Java Puzzle: unreachable statement

This is an interesting puzzle to test our understanding on java compiler. What happens when we compile below code blocks?

CODE A:

```
public void javapapers() {
    System.out.println("java");
    return;
    System.out.println("papers");
}
```

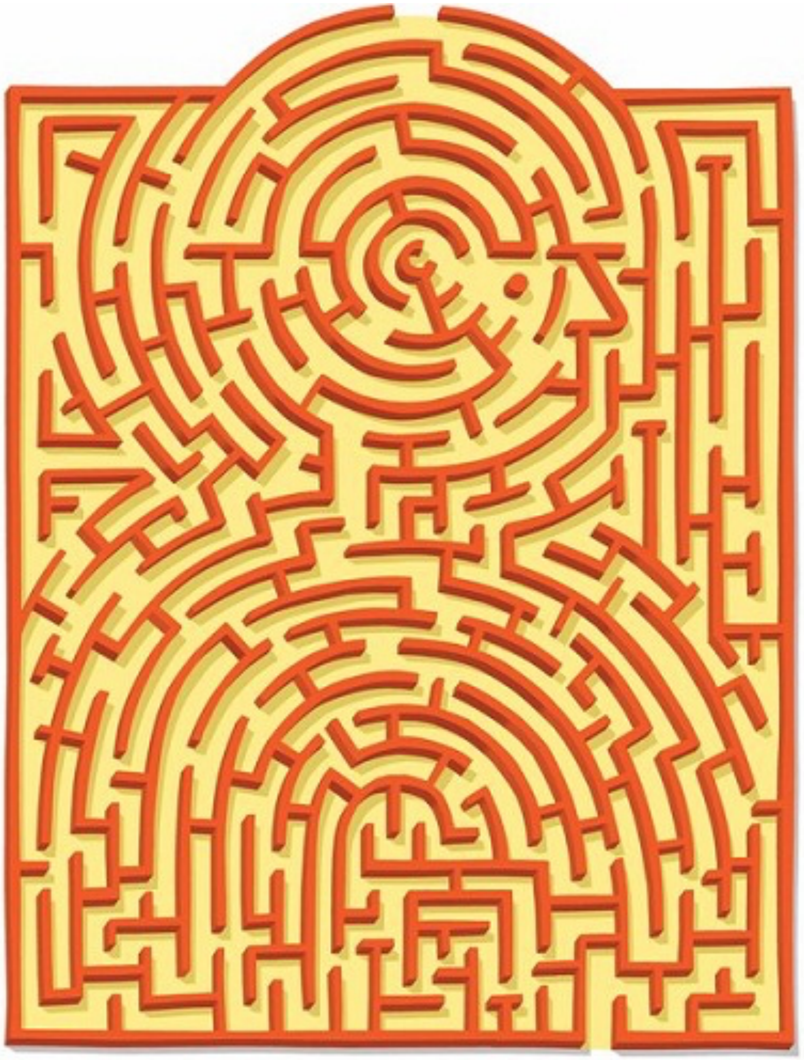
CODE B:

```
public void javapapers() {
    System.out.println("java");
    if(true) {
        return;
    }
    System.out.println("papers");
}
```

CODE C:

```
public void javapapers() {
    System.out.println("java");
    while(true) {
        return;
    }
    System.out.println("papers");
}
```

Got the answer? More than answer let us try to understand the reasons behind it.



Puzzle Solution

CODE A Output:

```
TestCompiler.java:5: unreachable statement
System.out.println("papers");
^
1 error
```

CODE A results in above compilation error and the reason is also clearly given as 'unreachable statement'. We have a return followed by first system.out.println and after that line whatever code is there it will be unreachable and so this error.

CODE B Output:

Compiles successfully.

CODE C Output:

```
TestCompiler.java:7: unreachable statement
System.out.println(“papers”);
^
1 error
```

Code C results in above compilation error and the reason is unreachable statement. Same as in Code A.

We all know very well about unreachable statement will result in compilation error. It is a mandatory rule in java language specification that unreachable statement should be an error.

Okay what is strange here?

Code B and C are similar than Code A. If Code B compiles successfully, Code C should also compile.

The word Unreachable is defined in java language specification for every block like if, for, while, etc individually.

> *The block that is the body of a constructor, method, instance initializer, or static initializer is reachable…*

like the above JLS goes on to define unreachable for all blocks including while, for, do, if, switch, etc separately.

The most general logic in these rules are, code surrounded by a block is unreachable if the conditional expression deciding it evaluates to false and then the block is decided as unreachable. Then it becomes an error.

Only for ‘if’ construct the rule is different. The conditional expression is not evaluated and decided. Whatever may be the expression inside the if condition, it is not evaluated to decide if the code block is reachable and so we will not get error even if we have a constant value ‘false’ as conditional expression in if statement. This is treated special to give a facility for the developer. Developers can use a flag variable in the if statement with default final values as false or true. Just by changing the flag variable from false to true or vice-versa developers can recompile the program and use it differently.

This Core Java tutorial was added on 09/10/2012.

Web Service Introduction Tutorial

Android DatePicker