

University of Lincoln
School of Computer Science
CMP2801M – Advanced Programming
Workshop 4

Task 1:

Following the swap function exercises in the lecture: *To develop a swap function to swap two variables.*

Testing case: e.g. assume the inputs are int a= 2, and int b=3, the expected results of the function swap(a,b) is a ==3, b== 2.

You are required to debug the provided source code (swap.cpp) uploaded on the blackboard. You are also required to complete the function `void swapv(int x, int y)` with the call-by-value

Source code:

```
#include <iostream>
using namespace std;

//to be completed
void swapv(int x, int y) {
}

void swapr(int* x, int* y) {
    int aux = *x;
    *x = *y;
    *y = aux;
}

void swap(int& x, int& y) {
    int aux = x;
    x = y;
    y = aux;
}

void (*pf)(int& x, int& y);
int main()
{
    int a = 2;
    int b = 3;
    printf("%d,%d \n", a, b);
    swapr(&a, &b);
    printf("%d,%d \n", a, b);
    // void (*pf)(int& x, int& y) = swapr;
    pf = &swap;
    pf(a, b);
    printf("pf %d,%d \n", a, b);
    return 0;
}
```

Task2:

In this workshop, you are asked to write a program that represents the summary of a given document. You will practice several facilities introduced in the lecture; these include pointers, references, Stack and Heaps, functions, overloaded functions. More specifically, you should apply the pointer to access the array/vector to count the words/sentences. You are encouraged to write the following functions to summarize the document.

Functions	Input parameters	Return value	Description
printSummary	void	void	Prints the number of sentences, the number of words and each word with its frequency (see format at the end of the assignment)
updateSentenceCount	std::string word	void	Takes a word as input and increases the variable numberOfSentences if the given word contains a “.”, “!” or “?” symbol.
removePunc	string & word	void	Remove the punctuations
addWord	std::string	void	Adds a given word into the wordList vector if it does not exist in the vector. Otherwise, it increases the corresponding count entry in vector.
analyseDocument	void	void	Reads the document word by word and calls addWord, removePunc, updateSentenceCount

Please use the following table as a reference, to define and use the variables

Variables	type	Description
word	String	is a string that holds a word
Count	Integer	is an integer that holds the number of occurrences of the word in the document
numberOfWords	Integer	is an integer that indicates the number of words in the document
numberOfSentences	Integer	is an integer that indicates the number of sentences in the document

filename	string	is a string that shows the name of the document
wordList	vector	is a vector of WordFreqPair objects. Each entry of the vector holds a WordFreqPair object that keeps the word and its count.

If you want to challenge yourself further, you can update your code to work with classes.

A test input file is given to evaluate your program. Your output should have the same format given below.

```

number of sentences is 3
number of words is 39
Today 1
is 2
Sunday 1
For 1
the 4
first 2
time 2
they 1
took 1
me 1
out 1
into 1
sun 1
today 1
And 1
for 1
in 1
my 1
life 1
I 1
was 1
aghast 1
that 1
sky 1
so 3
far 1
away 1
and 2
blue 1
vast 1

```

Task 3:

Following the work in task1, you are asked to write two functions: `getStatistics` and `searchWord`. You will practice call-by-value and call-by reference in this task.

The `getStatistics` function determines the **count of each distinct string**. The function declaration is as follows:

```
void getStatistics( vector<string> list,  
                  vector<string>& listOfDistinctWords,  
                  vector<int>& listOfCounts)
```

The inputs to the function are:	
list:	The whole list of strings read from the file
listOfDistinctWords	An output parameter, which will only hold the list of distinct strings
listOfCounts	An output parameter, which will hold the corresponding count of each word in the listOfDistinctWords

A `searchWord` function will search for a word in your list and return the count of the searched word (0 if the word does not exist in the list). The function declaration is as follows

```
int searchWord ( string searchedWord,  
                vector<string> listOfDistinctWords,  
                vector<int> listOfCounts)
```

Testing result example:

Please enter a word to search: sun