



Lets Code!

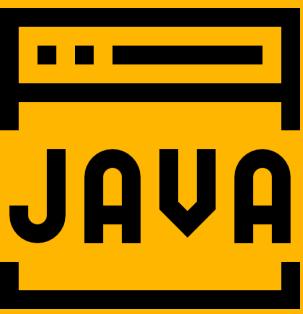
Fundamentals: Control Flow

Instructor: Tariq Hook

You can find me on github @code-rhino

Key Terms

- Block Scope
 - Conditional Statements
 - Loops
 - Switch Statements
- Determinate Loops
 - Nested Loops



Block Scope

Block Scope

Java, like any programming language, supports both conditional statements and loops to determine control flow.

Before learning about control structures, you need to know more about blocks.

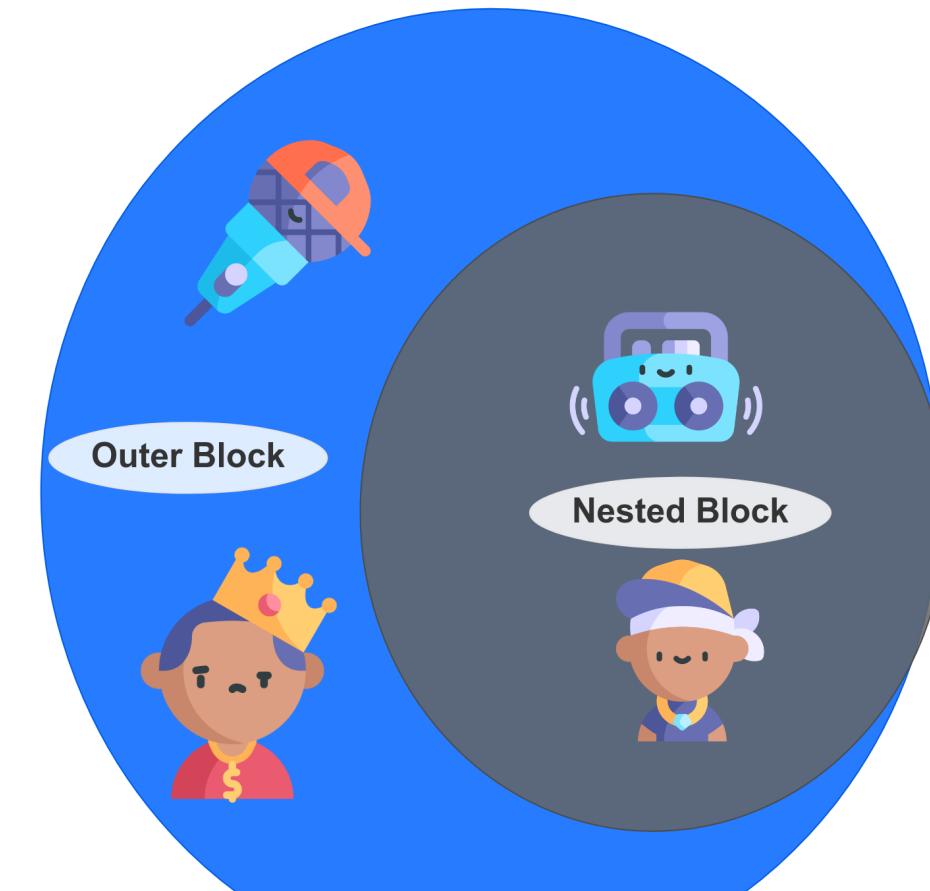
A block or compound statement consists of a number of Java statements, surrounded by a pair

```
public static void main(String[ ] args)
{
    int n;
    ...
    {
        int k;
        ...
    } // k is only defined up to here
}
```

Blocks define the scope of your variables.

Blocks can be nested.

Block Scope

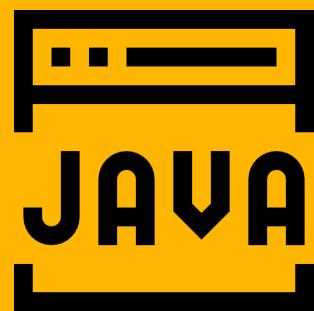


Block Scope

You may not declare identically named variables in two nested blocks

```
public static void main(String[ ] args)
{
    int n;
    ...
    {
        int k;
        int n;
    }
}
```

Error can't redefine n in inner block



Conditional Statements

Conditional Statements

The conditional statement in Java has the form.

```
if(condition) statement  
{  
    statement1  
    statement2  
    ...  
}
```

If/Else Statement

The conditional statement in Java has the form.

```
if(condition) statement
{
    statement1
    statement2
    ...
}

if (currentSensor.sensesMovement() == true) {
    currentSensor.reset();
}
```

If/Else Statement

The conditional statement in Java has the form.

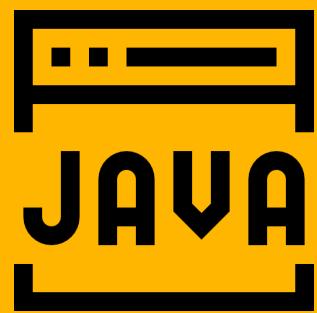
```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is false and  
    // condition2 is true  
} else {  
    // block of code to be executed if the condition1 is false and  
    // condition2 is false  
}
```

Control Flow

Control structure code creates branches, or different paths that the code can take.

One code path when the condition succeeds and one for when it fails.

```
if(condition) {  
    // do something  
} else {  
    // do something else  
}
```



Loops

Loops

The while loop executes a statement (which may be a block statement) while a condition is true

```
while(condition) statement  
  
while (player1.lives()) {  
    player1.calculateCurrentLevels();  
}
```

The while loop will never execute if the condition is false at the outset

If you want to make sure a block is executed at least once, you need to move the test to the bottom, using the do/while loop.

```
do statement while (condition);
```

```
do sensors1.processNextSensorEvent() while (sensor1.hasEvents());
```

Determinate Loops

The for loop is a general construct to support iteration controlled by a counter or similar variable that is updated after every iteration.

```
for (int i = 1; i <= 10; i++)
    System.out.println(i);

for (int i = 23; i != 0 10; i--)
    System.out.println(i);
```

Multiple Selections—The switch Statement

```
switch (choice)
{
    case 1:
        ...
        break;
    case 2:
        ...
        break;
    case 3:
        ...
        break;
    default:
        // bad input ...
        break;
}
```

A case label can be

- A constant expression of type char, byte, short, or int
- An enumerated constant
- Starting with Java SE 7, a string literal

Statements That Break Control Flow

The same break statement that you use to exit a switch can also be used to break out of a loop

```
while (years <= 100) {  
    balance += payment;  
    double interest = balance * interestRate / 100;  
    balance += interest;  
    if (balance >= goal) break;  
    years++;  
}
```

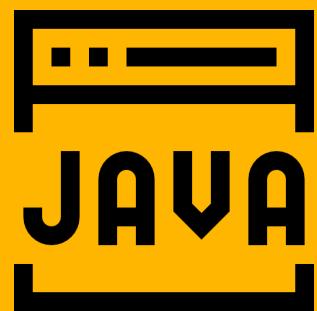
The continue statement transfers control to the header of the innermost enclosing loop

```
Scanner in = new Scanner(System.in);
while (sum < goal)
{
    System.out.print("Enter a number: ");
    n = in.nextInt();
    if (n < 0) continue;
    sum += n; // not executed if n < 0
}
```

Wrap Up

- Block Scope
- Conditional Statements
- Loops
- Switch Statements

- Determinate Loops
- Nested Loops



Keep Coding !!!

Clean Code is Happy Code