AP CSA

2022/10/01

Quiz from last class

11. Which of the following will evaluate to true only if boolean expressions `A`, `B`, and `C` are all false?

(A) `!A && !(B && !C)`
(B) `!A || !B || !C`
(C) `!(A || B || C)`
(D) `!(A && B && C)`
(E) `!A || !(B || !C)`

Today’s topic

1. Output statement
2. Control structure
3. Loop

Print and println

```
System.out.print("Hot");
System.out.println("dog");
```
prints Hotdog

```
System.out.println("Hot");
System.out.println("dog");
```
prints
Hot
dog

```
System.out.println(7 + 3);
```
prints 10

```
System.out.println(7 == 2 + 5);
```
prints true

```
int x = 27;
System.out.println(x);
System.out.println("Value of x is " + x);
```
prints 27

prints Value of x is 27

Escape Sequence

An *escape sequence* is a backslash followed by a single character. It is used to print special characters. The three escape sequences that you should know for the AP exam are

Escape Sequence	Meaning
\n	newline
\"	double quote
\\	backslash

Here are some examples:

```
System.out.println("Welcome to\na new line");
```

prints

```
Welcome to
a new line
```

The statement

```
System.out.println("He is known as \"Hothead Harry\".");
```

prints

```
He is known as "Hothead Harry".
```

The statement

```
System.out.println("The file path is d:\\myFiles\\..");
```

prints

```
The file path is d:\myFiles\..
```

Pop quiz1

2. What output will be produced by

```
System.out.print("\\* This is not\n a comment *\\");
```

(A)
```
* This is not a comment *
```

(B)
```
\* This is not a comment *\
```

(C)
```
* This is not
a comment *
```

(D)
```
\\* This is not
a comment *\\
```

(E)
```
\* This is not
a comment *\
```

Simple if statement

THE `if` STATEMENT

```
if (boolean expression)
{
    statements
}
```

Here the *statements* will be executed only if the *boolean expression* is `true`. If it is `false`, control passes immediately to the first statement following the `if` statement.

THE `if...else` STATEMENT

```
if (boolean expression)
{
    statements
}
else
{
    statements
}
```

Here, if the *boolean expression* is `true`, only the *statements* immediately following the test will be executed. If the *boolean expression* is `false`, only the *statements* following the `else` will be executed.

Nested if statement

If the statement part of an `if` statement is itself an `if` statement, the result is a *nested* `if` *statement.*

Example 1

```
if (boolean expr1)
    if (boolean expr2)
        statement;
```

This is equivalent to

```
if (boolean expr1 && boolean expr2)
    statement;
```

Pop quiz2

Suppose you want to print the integer if it is positive and even, and if the integer is not positive, print that it is not positive

```
Int x = 7;

If (x >0) {

If (x % 2 == 0) {

System.out.print(x)

}

}

else (x <= 0) {

System.out.print(“not positive”)

}
```

Problem of the following code

```
int n = IO.readInt();          //read user input
if (n > 0)
    if (n % 2 == 0)
        System.out.println(n);
else
    System.out.println(n + " is not positive");
```

A user enters 7 and is surprised to see the output

```
7 is not positive
```

The reason is that `else` always gets matched with the *nearest* unpaired `if`, not the first `if` as the indenting would suggest.

```
if (x > 0) {
    if (x % 2 == 0) {
        System.out.print(x);
    }
}
else {
    System.out.print(s: "not a positive");
}
```

Pop quiz3

9. What will the output be for the following poorly formatted program segment, if the input value for num is 22?

```
int num =  call to a method that reads an integer;
if (num > 0)
if (num % 5 == 0)
System.out.println(num);
else System.out.println(num + " is negative");
```

(A) 22
(B) 4
(C) 2 is negative
(D) 22 is negative
(E) Nothing will be output.

Extended if statement

EXTENDED `if` STATEMENT

For example,

```
String grade = IO.readString();          //read user input
if (grade.equals("A"))
    System.out.println("Excellent!");
else if (grade.equals("B"))
    System.out.println("Good");
else if (grade.equals("C") || grade.equals("D"))
    System.out.println("Poor");
else if (grade.equals("F"))
    System.out.println("Egregious!");
else
    System.out.println("Invalid grade");
```

If any of `A`, `B`, `C`, `D`, or `F` are entered, an appropriate message will be written, and control will go to the statement immediately following the extended `if` statement. If any other string is entered, the final `else` is invoked, and the message `Invalid grade` will be written.

For Loop

THE for LOOP

The general form of the for loop is

```
for (initialization; termination condition; update statement)
{
    statements          //body of loop
}
```

The termination condition is tested at the top of the loop; the update statement is performed at the bottom.

Example 1

```
//outputs 1 2 3 4
for (i = 1; i < 5; i++)
    System.out.print(i + " ");
```

Termination condition is tested before entering the loop.

Update is performed after execution of content in the loop

Pop quiz4

```
for (k = 20; k >= 15; k--)
    System.out.print(k + " ");
```

```
for (j = 2; j <= 10; j += 2)
    System.out.print(j + " ");
```

Important Note about for loop

NOTE

1. The loop variable should not have its value changed inside the loop body.
2. The initializing and update statements can use any valid constants, variables, or expressions.
3. The scope (see p. 100) of the loop variable can be restricted to the loop body by combining the loop variable declaration with the initialization. For example,

```
for (int i = 0; i < 3; i++)
{
    ...
}
```

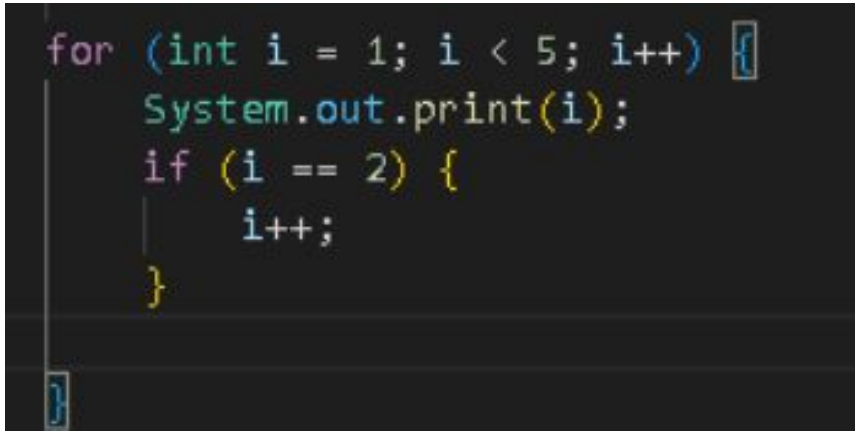
4. The following loop is syntactically valid:

```
for (int i = 1; i <= 0; i++)
{
    ...
}
```

The loop body will not be executed at all, since the exiting condition is true before the first execution.

Pop Quiz5

What will be printed?



While Loop

The general form of the `while` loop is

```
while (boolean test)
{
    statements          //loop body
}
```

The *boolean test* is performed at the beginning of the loop. If `true`, the loop body is executed. Otherwise, control passes to the first statement following the loop. After execution of the loop body, the test is performed again. If true, the loop is executed again, and so on.

Example 1

```
int i = 1, mult3 = 3;
while (mult3 < 20)
{
    System.out.print(mult3 + " ");
    i++;
    mult3 *= i;
}                          //outputs 3 6 18
```

Important note about while loop

NOTE

1. It is possible for the body of a `while` loop never to be executed. This will happen if the test evaluates to `false` the first time.
2. Disaster will strike in the form of an infinite loop if the test can never be false. Don't forget to change the loop variable in the body of the loop in a way that leads to termination!

The body of a `while` loop must contain a statement that leads to termination.

Example 2

```
int power2 = 1;
while (power2 != 20)
{
    System.out.println(power2);
    power2 *= 2;
}
```

Since `power2` will never exactly equal 20, the loop will grind merrily along eventually causing an integer overflow.

Nested Loop

NESTED LOOPS

You create a *nested loop* when a loop is a statement in the body of another loop.

Example 1

```
for (int k = 1; k <= 3; k++)
{
    for (int i = 1; i <= 4; i++)
        System.out.print("*");
    System.out.println();
}
```

Think:

```
for each of 3 rows
{
    print 4 stars
    go to next line
}
```

Output:

```
****
****
****
```

Pop quiz6

15. Given that n and count are both of type int, which statement is true about the following code segments?

I
```
for (count = 1; count <= n; count++)
    System.out.println(count);
```

II
```
count = 1;
while (count <= n)
{
    System.out.println(count);
    count++;
}
```

(A) I and II are exactly equivalent for all input values n.
(B) I and II are exactly equivalent for all input values $n \geq 1$, but differ when $n \leq 0$.
(C) I and II are exactly equivalent only when $n = 0$.
(D) I and II are exactly equivalent only when n is even.
(E) I and II are not equivalent for any input values of n.

Pop quiz7

23. What output will be produced by this code segment? (Ignore spacing.)

```
for (int i = 5; i >= 1; i--)
{
    for (int j = i; j >= 1; j--)
        System.out.print(2 * j - 1);
    System.out.println();
}
```

(A)
```
9 7 5 3 1
9 7 5 3
9 7 5
9 7
9
```

(B)
```
9 7 5 3 1
7 5 3 1
5 3 1
3 1
1
```

(C)
```
9 7 5 3 1
7 5 3 1 -1
5 3 1 -1 -3
3 1 -1 -3 -5
1 -1 -3 -5 -7
```

(D)
```
1
1 3
1 3 5
1 3 5 7
1 3 5 7 9
```

(E)
```
1 3 5 7 9
1 3 5 7
1 3 5
1 3
1
```

Homework

Part1: multiple choice questions in the book: 10, 14, 20, 24, 25

Part2: Coding question.