# Chapter 7. The branch operator of the switch algorithm

## 7.1 Syntax of the switch operator

The switch statement is designed to branch the calculation process into several directions – an extended version of if-else.

The program uses the keyword "switch", followed by a switching expression - it must be an integer, and then lists the actions that are performed with certain values of this expression [4].

The switch syntax starts with the “switch” keyword, followed by the expression that needs to be checked. This is followed by a block of code enclosed in curly brackets.

Several case statements are used inside the code block, each of which corresponds to a specific variable value. The switch statement provides several useful functions. Numbers, symbols, or enumerations can be used as case values. The “default” keyword can also be used, which will be executed if none of the case values matches the variable being checked. (Figure 7.1).

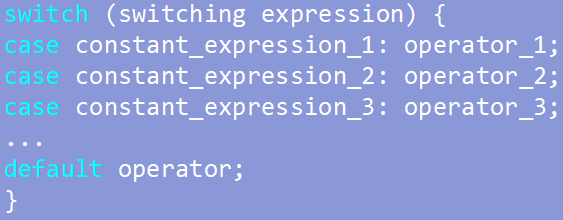


Figure 7.1 – Syntax of the switch statement

A variable can be declared inside the case block, but initialization is not allowed. To initialize a variable, a value must be assigned to a memory area, and execution of a case containing initialization may not be necessary. (Figure 7.2).

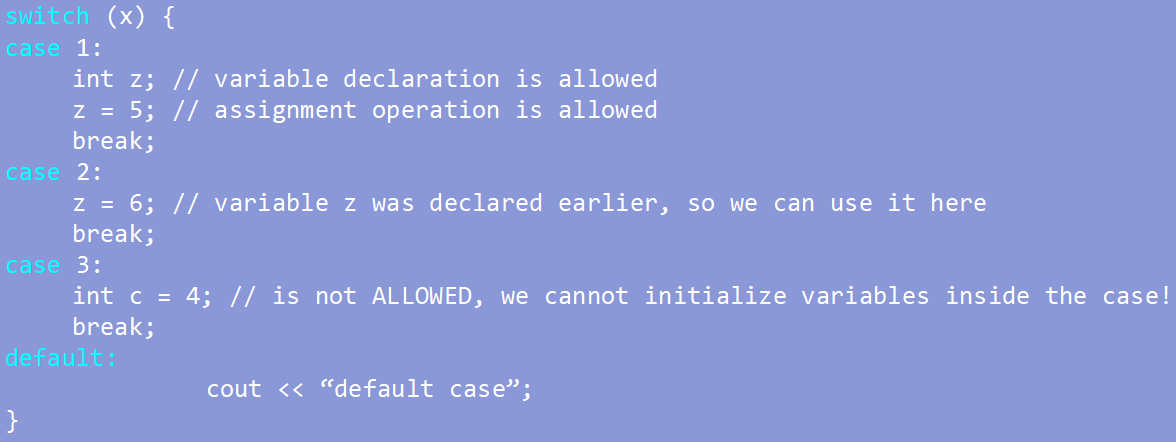


Figure 7.2 – Working with variables in the switch statement

## 7.2 Rules for working with the switch operator

1. Control is transferred to the branch for which the constant expression matches the switching one. If there is no match, then the default operator is executed, which can only be one.

2. The value of the expression being checked must be of an integer type or an enumeration type.

3. If there is no default value and there were no matches, then the compiler completely ignores the switch and executes the algorithm further.

4. If there is no exit or transition inside the case, then all statements are executed sequentially after it, starting from the case to which control was transferred.

5. Each block of the case code must be terminated with a break statement to prevent subsequent blocks from being executed (figure 7.3) [27].

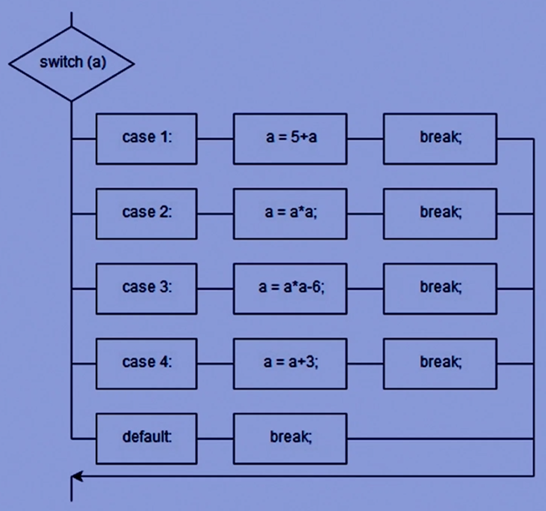


Figure 7.3 – Block diagram of the algorithm for executing the switch

branching operator

## 7.3 The break operator

The break statement allows for the completion of the execution of a block of switch statement code when a condition is met in one of the case blocks. This prevents subsequent blocks of code from being executed [29].

This statement terminates the execution of the switch branch. In this case, control is transferred to the first operator following the switch construction. Without the break statement, all code blocks will be executed after the corresponding case [18].

## 7.4 Practical tasks

To consolidate the material, three tasks presented below need to be completed.

Task 1:

What will be output as a result of the program execution? (Figure 7.4)

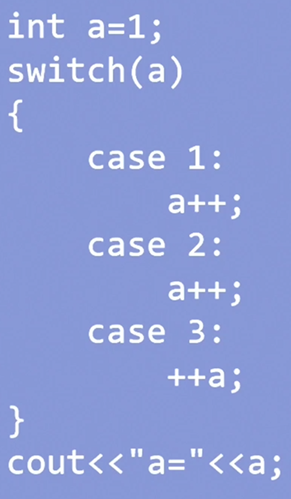


Figure 7.4 – A fragment of the program code for task 1

Task 2:

What will be output as a result of the program execution? (Figure 7.5)

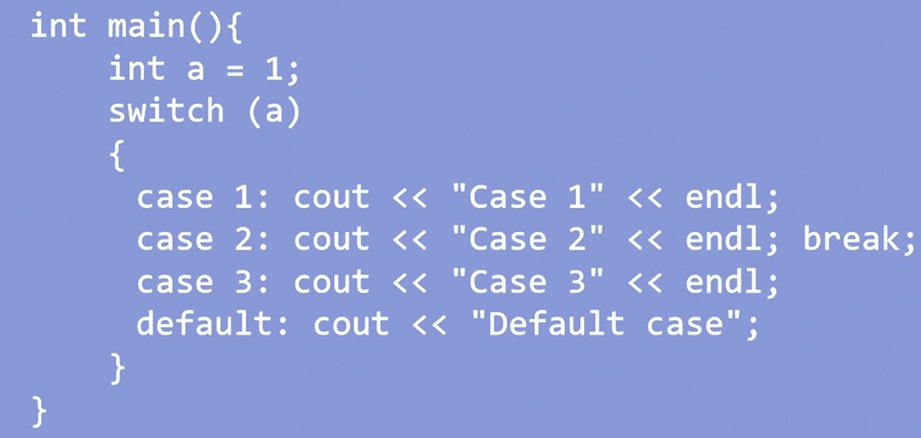


Figure 7.5 – A fragment of the program code for task 2

Task 3:

What will be output at ch = C, at ch = A, at ch = a? (Figure 7.6)

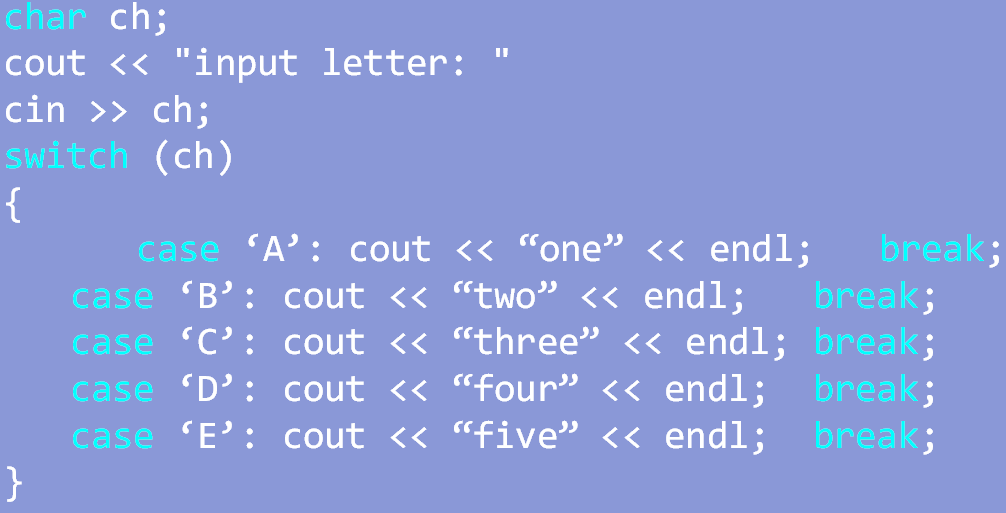


Figure 7.6 – A fragment of the program code for task 3