Swinburne University Of Technology

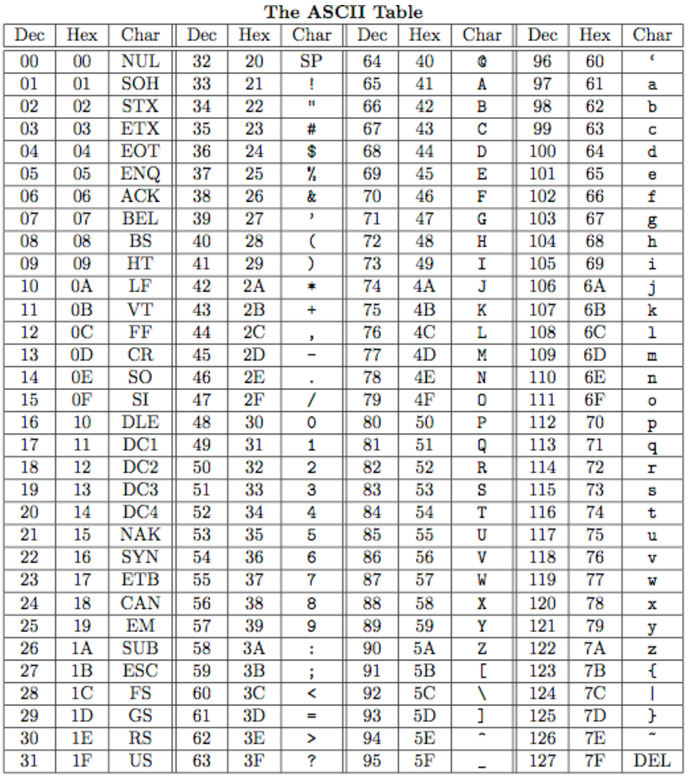
Faculty of Science, Engineering and Technology

LABORATORY COVER SHEET

Subject Code:	COS30008
Subject Title:	Data Structures and Patterns
Lab number and title:	2, Basic I/O
Author:	Dr. Markus Lumpe
Edited by:	Carmen Chai

A journey of a thousand miles begins with a single step.

Lao Tsu



The ASCII Table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
00	00	NUL	32	20	SP	64	40	@	96	60	`
01	01	SOH	33	21	!	65	41	A	97	61	a
02	02	STX	34	22	"	66	42	B	98	62	b
03	03	ETX	35	23	#	67	43	C	99	63	c
04	04	EOT	36	24	$	68	44	D	100	64	d
05	05	ENQ	37	25	%	69	45	E	101	65	e
06	06	ACK	38	26	&	70	46	F	102	66	f
07	07	BEL	39	27	'	71	47	G	103	67	g
08	08	BS	40	28	(	72	48	H	104	68	h
09	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Basic I/O in C++

The goal of this laboratory session is to develop a small Win32 console application that counts the occurrences of each printable non-whitespace character in a given input text stream. The application consists of two parts: a class `CharacterCounter` and a `main` function that drives the counting process.

CharacterCounter.h

Create a CharacterCounter.h in the Header Files folder and feel free to copy paste the codes above… or you can type it if you want…

The class `CharacterCounter` is specified as follows:

Inside CharacterCounter.h:

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

class CharacterCounter
{
private:
    int fTotalNumberOfCharacters; //to store the total number of characters
in the char array
    int fCharacterCounts[256];    // We count all 256 byte values

public:
    CharacterCounter(); //set everything to 0

    void count(char* str); //reset character counts, loop through char
array to increase count

    //print out both fCharacterCounts and fTotalNumberOfCharacters
    friend ostream& operator<<(std::ostream& aOStream,
          const CharacterCounter& aCharacterCounter);
};
```

CharacterCounter.cpp

You will have to work out CharacterCounter.cpp based on the requirements below. However, the CharacterCounter.h and main.cpp codes have been provided for you in this document.

The class `CharacterCounter` records the total number of counted characters and the frequencies of those characters. The class has one constructor to properly initialize the data members, has a `count` method that takes a char array, and declares the stream output `operator<<` for `CharacterCounter` as a friend of class `CharacterCounter`.

The implementations of the constructor and the `count` method are straightforward. The constructor initializes all data members with 0 (remember to loop through the array to set each value to 0), whereas `count` has to increment the corresponding data members.

The **`operator<<`** should only print those characters that actually occur in the input text stream (i.e., you need to filter the characters with 0 count). Second, you need to use a simple trick to print an integer value as a character value. You can achieve this using the cast operator `(type)value`. For example, if your program defines an integer variable `lIntValue`, then `(char)lIntValue` yields a character value. Remember, there are 256 unsigned chars in the array so the range should be between 0 and 255.

main.cpp

The `main` function in main.cpp:

i) Declares an object of type `CharacterCounter`

ii) Declares a `char array` variable of size 50

iii) Prints out all 256 unsigned char

iv) Receives a char array input

v) Prints out the input received

vi) Performs the counting process with the char array

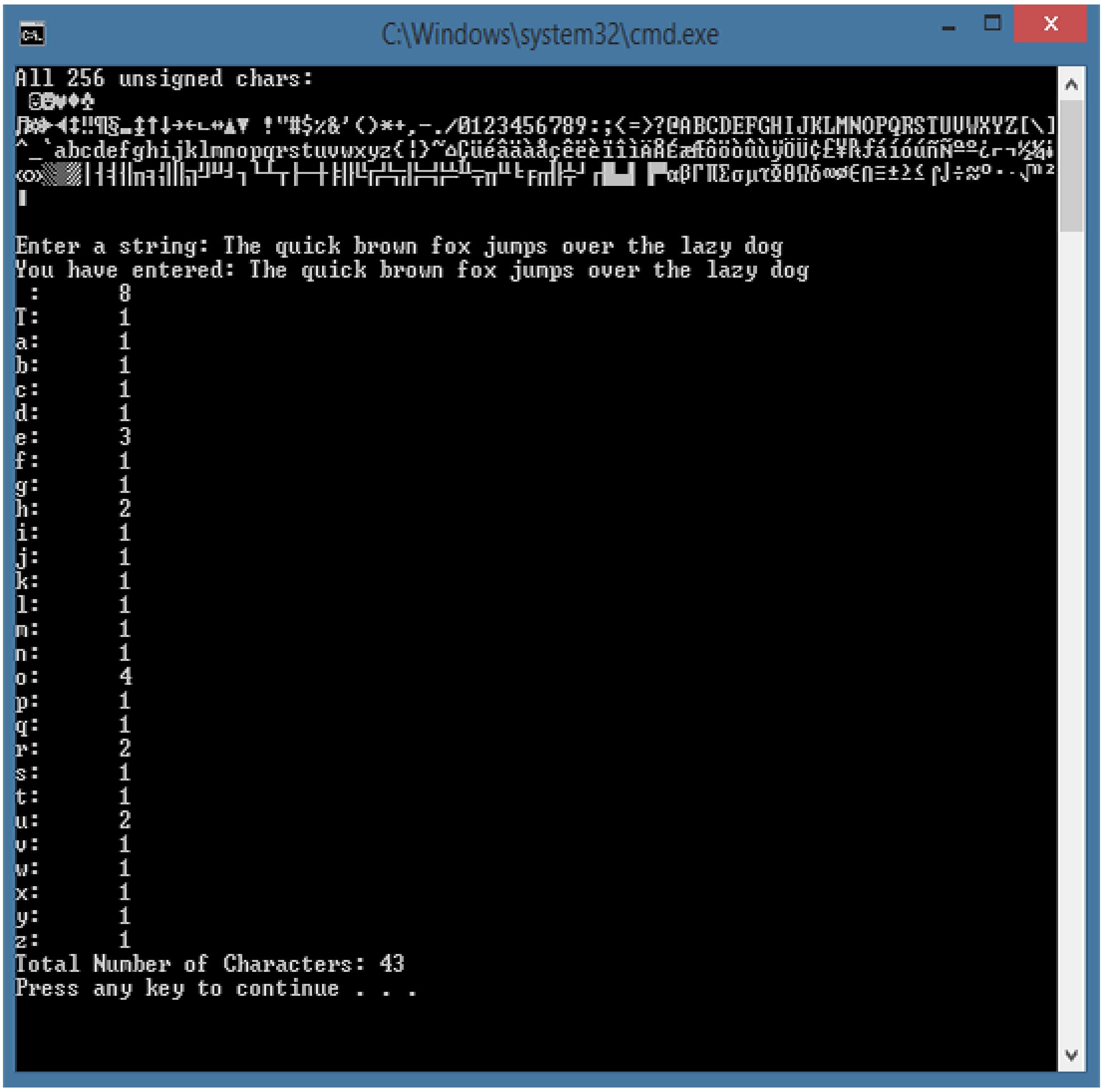
vii) Prints out the character count with the overloaded operator <<

Feel free to copy paste the code on the next page to your main.cpp… but it might be better if you try to work it out on your own first …

Inside main.cpp:

```
#include "CharacterCounter.h"
#include <iostream>

using namespace std;

int main() {
    CharacterCounter lCounter;

    const int arraySize = 50;
    char str[arraySize]; //char array of size 50

    //output all 256 unsigned chars
    cout << "All 256 unsigned chars:" << endl;

    for (int i = 0; i < 256; i++)
    {
        cout << (unsigned char)i;
    }

    cout << "\n" << endl;

    //test CharacterCounter class
    cout << "Enter a string: ";

    //cin >> str; //okay for single words
    cin.getline(str, arraySize);

    cout << "You have entered: " << str << endl;

    lCounter.count(str);
    cout << lCounter;

    return 0;
}
```

Console output example screenshot available on the next page.



```
All 256 unsigned chars:
 ☺☻♥♦♠
♫☼►◄↕‼¶§▬↨↑↓→←∟↔▲▼ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~⌂ÇüéâäàåçêëèïîìÄÅÉæÆôöòûùÿÖÜ¢£¥₧ƒáíóúñÑªº¿⌐¬½¼¡
«»░▒▓│┤╡╢╖╕╣║╗╝╜╛┐└┴┬├─┼╞╟╚╔╩╦╠═╬╧╨╤╥╙╘╒╓╫╪┘┌█▄▌▐▀αßΓπΣσµτΦΘΩδ∞φε∩≡±≥≤⌠⌡÷≈°∙·√ⁿ²
■

Enter a string: The quick brown fox jumps over the lazy dog
You have entered: The quick brown fox jumps over the lazy dog
 :      8
T:      1
a:      1
b:      1
c:      1
d:      1
e:      3
f:      1
g:      1
h:      2
i:      1
j:      1
k:      1
l:      1
m:      1
n:      1
o:      4
p:      1
q:      1
r:      2
s:      1
t:      1
u:      2
v:      1
w:      1
x:      1
y:      1
z:      1
Total Number of Characters: 43
Press any key to continue . . .
```