

## < Loops

- repeat certain actions as long as a certain condition exists when tested
- Should have the following components:
  - Initialize of the tested variable
  - Boolean\_expression with tested
  - variable
  - Code\_block to be executed
  - Update of the test variable when needed

## < Loop types

- Event-controlled loop
- Event initialized
- Test event
- Execute body of loop
- Update event if needed
- Go back to test
- Count-controlled loop
- Initialize count
- Test count
- Execute body of loop
- Update count if needed
- Go back to test

## < Loops

- for
- while
- do ... while
- Getting out

## < while Statements

- Repeats indefinitely through statements and performs the statements zero or more times while the Boolean expression is true

The while loop:

```
while (boolean)
{
Statement or block;
}
```

## < while example

```
int i = 0;
while ( i < 10)
{
System.out.println("Are you finished yet?");
i ++;
}
System.out.println("Done");
```

## < do ... while

- do – Iterates indefinitely through statements and performs the statements *one* or more times compared to while loop and for loop are zero/many iterative loops

- The do loop is:

```
do
{
code of statements to be executed
} while (boolean);
```

## < do ... while

```
int i = 0;
do
{
System.out.println("Are we there yet?");
i++;
} while (i<10);
System.out.println("Done");
```

## < The for Loop

- For count-controlled loops only
- Loops through a sequence of statements a predetermined number of times
- Syntax:

```
for (init_expr; boolean test; update_expr)
{
Statement or block;
}
```

## < for example

```
for (int i = 0; i<10; i++)
{
System.out.println ("Are we there yet?");
}
System.out.println ("Finally!");
```

## < Special Loop Flow Control

- `break [label];`
- `continue [label];`
- `Label: statement; //` where statement should be a loop

## < The `continue` Keyword

- Permits you to end a loop iteration but the continue the execution of the loop
- Used inside `while`, `for`, and `do` loops only
- Should be used when the alternative code is much more complex

## < Loops `break`

- Permits you to end the loop iteration and you exit immediately

## < Nested Loops

- Required to print this:  
    &&&&&  
    &&&&&  
    &&&&&
- Inner loop prints rows, outer loop prints columns
- “Rows go slow, columns go fast”