

SAT PREP

Assignment : AP Computer Science: Control Structure

Iteration

Java has three different control structures that allow the computer to perform iterative tasks: the for loop, while loop, and do...while loop. The do...while loop is not in the AP Java subset.

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.

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.

Example

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

THE 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.

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!

Example :

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

NESTED LOOPS

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

Example

```
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:

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