10 points.

This is a realistic first assignment: ENHANCE AN EXISTING

PROGRAM BY ADDING FUNCTIONALITY TO IT.

Assignment: You're given one program, startStd.cpp, which

uses Standard I/O (printf()). Alter it so it prints out

the Sizes of all basic Data Types, and the Local System's

ASCII Character Set and Collating Sequence.

This is the first program which all programmers must

run on a new combination of computer, operating system,

and compiler. It tells hir the size data objectscts

she's using.

Unlike Java, the international standards for C and C++

don't state how big data types are. They say things

like: "A short may not be longer than a long, nor a long

shorter than a short." All we really can be sure of is

that a char is an 8 bit byte.

They may all be the same size, all different sizes, some

the same and some not, and so on.

So: write a program which prints the sizes of all of

C++'s Basic ("primitive") data types. Don't ask anybody

else what they are, find out on your own.

You must use the sizeof() operator to get the size of a type.

for example:

cout << "Integer: " << sizeof(int) << endl ;

As a professional programmer, you're expected to produce neat and

businesslike output such as this:

type 01

type 02

type 04

type 04

short type 08

type 04

type 08

long type 12

void \* (ptr to something) 04

...that is to say, not only must you have your ducks

lined up, but your digits as well. Numeric data MUST

be arranged in neat, vertically-aligned tabular format,

or people will laugh at you and refuse to sign your

paycheck.

DISCUSSION:

There are two systems to send output to the screen,

Standard I/O and Iostream. (Iostream doesn't work under C,

only only under C++, Standard I/O works with both.)

stdio: printf("char %02d\n", sizeof(char)) ;

iostream: cout << "char " << setw(2) << sizeof(char) << endl ;

We're starting with Stdio, because it's the world standard for

output, and exists in Java, awk, PERL, and a host of others.

In crafting your program, refer to the program structure

in the text and Syllabus, copy as much as you can, and start

memorizing what you're copying.

PART II

After the sizes, print the ASCII Character set where

possible: print the (if printable) ASCII character and

the number of each character, expressed both as Base-8,

Base-10 and Base-16. (Octal, Decimal, and Hexidecimal.)

Using a for() loop like this:

for (int i = 32 ; i <= 127 ; i++)

{

printf("%c %o %d %x \n", something, something, something, something ) ;

}

What should replace "something"? Think about it.

(Note that you should use constant widths for the

numbers so that the units, tens, etc. places align

vertically.) For cout, this requires setw() and #include

<iomanip>. Alternatively, you can do the same thing

with fewer keystrokes using printf() #include <cstdio>.

Learn both.

If you are planning on doing well in the course, then print

the whole range from 0 to 127. Print the printables,

and alias the non-printable chars with '.'. How do you

find out?

isprint(c) ? c : '.' ;

Stop until you understand that line. (Look up: isprint and

Conditional Operator, read and understand, then use them.

This tests your ability to read at the high school level,

and use answers you find.)

Sample output:

The ASCII Characters

. 000

. 001

. 002

. 003

. 004

. 005

. 006

. 007

. 008

. 009

. 010

. 011

. 012

. 013

. 014

. 015

. 016

. 017

. 018

. 019

. 020

. 021

. 022

. 023

. 024

. 025

. 026

. 027

. 028

. 029

. 030

. 031

032

! 033

" 034

# 035

$ 036

% 037

& 038

' 039

( 040

) 041

\* 042

+ 043

, 044

- 045

. 046

/ 047

0 048

1 049

2 050

3 051

4 052

5 053

6 054

7 055

8 056

9 057

: 058

; 059

< 060

= 061

> 062

? 063

@ 064

A 065

B 066

C 067

D 068

E 069

F 070

G 071

H 072

I 073

J 074

K 075

L 076

M 077

N 078

O 079

P 080

Q 081

R 082

S 083

T 084

U 085

V 086

W 087

X 088

Y 089

Z 090

[ 091

\ 092

] 093

^ 094

\_ 095

` 096

a 097

b 098

c 099

d 100

e 101

f 102

g 103

h 104

i 105

j 106

k 107

l 108

m 109

n 110

o 111

p 112

q 113

r 114

s 115

t 116

u 117

v 118

w 119

x 120

y 121

z 122

{ 123

| 124

} 125

~ 126

-----------------------------

Handing it in:

TRANSFERRING FILES TO AND FROM A SERVER USING FTP, (PSFTP.EXE)

OR SCP (scp).

You will use one or another FTP (file transfer protocol)

program intensively your whole working life.

If you are using a Windows PC:

When you are ready to upload the program file to the Mesa

system, you can use the "psftp.exe" program on the class

web page to get work from the local system to Mesa:

Put the USB drive in the USB port,

From Firefox, navigate to the class

web page at http://209.129.16.61/~hhaller/data/cisc192

double-click psftp.exe

a command-window will pop up, type:

open 209.129.16.61

when prompted "login:" enter your assigned login name, when prompted

"passwd:" enter your password

if you're uploading assignment 0, you would change directory (cd) to

the asst1 subdir so:

cd asst0

then, type

put e:\yourfilename.cpp

(e: may be wrong, you must know which disk letter your thumbdrive or

floppy is, what subdirectory holds your file, and use that.)

wait until the transfer is complete, then type: quit

(To reverse the process and copy the file to your computer, after you

create an ftp connection as described above, type

lcd e:\

(that's Local Change Directory)

get yourfilename.cpp

and your file will be downloaded from the mainframe and

saved on your A: drive.

If you're on a system with a Unix command line, you may

copy files to or from your directory on the distance system:

scp myacct@distant.com:filewanted .

or:

scp ./filetosend myacct@distant.com: