

# Video 1 - Explains

## Coding workshop: implement tokenise

In this worksheet, you will implement a tokenise function in C++ from a pseudocode representation.

### The pseudocode:

```
string csvLine = 'thing,thing2,thing3' ## the string we are processing
string vector tokens ## stores the tokens function
char separator = ',' ## the character that separates the tokens

int start, end ## used to delineate the position of the tokens

start = csvLine.find_first_not_of(separator)
do
end = next 'separator' after start
if start == csvLine.length or start == end ## nothing more to find
    break
if end >= 0 ## we found the separator
    token = csvLine.substr(start, end - start) ## start, substring length
else
    token = csvLine.substr(start, csvLine.length - start) ## end is invalid

add token to the end of the tokens vector
start = end + 1 ## move past this token

while (end > 0) ## continue loop condition
```

Now convert that into a C++ function, with the following signature:

```
std::vector<std::string> tokenise(std::string csvLine, char separator);
```

Here is a complete main file that you can use to test your function - you could comment out whatever you have in your main file and put this below it, if you don't want to start a fresh project:

```
#include <iostream>
#include <vector>
#include <string>

/** function prototype*/
std::vector<std::string> tokenise(std::string csvLine, char separator);

/** function implementation */
```

```

std::vector<std::string> tokenise(std::string csvLine, char separator)
{
    std::vector<std::string> tokens;
    // put your implementation of the pseudocode here!

    return tokens;
}

int main()
{
    std::string s = "hello,I,have three tokens";
    std::vector<std::string> tokens = tokenise(s, ',');
    // note the use of const and & below
    // const says I won't edit t
    // & says I don't want to copy t from one of the tokens
    // I want to access the original version of the token
    for (const std::string& t : tokens)
    {
        std::cout << t << std::endl;
    }
}

```

## Test the function

Once you have your implementation of tokenise, test it out by sending it different strings.

## Conclusion

In this worksheet, you have implemented a tokenise function in C++ from a pseudocode representation.