Control Flows (2)
Lecture 05

Min Zhang

zhangmin@sei.ecnu.edu.cn

2020.10.26

Software Engineering Institute

10:00-11:40, Monday, Room 319
Software Engineering Institute, East China Normal University

Review

Last lecture:

1. Expression, statement, block
2. `if, if-else, if-else-if-else...`
3. `switch-case`
4. `while`

control flow: `for`

for-statement

Syntax of for-statement

```
for(expression1;expression2;expression3)
  a statement/a block
```

Semantics of for-statement

1 to evaluate expression1

2 to evaluate expression2

- if the value of expression2 is not 0,
    - to execute a statement/a block
    - to evaluate expression3
    - to go to 2
- otherwise, done

An example of `for`-statement

Example (sum from 1 to 100)

```c
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 0

The value of sum: 0

An example of `for`-statement

Example (sum from 1 to 100)

```c
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 1

The value of sum: 0

An example of `for`-statement

Example (sum from 1 to 100)

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 1

The value of sum: 0

An example of `for`-statement

Example (sum from 1 to 100)

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 1

The value of sum: 1

An example of `for`-statement

Example (sum from 1 to 100)

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 2

The value of sum: 1

An example of `for`-statement

```c
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of `i`:   2

The value of `sum`: 1

An example of `for`-statement

Example (sum from 1 to 100)

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 2

The value of sum: 3

An example of `for`-statement

Example (sum from 1 to 100)

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 3

The value of sum: 3

An example of `for`-statement

```c
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 3

The value of sum: 3

An example of `for`-statement

```
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

The value of i: 3

The value of sum: 6

Calculate the sum from 1 to 100 by `for`-statement

```
1 for(i=1;i<=100;i++){
2    sum+=i;
3 }
```

Code 1

```
1 for(i=1;i<=100;){
2    sum+=i++;
3 }
```

Code 2

```
1 i=1;
2 for(;i<=100;){
3    sum+=i++;
4 }
```

Code 3

```
1 i=1;
2 for(;;){
3    sum+=i++;
4    if(i>100){
5       break;
6    }
7 }
```

Code 4

```
1 i=1;
2 for(;;){
3    sum+=i;
4    if(i<100){
5       i++;
6    }else{
7       break;
8    }
9 }
```

Code 5

All of them are equivalent!

Relation between `while`-statement and `for`-statement

Example (sum from 1 to 100)

```c
for(i=1;i<=100;i++){
    sum=sum+i;
}
```

Example (Write a `while`-statement for the above code)

```c
i=1;
while(i<=100){
    sum=sum+i;
    i++;
}
```

for-statement

```
1  for(expression1;expression2;expression3)
2     a statement/a block
```

is equivalent to

for-statement

```
1  expression1;
2  while(expression2){
3    a statement/a block
4    expression3;
5  }
```

Question: Why do need both `for` and `while`?

do-while **statement**

Syntax of do-while statement

```
1  do
2    a statement/a block
3  while(expression);
```

Semantics of do-while statement

1. to execute a `statement`/a `block`
2. to evaluate expression
   - if the value of expression is not 0, go to 1
   - otherwise, done!

Calculate the sum from 1 to 100 by `do-while`-statement

```
1 i=1;
2 do{
3    sum+=i;
4    i++;
5 }while(i<=100);
```
Code 1

```
1 i=1;
2 do{
3    sum+=i++;
4 }while(i<=100);
```
Code 2

```
1 i=1;
2 do{
3    ;
4 }while(sum+=i++,i<=100);
```
Code 3

```
1 i=1;
2 do{
3   sum+=i++;
4   if(i>100){
5     break;
6   }
7 }while(1);
```
Code 4

```
1 i=1;
2 do{
3   if(i<=100){
4     sum+=i++;
5   }else{
6     break;
7   }
8 }while(1);
```
Code 5

All of them are equivalent!

goto-**statement**

Syntax of goto-statement

goto label;

Semantics of goto-statement

Go to the statement labeled by label and continue to execute until the function is finished.

Label of a statement

label : a statement/a block

Remark: goto-statement is rarely used in practice.

Summary of Chapter 3:

Control flows:

1. `if-statement`
2. `switch-statement`
3. `while-statement`
4. `for-statement`
5. `do-while-statement`
6. `break and continue`
7. `goto-statement`