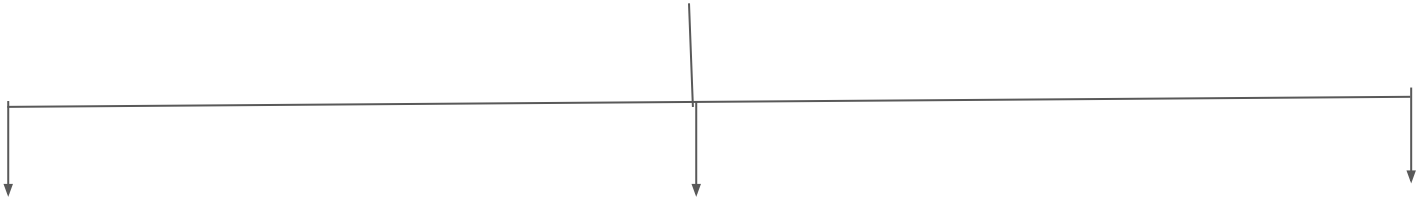
Control Statements

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering,Pune
www.harishgadade.com

Control Statements



Decision Making Statements
- If - Statements
- Switch-Case Statements
- Conditional Statements

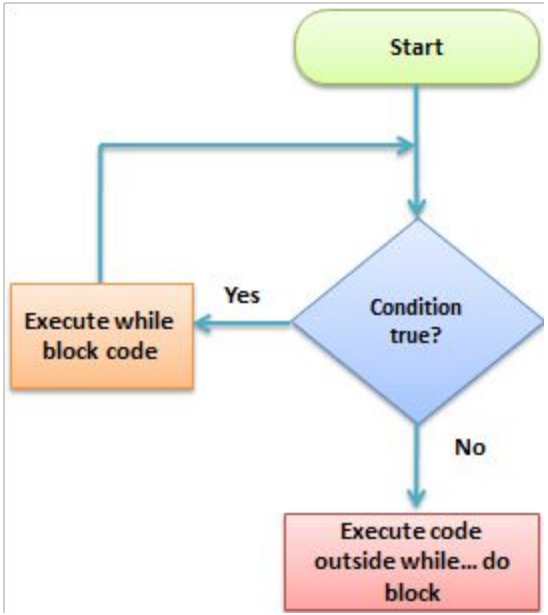
Looping / Iterative Statements
- While Loop
- Do-While Loop
- For Loop

Jumping Statements
- Break Statements
- Continue Statements
- Goto Statements

While Loop

- It is used to repeat a block of statements until condition becomes true
- **Syntax**

```
initialization;
while(condition)
{
    Block of Statements;
    Increment / decrement;
}
```



While Loop

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

Output: 1 2 3 4

Examples of Infinity While Loop

```
#include <stdio.h>
int main()
{
      int var=1;
      while (var <=2)
      {
        printf("%d ", var);
      }
}
```

```
#include <stdio.h>
int main()
{
      int var = 6;
      while (var >=5)
      {
         printf("%d", var);
         var++;
      }
   return 0;
}
```

```
#include <stdio.h>
int main()
{
     int var =5;
     while (var <=10)
     {
        printf("%d", var);
        var--;
     }
     return 0;
}
```

Examples of Infinity While Loop

```
#include <stdio.h>
int main()
{
      int var=1;
      while (var <=2)
      {
        printf("%d ", var);
      }
}
```

The program is an example of **infinite while loop**. Since the value of the variable **var** is same (there is no ++ or – operator used on this variable, inside the body of loop) the condition **var<=2** will be true forever and the loop would never terminate.

```
#include <stdio.h>
int main()
{
      int var = 6;
      while (var >=5)
      {
         printf("%d", var);
         var++;
      }
   return 0;
}
```

Infinite loop: var will always have value **>=5** so the loop would never end.

```
#include <stdio.h>
int main()
{
     int var =5;
     while (var <=10)
     {
        printf("%d", var);
        var--;
     }
     return 0;
}
```

Infinite loop: var value will keep decreasing because of **- -** operator, hence it will always be **<= 10.**

Do-While Loop

- A do...while loop in C is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop.
- **Syntax**

```
do
{
    Block of Statements;
    Increment / decrement;
}while(condition);
```

Do-While Loop

```
#include<stdio.h>
void main()
{
    int i=11;
    do
    {
        printf("%d\n",i);
        i++;
    }while(i<10);
    printf("Hii\n");
}
```

For Loop

- A for loop is a more efficient loop structure in 'C' programming.
- **Syntax**

```
for (initial value; condition; incrementation or decrementation )
{
  Block of statements;
}
```

For Loop

```
for (initial value; condition; incrementation or decrementation )
{
  Block of statements;
}
```

Forms of For Loop

```
int i;
for(i=0;i<n;i++)
{
    printf("%d",i);
}
```

```
int i=0;
for(;i<n;i++)
{
    printf("%d",i);
}
```

```
int i;
for(i=0;i<n;)
{
    printf("%d",i);
    i++;
}
```

```
int i=0;
for(;i<n;)
{
    printf("%d",i);
    i++;
}
```

```
int i,n;
for(i=0;i<n;i++);
```

```
int i;
for(i=10;i>0;i--)
{
    printf("%d",i);
}
```