



# String Value Parameters

Imagine you want to write a function named `count_vowels()`, which counts the number of vowels in a `string`. Here's a first attempt:

```
int count_vowels(string str) {  
    int vowels = 0;  
    for (char c : str) {  
        switch (c) {  
            case 'a': case 'A': case 'e': case 'E': case 'i':  
            case 'I': case 'o': case 'O': case 'u': case 'U':  
                vowels++;  
        }  
    }  
    return vowels;  
}
```

The code in this function is correct, readable, and quite efficient. However, it has **one flaw**. Imagine calling the function with a long `string`, say the text of *War and Peace*. Because the parameter variable `str` is a **value** parameter, your code will make a copy of the whole text of the book and store that in `str`.

```
string book =  ;  
int vowels = count_vowels(book);  
int count_vowels(string  str)
```

Thus, using pass-by-value with `string` arguments is **very inefficient**.

*Never pass class types, such as `string` and `vector` by value.*



This course content is offered under a [CC Attribution Non-Commercial](#) license. Content in this course can be considered under this license unless otherwise noted.