers = 171
The average of first 18 numbers = 9.00
```

Write a program using a do-while loop to display the square and cube of first n natural numbers.

```
 #include <stdio.h>
#include <conio.h>
int main()
{
int i,n;
clrscr();
printf("\n Enter the value of n: ");
scanf("%d", &n);
printf("\n----------------------------------");
i=1;
do
{
printf("\n \t%d \t | \t%d \t \t %ld \t |", i, i*i, i*i*i);
i++;
}while(i<=n) ;
 printf("\n--------------------------");
return 0;
}
```

Output

```
Enter the value of n: 5
-------------------------
|1| |1| |1|
|2| |4| |8|
|3| |9| |27|
|4| |16| |64|
|5| |25| |125|
--------------------------
```

Write a program to list all the leap years from 1900 to 2100.

```
#include <stdio.h>
#include <conio.h>
int main()
{
int m=1900, n=2100;
clrscr();
do
{
if (m%4 ==0)
printf("\n %d is a leap year", m);
else
printf("\n %d is not a leap year", m);
m = m+1;
} while (m<=n);
return 0;
}
```

For Loop:

Look at the following code which prints the first n numbers using a *for* loop.

```
#include <stdio.h>
int main()
{
int i, n;
printf("\n Enter the value of n :");
scanf("%d", &n);
for (i=1;i<=n;i++)
printf("\n %d", i);
return 0;
}
```

```
#include <stdio.h>
int main()
{
for(;;)
printf(" C Programming");
return 0;
}
```