# More on *switch*

**Consider this code fragment inside a `switch`:**

```
case 'a':
case 'e':
case 'i':
case 'o':
case 'u':
    cout << "vowel";
    break;
```

As you can see, `break` statements are **not required** at the end of each `case`. If the `break` is missing, the program continues executing the **next clause** after it finishes the selected one. We say the `case` **falls-through**.

This is useful as shown here where the output is printed for all of the lower-case vowels.

```
case ' ': case '\t': case '\n':
    cout << "whitespace";
    break;
```

If there is nothing in the body of the case, it may be more readable to format it like the whitespace block shown here.

If there is **any code** inside a `case` block that falls through, most compilers will issue a warning. If you **intend** to fall through, and you want to suppress the warning, add a comment like this, just before the second case:

```
// fall through
```

## A Few More Rules

- Two `case` labels may not have the same value
- A label must precede a statement or another `case` label. It may not be alone.
- Variables **may not** be defined inside one `case` block and then used in another.