

IMPORTANCE OF LOOPS

Notice

Our Office will be closed on Sunday,
September 5.
Enjoy your holiday!



Program to send notice to 1000 employees.

```
int employeeCode = 1000;  
while(employeeCode != 0)  
{  
    Send the above notice  
    employeeCode --;  
}
```

WHILE LOOP

Syntax:

```
while(expression)
{
    Statement1;
    Statement2;
    .
    .
    .
}
```

Example:

```
int i = 3
while(i > 0 )
{
    print ( i );
    i--;
}
```

i

3

Console Window

WHILE LOOP

Syntax:

```
while(expression)
{
    Statement1;
    Statement2;
    .
    .
    .
}
```

Example:

```
int i = 3
while( FALSE )
{
    print ( 1 );
    i--;
}
```

Come out of
the loop!

i

0

Console Window

3 2 1

FOR LOOP

```
for(initialization; condition; increment/decrement)
{
    Statements;
}
```

```
int i;
i = 3;
while(i > 0)
{
    print (i);
    i--;
}
```

≡

```
for(i = 3; i > 0; i--)
{
    print (i);
}
```

DIFFERENCE BETWEEN WHILE LOOP AND DO-WHILE LOOP

While

```
int i = 0;
while(i > 0)
{
    printf("%d", i);
    i--;
}
```

Output: No Output

do-While

```
int i = 0;
do
{
    printf("%d", i);
    i--;
} while(i > 0);
```

Output: 0

WHEN SHOULD I PREFER DO-WHILE OVER WHILE?

Write a program which allows user to enter an integer until he/she enters a value zero.

```
int n;
printf("Enter an integer\n");
scanf("%d", &n)

while(n != 0)
{
    printf("Enter an integer\n");
    scanf("%d", &n);
}

printf("You are out of the
loop")
```

```
int n;
do
{
    printf("Enter an integer\n");
    scanf("%d", &n);
} while(n != 0)

printf("You are out of the
loop")
```


BREAK

Used to terminate from the loop

Example: WAP to allow user to enter integers until he/she enters a negative number or a zero.



Management x

Projects Symbols

Workspace

Start here x Untitled1.c x Untitled2.c x

```
1 //program to allow user to enter integers until he/she enters a negative number or zero
2
3 #include <stdio.h>
4
5 int main() {
6     int n;
7     printf("Enter a number\n");
8     scanf("%d", &n);
9
10    while(n != 0)
11    {
12        if(n < 0)
13            break;
14        printf("Enter a number\n");
15        scanf("%d", &n);
16    }
17 }
18
```

Logs & others

Cccc x Build log x Build messages x CppCheck x CppCheck messages x Cscope x Debugger x DoxyBlocks x

CONTINUE

Similar to break but instead of terminating from the loop, it forces to execute the next iteration of the loop.

Example: WAP to print all odd numbers from 1 to 20.



Management x

Projects Symbols

Workspace

Start here x Untitled1.c x Untitled2.c x

```
1 //program to print all odd numbers from 1 to 20
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int i, n=2;
8     for(i=1; i<=20; i++)
9     {
10         if(i == n)
11         {
12             n = n+2;
13             continue;
14         }
15         printf("%d ", i);
16     }
17 }
18
```