Repetition Statements

Introduction

This unit covers how to write code segments that repeat.

Repetition Statements

Repetition statements are statements that repeat a group of code while its condition is true. Repetition statements are also known as loops.

While Statements

A while statement runs its code when the condition is true and skips the code when the condition is false. After running its code a while loop will recheck the condition and run the code again if the condition is still true.

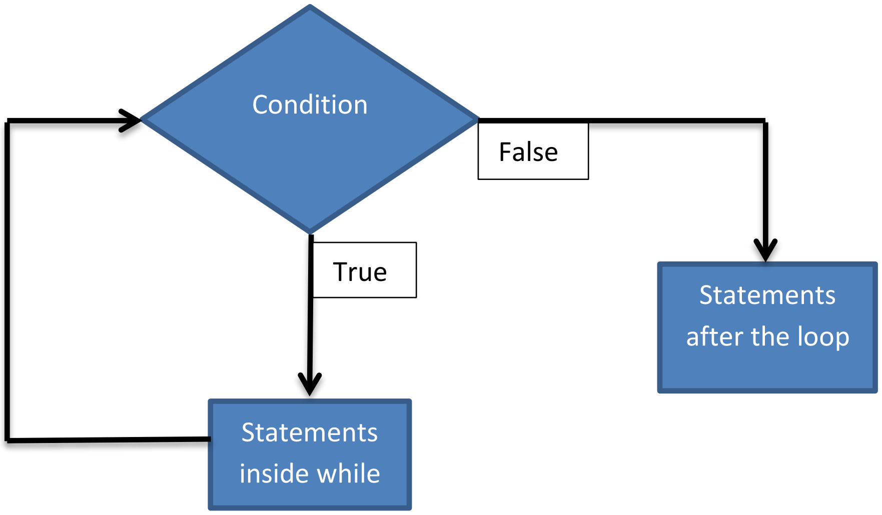
Format:

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

Times the code will run:

Minimum: 0
Maximum: infinity

Flowchart for a while:



Example:

```
int x = 0;
while(x <= 10)
{
    System.out.println(x);
    x+=2;
}
```

Output:

```
0
2
4
6
8
10
```

Do Statements

A Do statement runs the code inside it once and then it will continue to run the code over and over until the condition becomes false.

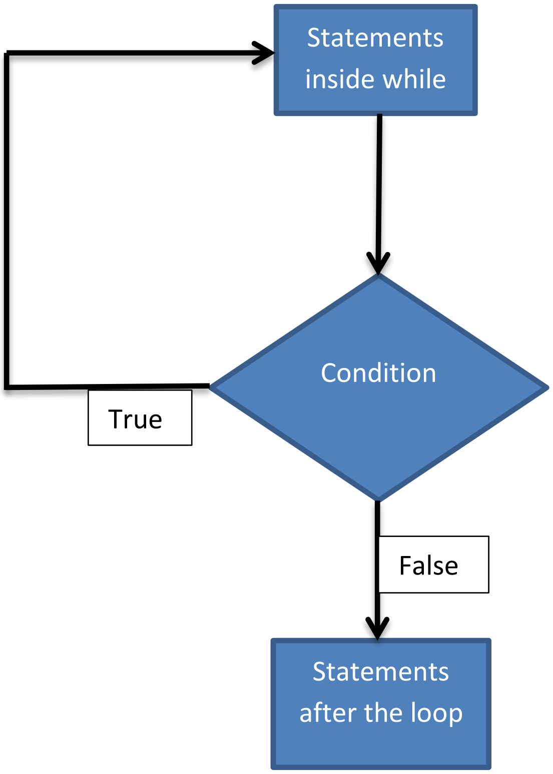
Format:

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

Times the code will run:

Minimum: 1
Maximum: infinity

Flowchart for a do:



Example:

```
Scanner keyboard = new Scanner(System.in);
int x = 0;

do
{
        System.out.print(“Enter a positive number: ”);
        x=keyboard.nextInt();
        if(x <=0)
                System.out.println(“Try again.”)
} while(x <= 0);
```

Result of code:

This code will keep asking the user for a positive integer until a number greater than 0 is entered.

For Statements

The for statement is designed to run its code a set number of times.

Parts of a for:

- Initialization – Sets the value for a counter.
  - A variable can be created here
  - This part can be left blank
- Condition – Boolean expression to determine if the code in the for will be run.
  - This part cannot be left blank
- Incrementing – Changes the value of the counter
  - Any variable change is valid here
  - This part can be left blank

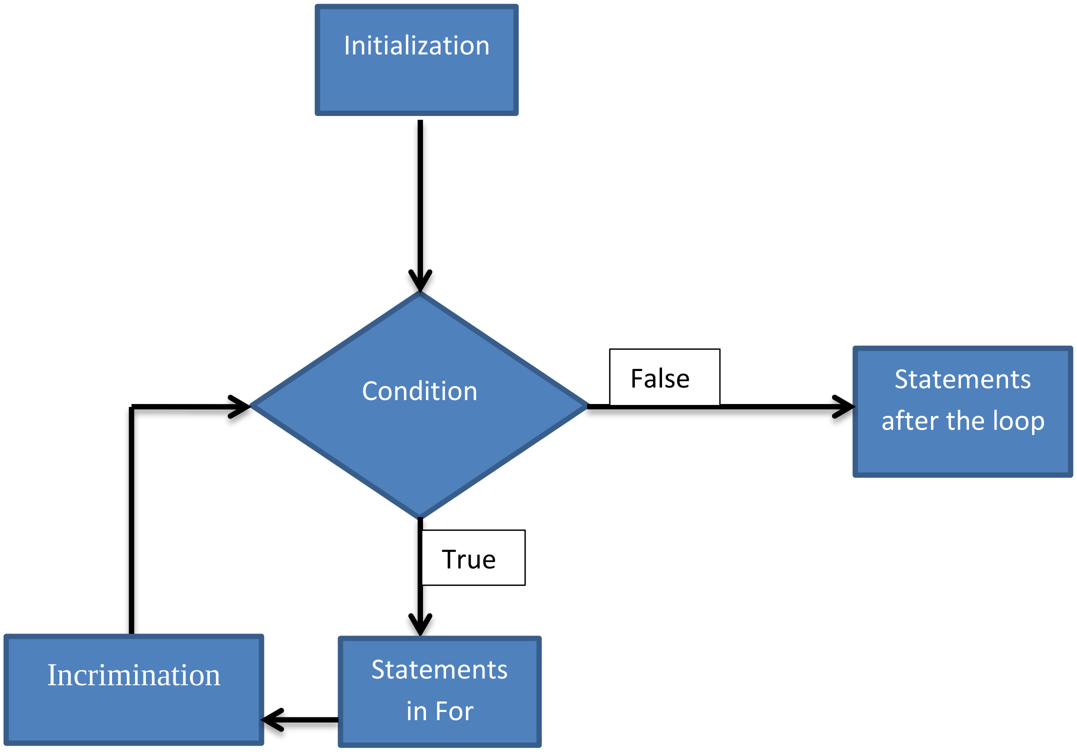
Format:

```
for(initialization; condition; incrimination)
{
        //code
}
```

Times the code will run:

Minimum: 0
Maximum: infinity

Flowchart for a for:



Example:

```
int answer = 1;
int base=2;
int power = 3;

for(int a =0; a<power; a++)
        answer  = answer*base;

System.out.println(base + “^” + power + “ = “ + answer);
```

Output:

```
2^3 = 8
```

Break & Continue

Break

The break statement can be used in a repetition statement to stop the current iteration and exit the loop. An **iteration** is execution of the code in a repetition statement. When a break statement is reached the code remaining in the current iteration is skipped and the loop does not again.

Format:

```
break;
```

Continue

The continue statement can be used in a repetition statement to stop the current iteration and then check to see if the loop should be run again. When a continue statement is reached the code remaining in the current iteration is skipped and the loop is checked to see if it should run again.

Format:

```
continue;
```

Blue Pelican Sections

Lesson 11
Lesson 12

Terms

Iteration	A single run of the code in a repetition statement.
Repetition Statements	Statements that repeat a group of code while its condition is true.