

Programming in the Small II: Control



188230 Advanced Computer Programming

Asst. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)
Department of Computer Engineering
Khon Kaen University

Agenda



Selection Statements

Using if and if...else

Nested if Statements

Using switch Statements

Repetition Statements

Looping: while, do, and for

Nested loops

Using break and continue

Selection Statements



if Statements

switch Statements

if Statements



Syntax

```
if (booleanExperssion) {  
    statement(s);  
}
```

Example:

```
public static void main(String[] args) {  
    if (args.length < 0) {  
        System.err.println("Usage:Hello <your name>");  
    }  
}
```

if...else Statements



Syntax

```
if (booleanExpression) {  
    statements for the true case;  
} else {  
    statements for the false case;  
}
```

if...else Example



```
if (radius >= 0) {  
    area = radius*radius*Math.PI;  
} else {  
    System.out.println("A circle radius cannot be  
        negative");  
}
```

Nested if Statements



Syntax

```
if (booleanExp 1) {  
    statements for the case booleanExp1 is true  
} else if (booleanExp2){  
    statements for the case booleanExp2 is true  
} else {  
    statements for the case that booleanExp1 is false and  
    booleanExp2 is false  
}
```

Note that there can be any number of boolean expressions, not just two.

Nested if Example



```
if (activity == playGame) {  
    System.out.println("Don't have time to read a book");  
} else if (activity == watchVideo) {  
    System.out.println("Which video are you watching");  
} else {  
    System.out.println("Hopefully, you read a book or a  
        slide before you come to class");  
}
```


switch Statements



Syntax

```
switch (variable) {  
    constant1:  
        statements when variable = constant1;  
        break;  
    constant2:  
        statements when variable = constant2:  
        break;  
    default:  
        statements when variable != constant1 and variable !=  
            constant2;  
}
```

switch Statements Example



```
switch (course) {  
    case 188200:  
        System.out.println("Discrete Maths");  
        break;  
    case 188230:  
        System.out.println("Advanced Compro");  
        break;  
    default:  
        System.out.println("Yeah :) I don't have to listen the course  
        material in English");  
}
```

Agenda



Selection Statements

- Using if and if...else

- Nested if Statements

- Using switch Statements

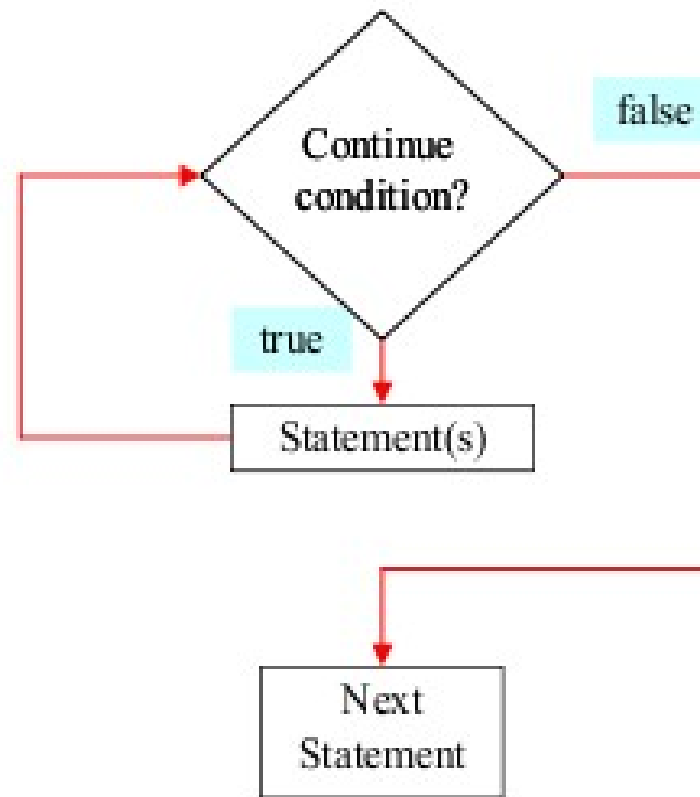
Repetition Statements

- Looping: while, do, and for

- Nested loops

- Using break and continue

while Loop Flow Chart



While Syntax and Example



Syntax

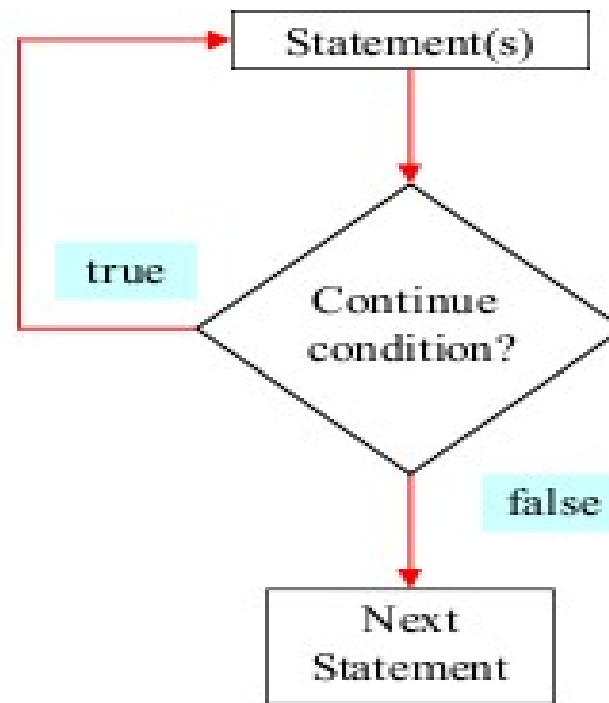
```
while (booleanExpression) {  
    statements;  
}
```

Sample

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

What is the output?

do...while Loop Flow Chart



do...while Example



Syntax

```
do {  
    statements;  
} while (booleanExpression);
```

Sample

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

What is the output?

while... and do...while



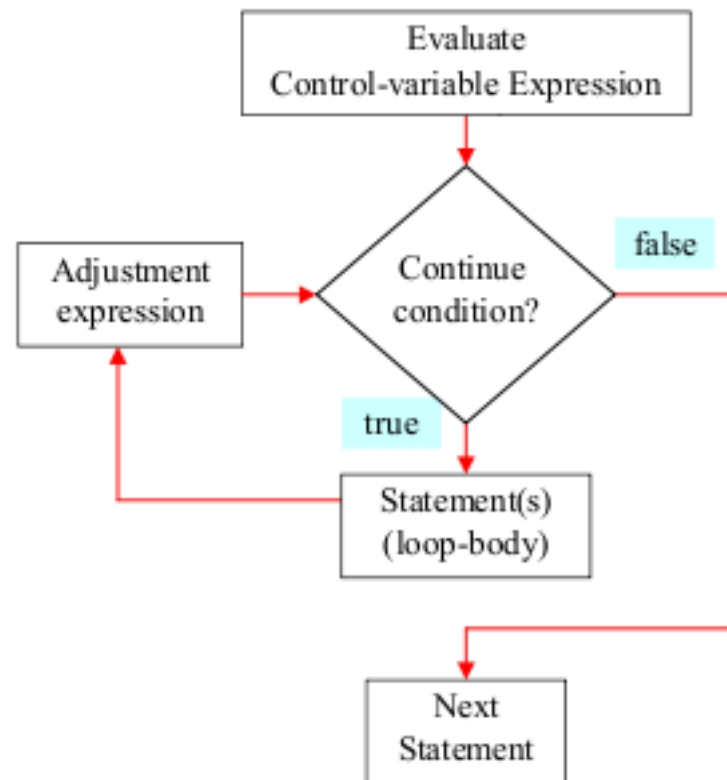
```
int i = 0;
while (i != 0) {
    System.out.println(i++);
}
```

What is the output?

```
int i = 0;
do {
    System.out.println(i++);
} while (i != 0);
```

What is the output?

for Loop Flow Chart



for syntax



```
for ((initialization); (continuation-condition);  
    (update)) {  
    statements;  
}
```

The initialization part is executed once before the loop begins

The continuation condition is execution is executed before each execution of the loop

The loop ends when this condition is false

The update can be any expression

The forever loop



```
for (;;) {
```

```
    ....
```

```
}
```

```
while (true) {
```

```
    ....
```

```
}
```

```
do {
```

```
    ...
```

```
} while (true);
```

for Example



```
// Initialize sum
float sum = 0;
// Keep adding 0.1 to sum
for (float i = 0.1f; i <= 1.0f; i = i+0.1f) {
    System.out.println("The current sum is " + sum);
    sum += i;
}
```

What is the output?

Nested for loop



If we want to get the output

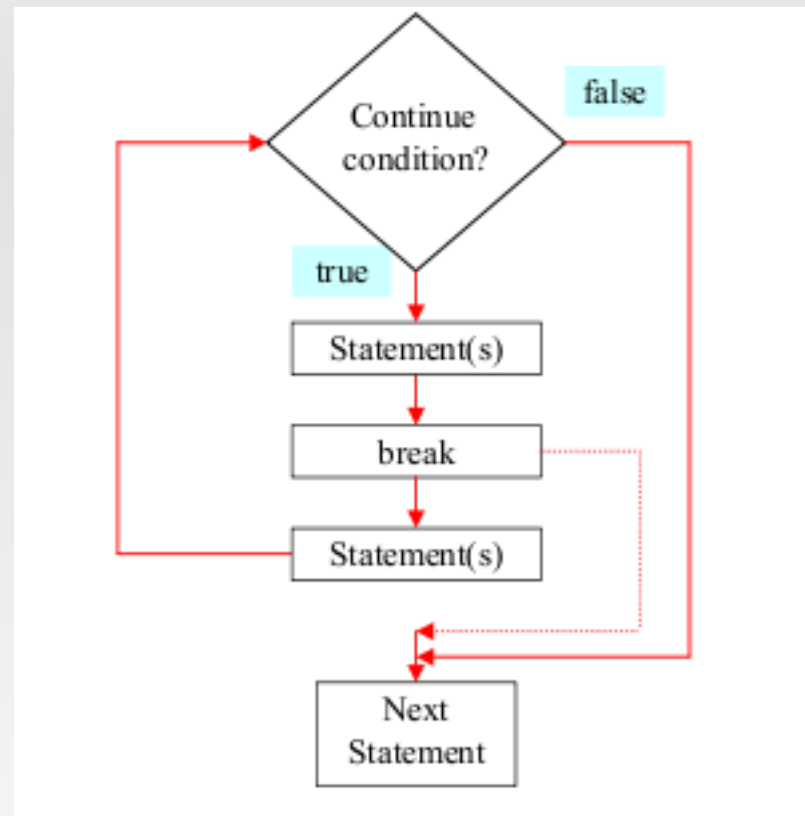
1 2 3 4

2 4 6 8

What will the code be?

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

break Flow Chart



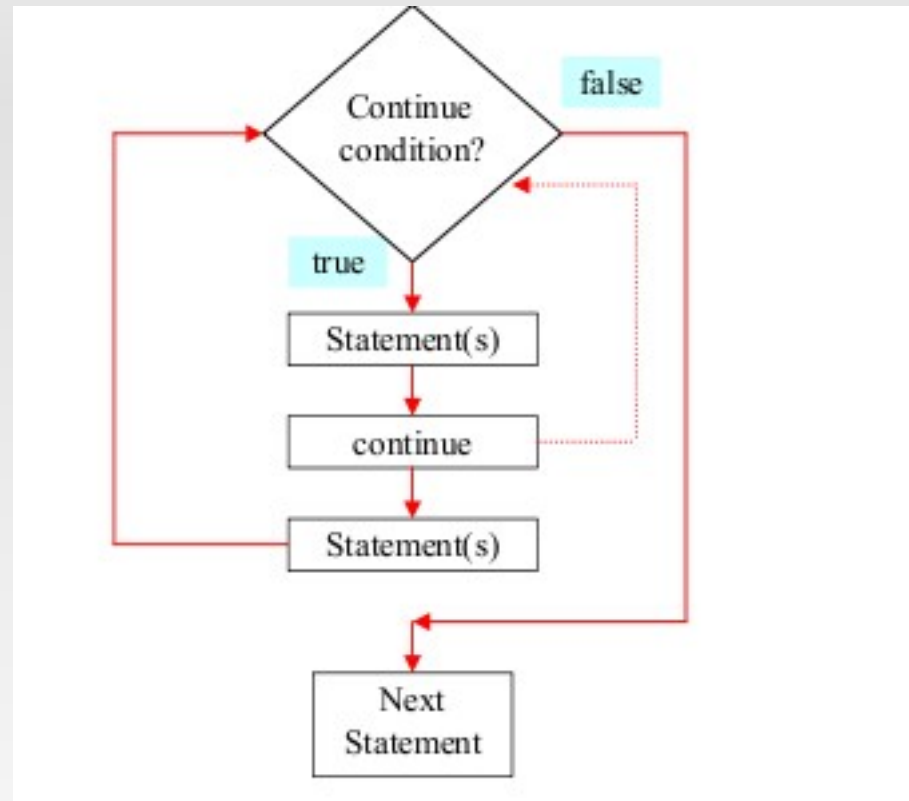
break Example



```
int sum = 0;
int item = 0;
while (item < 5) {
    System.out.println("Current sum is " + sum);
    item++;
    sum += item;
    if (sum >= 6)
        break;
}
```

What is the output?

continue Flow Chart



continue Example



```
int sum = 0;
int item = 0;
while (item < 5) {
    item++;
    if (item == 2)
        continue;
    sum += item;
}
```

```
System.out.println("The sum is " + sum);
```

What is the output?

The else Dangling Problem



What would be the output of this code

The computer follows attaches the else to “if ..”
that is closer

```
int x = -1, y = -1;  
if (x > 0)  
    if (y > 0)  
        System.out.println("First");  
else  
    System.out.println("Second");  
    System.out.println("Third");
```

References



- David J. Eck, "Introduction to Programming Using Java", Version 5.0, December 2006
<http://math.hws.edu/javanotes/>
- eclipse-tutorial:Developing open source Java applications with java.net and Eclipse, Available at
<https://eclipse-tutorial.dev.java.net/eclipse-tutorial/pa>
- Patrick Bouklee, "Java Package Tutorial", Available at
http://www.jarticles.com/package/package_eng.html