CCS0023/L

Object Oriented Programming (Java)

Flow Controls

04

LESSON #4

Flow Controls

JAVA CONTROL STRUCTURES

if, else statement
switch, case statement
while loop
do, while loop
for loop

if-else Statement

The ***if*** statement enables your program to selectively execute other statements, based on some criteria.

The ***else*** statement performs a different set of statements if the expression is false.

Syntax:

```
if(condition/boolean expression) {
    //codes to execute if the condition is true
}
else {
    //codes to execute if the condition is false
}
```

switch Statement

The ***switch*** statement is used to conditionally perform statements based on an integer expression.

Syntax:

```
switch(varName)
{
    case const1: codes here; break;
    case const2: codes here; break;
    default: codes here; break;
}
```

Keywords

The ***break*** statement causes the program flow to exit from the body of the switch construct.

Each case label takes only a single argument, but when execution jumps to one of these labels, it continues downward until it reaches a break statement.

Keywords

The ***default*** keyword is comparable to the else part of the if statement.

The statements associated with this keyword is executed if the value of the switch variable does not match any of the case constants.

while Statement

The ***while*** loop executes a given statement or block of statements repeatedly as long as the value of the expression is true.

Syntax:

```
while(condition/expression) {
    //codes to execute if condition is true
}
```

do-while Statement

The ***do-while*** loop facilitates evaluation of condition or expression at the end of the loop to ensure that the statements are executed at least once .

Syntax:

```
do{
        //codes to execute if condition is true
} while(condition/expression);
```

for Statement

The ***for*** loop executes a given statement or a block of statements for a definite number of times.

Syntax:

```
for (initialization; condition; altering list) {
    //codes to execute if condition is true
}
```

foreach Statement

The ***foreach*** loop is for traversing items in a collection. ***foreach*** is usually used in place of a standard for statement. It usually maintain no explicit counter: they essentially say "do this to everything in this set", rather than "do this x times".

Syntax:

```
for(initialization : [collection/array]) {
        //codes to execute if condition is true
}
```

foreach Loop Example

```
int quizzes[ ] = {100,90,80};
for (int grade : quizzes) {
      System.out.print(grade+”\t”);
}
```

OUTPUT:

```
100   90     80
```

The following are used to further control the loop statements:

The ***break*** statement causes the program flow to exit prematurely from the body of the loop statement.

Syntax: break [label];

The following are used to further control the loop statements:

The ***continue*** statement causes the program flow to skip over and jump to the end of the loop body, and then return the control to the loop control statement.

Syntax: continue [label];

The following are used to further control the loop statements:

The ***label*** identifies any valid statement to which control must be transferred

Syntax: [label:] statements;

THANK YOU