Programming Constructs

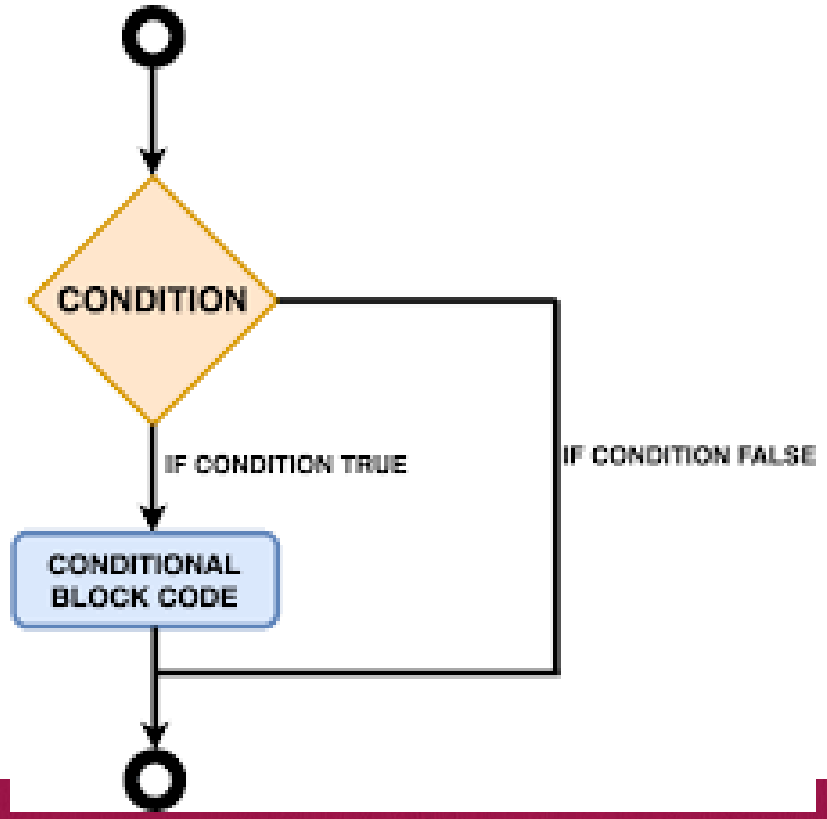
Dr. Rekha P
Amrita Center for Wireless Networks & Applications
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
rekhap@am.amrita.edu

If Statement

- Used to execute a piece of code based on a condition

```
If(boolean-expression){
        Statements
}
```

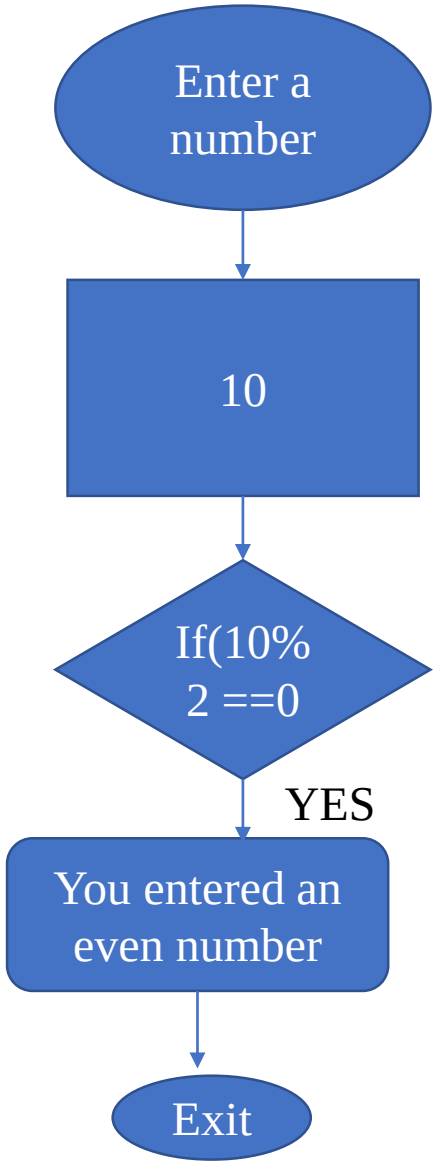
The statements are executed if the boolean expression evaluates true



If Statement

- Check the given number is Even

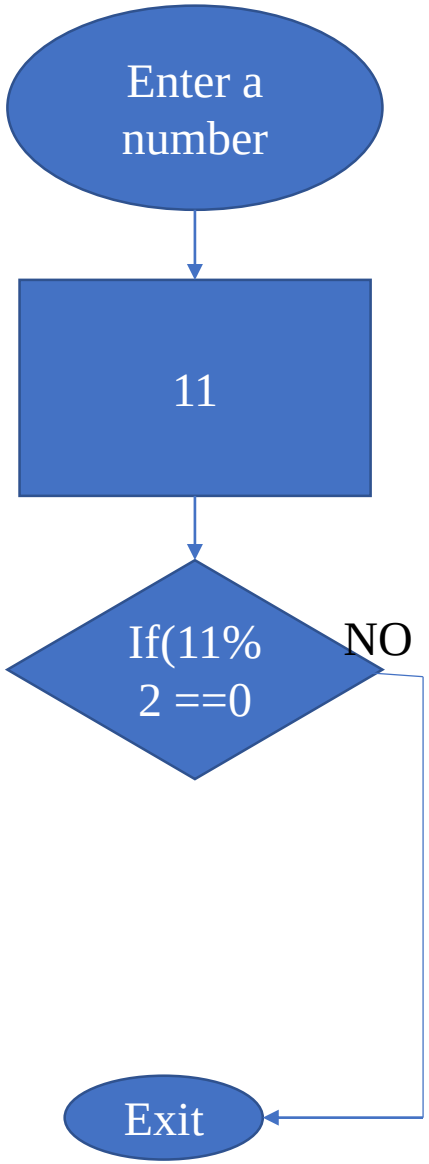
```
import java.util.Scanner;
public class Even {
    public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scan.nextInt();
      scan.close();
        //checking even
      if(num%2 == 0)
            System.out.println("You  entered  an
Even number ") ;
    }
}
```



If Statement

- Check the given number is Even

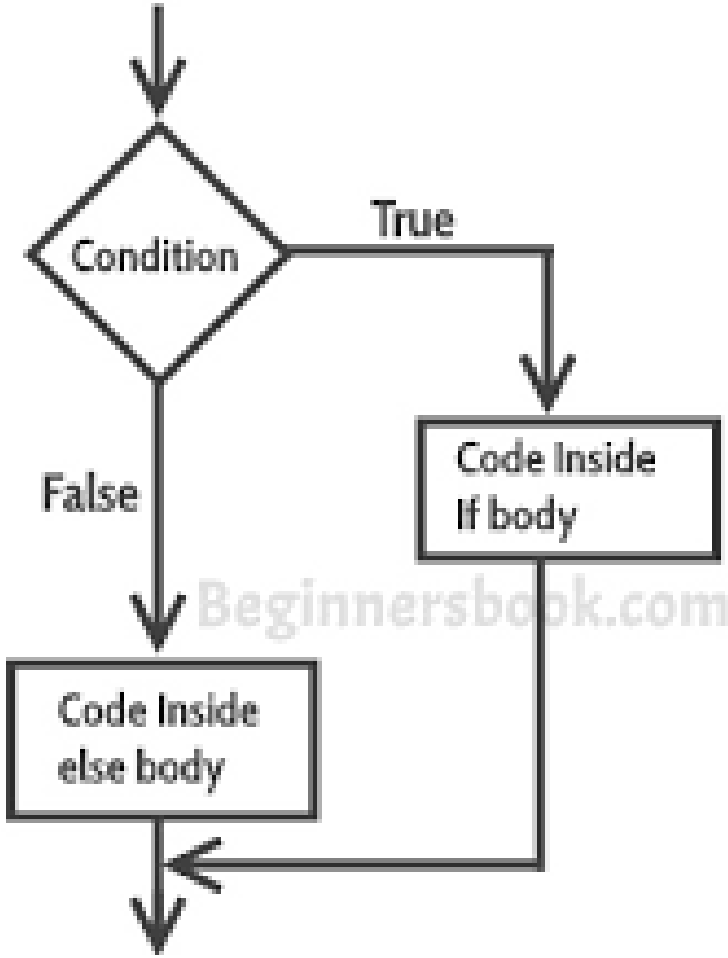
```
import java.util.Scanner;
public class Even {
    public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scan.nextInt();
      scan.close();
        //checking even
      if(num%2 == 0)
            System.out.println("You  entered  an
Even number ") ;
    }
}
```



If –Else Statement

- Used to execute a piece of code based on a condition

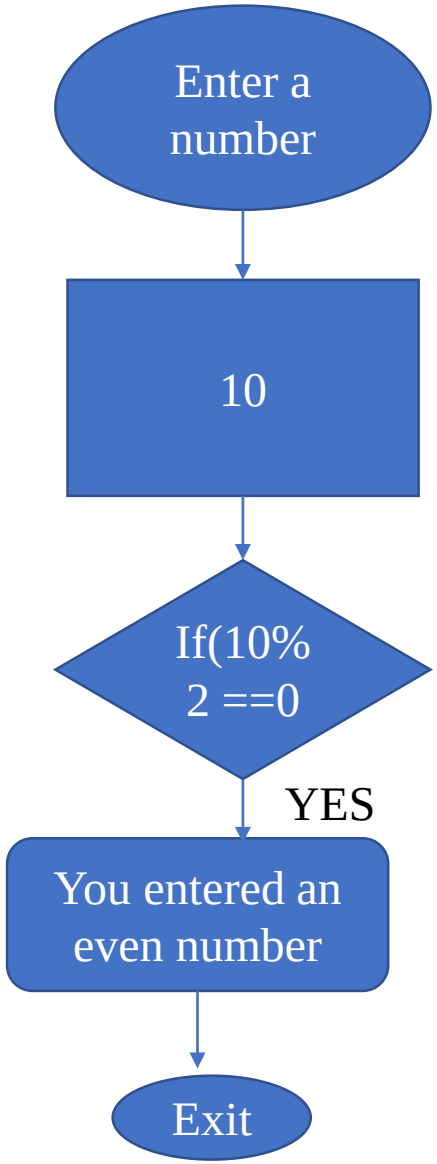
```
If(boolean-expression){
        Statements-1
}
else{
Statements-2
}
```



If-Else Statement Example

- Check the given number is Even/Odd

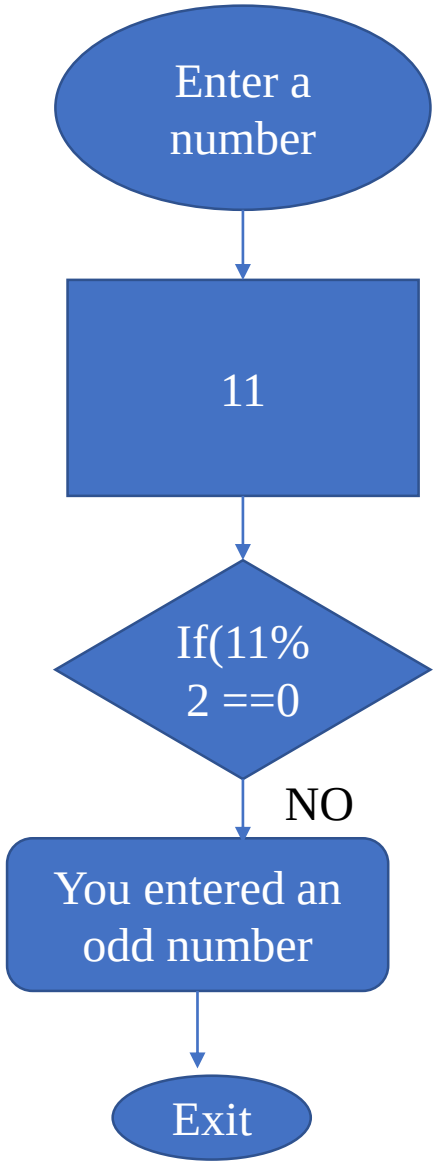
```
…{
Scanner scan = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scan.nextInt();
      scan.close();
      if(num%2 == 0)
            System.out.println("You   entered   an
Even number ") ;
      else
            System.out.println("You   entered   an
Odd number ") ;
    }
}
```



If-Else Statement Example…

- Check the given number is Even/Odd

```
…{
Scanner scan = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scan.nextInt();
      scan.close();
      if(num%2 == 0)
            System.out.println("You   entered   an
Even number ") ;
      else
            System.out.println("You   entered   an
Odd number ") ;
    }
}
```



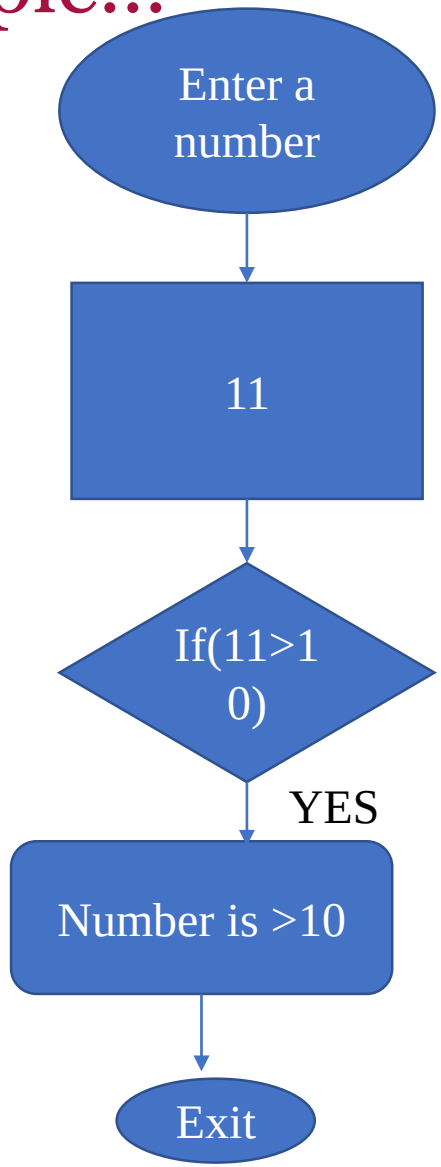
Nested If –Else Statement

- Using an if-else statement inside another if-else statement

```
if(boolean-expression1)
{
      Statements-1
}
else
      {
            if(boolean-expression2)
            {
                  Statements-2
            }
            else
            {
                  Statements-3
            }
      }
```

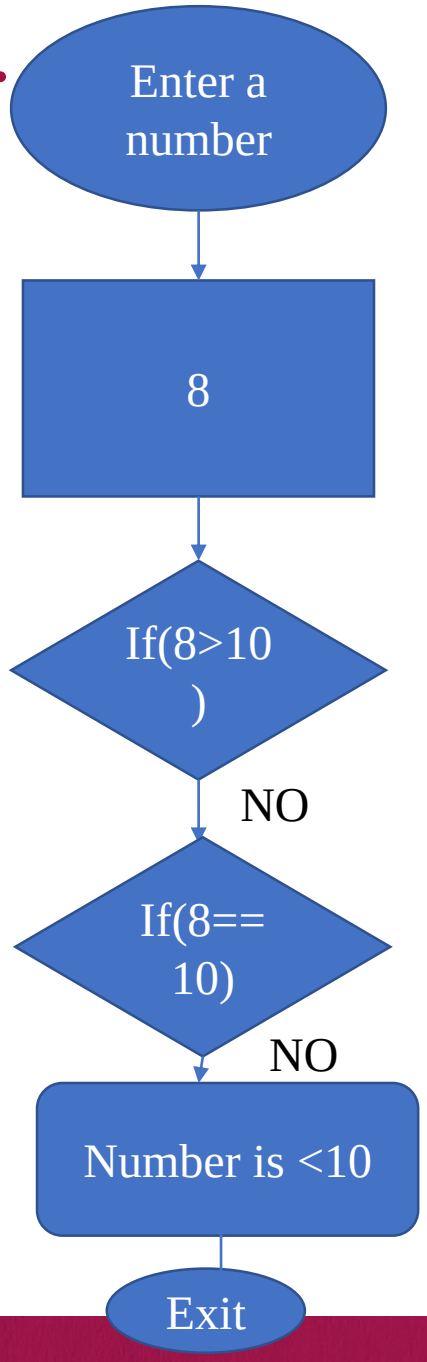
Nested If-Else Statement Example...

```
…{
    if(num>10)
        System.out.println(“Number      is
>10 ") ;
    else
        if(num ==10)
            System.out.println(“Number
is =10 ") ;
   else
System.out.println(“Number is <10 ") ;
}
```



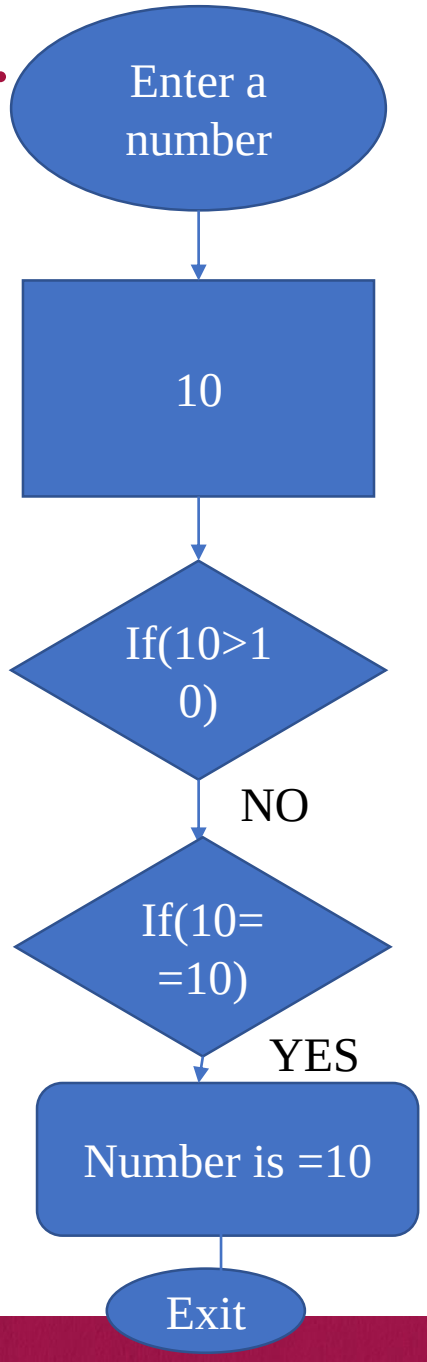
Nested If-Else Statement Example...

```
…{
    if(num>10)
        System.out.println(“Number      is
>10 ") ;
    else
        if(num ==10)
            System.out.println(“Number
is =10 ") ;
   else
System.out.println(“Number is <10 ") ;
}
```



Nested If-Else Statement Example...

```
…{
    if(num>10)
        System.out.println(“Number      is
>10 ") ;
    else
        if(num ==10)
            System.out.println(“Number
is =10 ") ;
   else
System.out.println(“Number is <10 ") ;
}
```



Switch Statement

- Used to execute different statements based on equality

```
switch(expression){
case expression_1:
    Statements;
    break ;
case expression_2:
    Statements;
    break ;
…
default:
    Statements;
    break ;
}
```

Switch Statement Example

```
int n = scan.nextInt() ;
switch(n){
case 10:
      System.out.Println("You entered 10") ;
      break ;
case 50:
      System.out.Println("You entered 50") ;
      break ;
      case 30:
      System.out.Println("You entered 30") ;
      break ;
default:
      System.out.Println("You entered something else") ;
      break ;
}
```

Switch Statement Example...

```
int n = scan.nextInt() ;
switch(n%2){
case 0:
    System.out.Println("You entered an Even number") ;
    break ;
default:
    System.out.Println("You entered an Odd number") ;
    break ;
}
```

Loops

- Used to execute a block of code more than once

```
System.out.Println("Welcome!!!") ;
System.out.Println("Welcome!!!") ;
System.out.Println("Welcome!!!") ;
System.out.Println("Welcome!!!") ;
System.out.Println("Welcome!!!") ;
int i = 1;
while(i <= 5){
        System.out.Println("Welcome!!!") ;
        i++
}
```

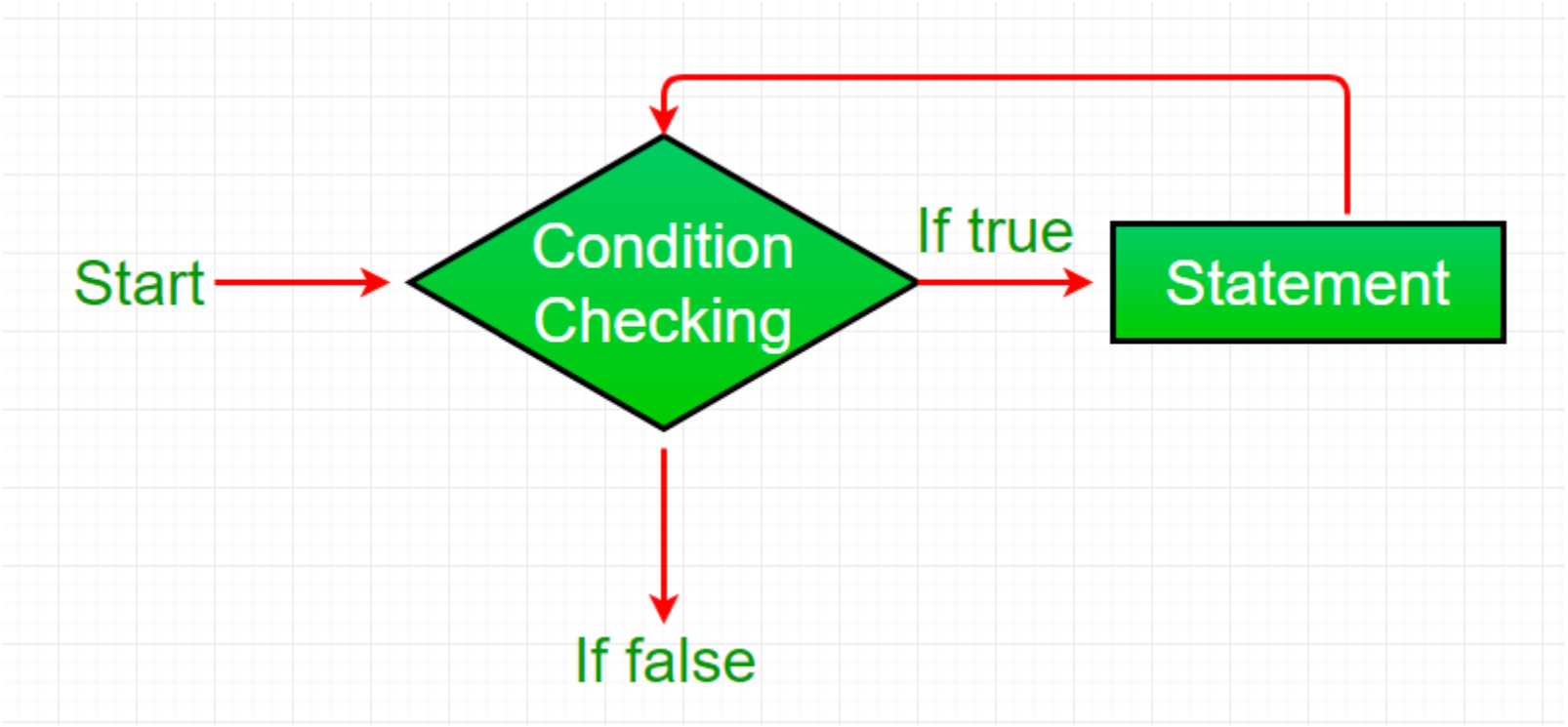
Loops

- Used to execute a block of code more than once

```
System.out.Println("1. Welcome!!!") ;
System.out.Println("2. Welcome!!!") ;
System.out.Println("3. Welcome!!!") ;
System.out.Println("4. Welcome!!!") ;
System.out.Println("5. Welcome!!!") ;
int i = 1;
while(i <= 5){
        System.out.Println(i+". Welcome!!!") ;
        i++
}
```

Types of Loops

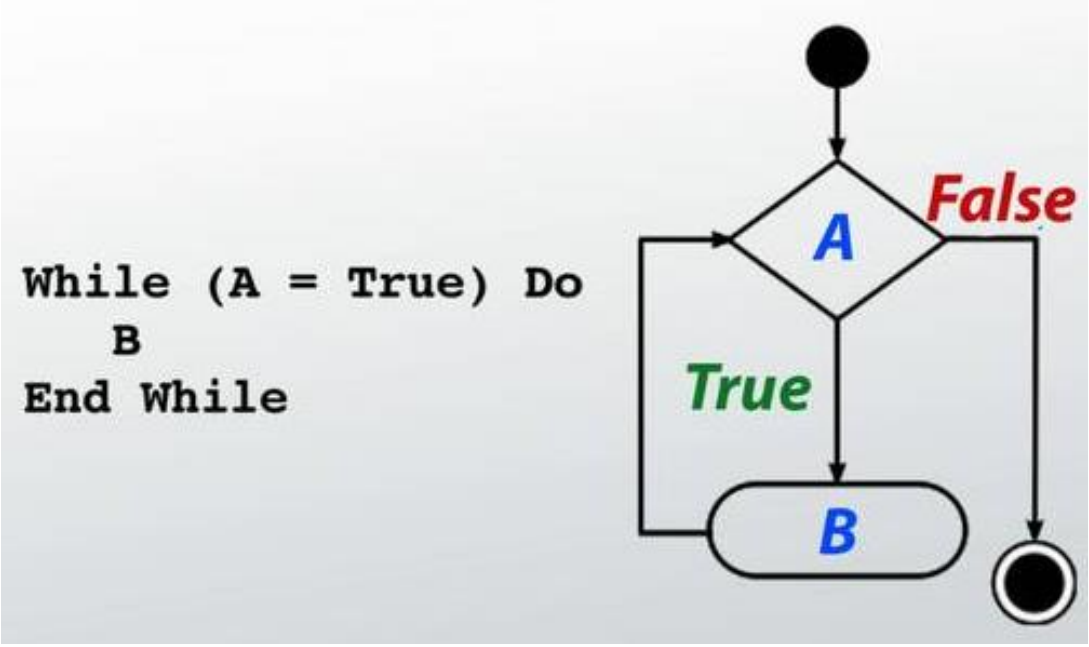
- While
- Do While
- For



While Loop

```
While(boolean expression){
Statements;
}
```

- Statements keep executing as long as the boolean expression evaluates true
- Each execution is called an iteration



While Loop Example

```
int i = 1; // used in the condition
while(i <= 5){
        System.out.Println(i+". Welcome!!!") ;
        i++ ; //Affects the condition
}
```

i=1	1. Welcome!!!	Iteration 1
i=2	2. Welcome!!!	Iteration 2
i=3	3. Welcome!!!	Iteration 3
i=4	4. Welcome!!!	Iteration 4
i=5	5. Welcome!!!	Iteration 5

Infinite Loops

- Loops that never ends

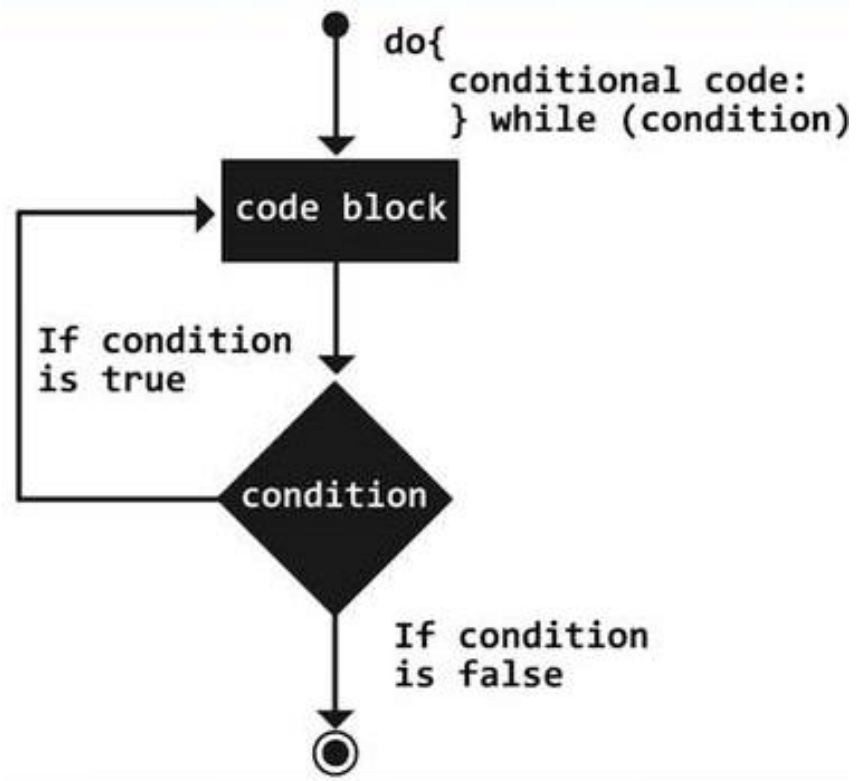
```
int i = 1; // used in the condition
while(i <= 5){
        System.out.Println(i+“. Welcome!!!”) ;
}
```

```
while(true){
        System.out.Println(Welcome!!!”) ;
}
```

Do While Loop

```
do{
Statements;
} While(boolean expression);
```

- Statements keep executing as long as the boolean expression evaluates true



Do While Loop Example

```
int i = 1; // used in the condition
do{
        System.out.Println(i+". Welcome!!!") ;
        i++ ; //Affects the condition
} while(i <= 5);
```

		Iteration 1
i=2	2. Welcome!!!	Iteration 2
i=3	3. Welcome!!!	Iteration 3
i=4	4. Welcome!!!	Iteration 4
i=5	5. Welcome!!!	Iteration 5

Difference between While & Do While

- While – Check condition and then execute
- Do While – Execute and then check the condition

```
int i = 100;
while(i <= 5){
        System.out.Println(i+“. Welcome!!!”) ;
}
```

0 Iteration

```
do{
        System.out.Println(i+“ Welcome!!!”) ;
} while(i <= 5);
```

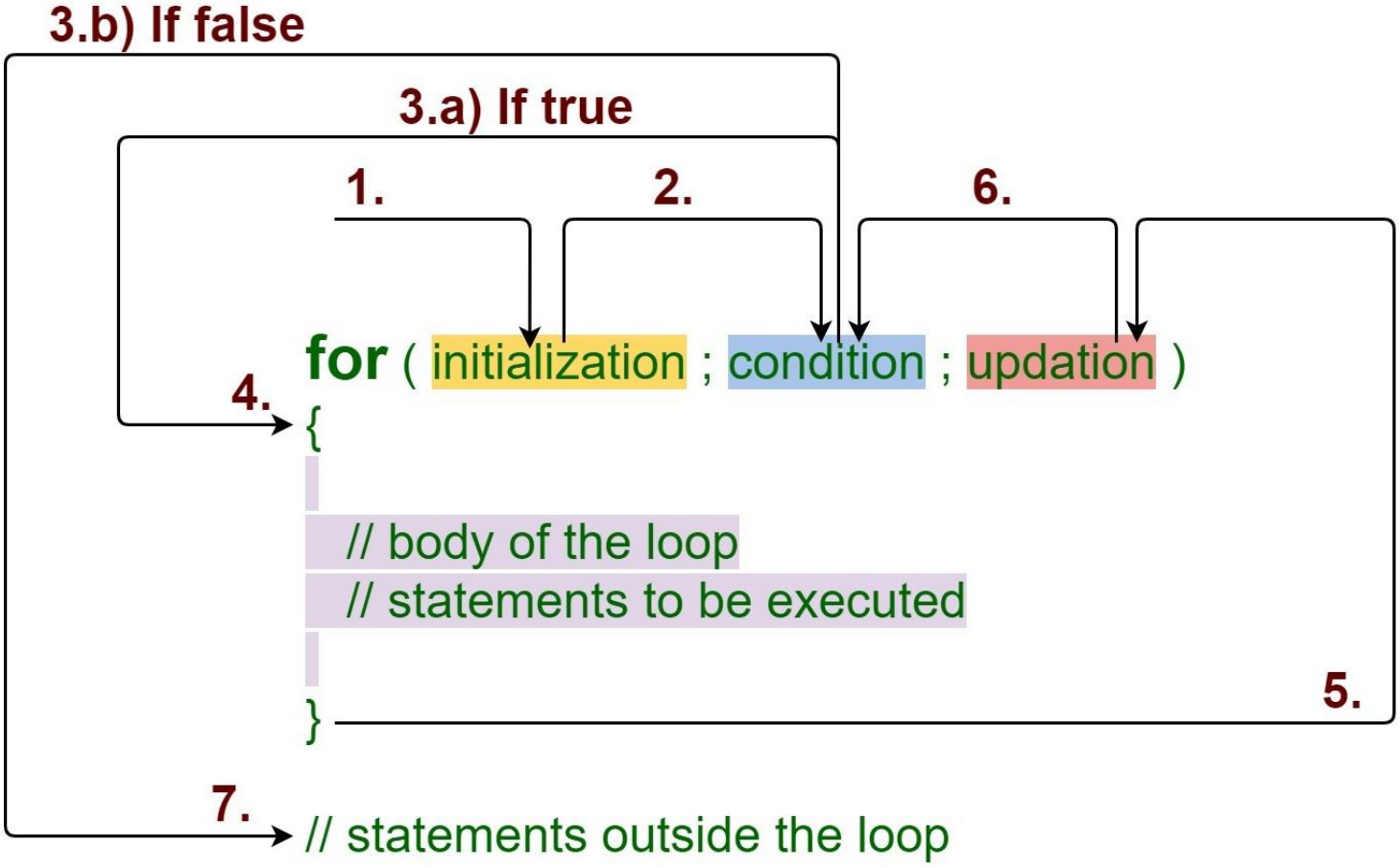
1 Iteration

100. Welcome!!!

For Loop

```
for(initialisation;condition;updation){
Statements;
}
```

- Statements keep executing as long as the boolean expression evaluates true



Comparison between While & For

```
int i = 1 ;
while(i <= 5){
        System.out.Println(i+“. Welcome!!!”) ;
        i++ ;
}
```

```
for(int i = 1; i<=5; i++{
        System.out.Println(i+“ Welcome!!!”) ;
}
```

Infinite For loops

```
for(; ;)
        System.out.Println(“ Welcome!!!”) ;
}
```

```
for(;true;)
        System.out.Println(“ Welcome!!!”) ;
}
```

Nested Loops

- Using one loop inside another loop

```
for(i=1 ; i < 4 ; i++){
    for(j=1 ; j < 2 ; j++){
        System.out.Println(i + " "+ j) ;
    }
    System.out.println();
}
```

i/j		
i=1	J=1	J=2
i=2	J=1	J=2
i=3	J=1	J=2

```
1 1
1 2

2 1
2 2

3 1
3 2
```

Concluding Thought...

"Whatever you do and wherever you are, relax and you will see how much power you will gain. "— **Amma**