# SFT221 – Workshop 1

## Learning Outcomes

- Understand the difference between testing and debugging.
- How to do simple software tests.
- How to do simple debugging.
- Appreciate the difficulty in testing and debugging.

## Instructions

The following code contains a function to determine if a string starts with another string and a second function to test if a string ends with another string. The developer wrote two tests for the functions and concluded that they work.

Your job is to:

- Test the code to determine if it works correctly:
  - Show the test code for every test and describe what you are testing.
  - Present the result of the test and your interpretation of the result and an explanation of how this result happened if the result was not what you expected.
  - Inspect the code to find any bugs not revealed in testing.
- Debug the code:
  - Use whatever tools you like to determine the cause of an errors you found in the code regardless of the means you used to find the error.
  - Fix the code and explain what the error was and how you fixed it.

Here is the code you will be working on.

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

/**
* Determine if the string s starts with the string prefix.
* @param s - the string to see if it starts with prefix
* @param prefix - the prefix to test
* @returns true if the strings begins with the prefix.
*/
int startsWith(const char s[], const char prefix[])
{
    char buf[25];
    int i;

    int sz = strlen(prefix);
    for (i = 0; i < sz; i++)
    {
        buf[i] = s[i];
    }
    buf[sz] = '\0';

    return 0 == strcmp(buf, prefix);
}
/**
* Determine if the string s ends with the string suffix.
* @param s - the string to see if it ends with suffix
* @param suffix - the suffix to test
* @returns true if the strings ends with the suffix.
*/
int endsWith(const char s[], const char suffix[])
{
    int sz = strlen(suffix);
    int slen = strlen(s);

    return 0 == strcmp(s + slen - sz, suffix);
}

int main(void)
{
    char s1[] = { "upended" };
    char prefix[] = { "up" };
    char suffix[] = { "ed" };

    printf("%s does %s start with %s\n",
        s1, startsWith(s1, prefix) ? "" : " not", prefix);

    printf("%s does %s end with %s\n",
        s1, endsWith(s1, suffix) ? "" : " not", suffix);

    return 0;
}
```

# Deliverables

## Due Date:

This workshop is due **at 11:59 pm 2 days after your lab day**. Late submits will not be accepted.

Explanations should not be trivial but state where the bug is and under what conditions it can occur. Similarly, explanation of bug fixes should explore the bug in depth, state when it could happen and how the fix to the code will prevent it from happening in the future. Reflections should consider the questions in depth and not be trivial. Some reflections might require research to properly answer the question. As a rough guideline, the answer to each reflection questions should be about 100 words in length.

You should submit:

## Testing and Debugging Results

- A list of the tests you performed including:
    - The code for the test (which might be a single function call with parameters)
    - A description of what you were testing for
    - The result of the test
    - Your interpretation of the test result and an explanation of what happened if the result was not what you expected.
    - State whether this test has identified a bug or not.
- After inspecting the code:
    - Identify any additional bugs you found,
    - If you wrote a test to verify this was a bug, include the code,
    - Explain how this bug could cause erroneous result or program failure.
- For every bug identified in the first two sections:
    - State which bug you are fixing,
    - Show the code before and after you fixed it. Do not repeat all of the code, but just the lines modified to fix this particular bug.
    - An explanation of how your modification fixes the bug.

## A Reflection, Research and Assessment

- A reflection where you consider whether testing or inspection identified more bugs in this case. State why you think one way worked better than the other. How could you improve the technique that worked less well?
- Did you find it difficult to find the bugs in this assignment? If not, what helped find them quickly? If you did find it difficult, what made finding the bugs so difficult?

# Marking Rubric

| Comprehensiveness of test cases | 10% |
|---|---|
| Explanation of erroneous test results | 15% |
| Explanation of bugs discovered by inspection | 15% |
| Discovered at least 4 bugs | 15% |

| Quality of bug fixes | 10% |
|---|---|
| Explanation of how bug fix works | 15% |
| Reflection | 20% |