

COP 4530 :: Recitation#1 Fall 2011 (2nd September, 2011)

Objective:

1. Assignment#1 discussion
2. Program Compilation
3. Using makefiles

1. Assignment#1 Discussion:

a) **String operation (basic):** Can you fill up the dotted sections ?

```
#####  
#include<iostream>  
#include<string>  
  
using namespace std;  
  
int main()  
{  
  
string mystring("data structures ");  
  
// write one or two lines of code to append "fun" to mystring  
.....  
  
//write one or two lines of code to replace "structure" with "model" in mystring  
.....  
  
//write one or two lines of code to clear the data from mystring without assigning  
it to something else  
.....  
  
}  
#####
```

(Hint: Use replace, append and clear member functions of the string class)

b)**String operation (Continued):** *find_first_of* and *find_first_not_of* are two constant member functions of the string class. They can be used to search the occurrence of characters in the string instance. Here is a basic prototype:

```
//Returns position of first occurrence of character 'ch' starting from 'pos' in the string object  
size_t find_first_of ( char ch, size_t pos = 0 ) const;
```

```
//Returns position of first occurrence of a character which is not 'ch' starting from 'pos' in the string  
object
```

```
size_t find_first_not_of ( char c, size_t pos = 0 ) const;
```

Also, *size_t* is a typedef that can be converted to an integer. In case of an unsuccessful search the static constant *npos* (you can use in your code as `string::npos`) is returned

Try to fill out the dotted lines in the following code snippet.

```
#####  
#include<iostream>  
#include<string>  
  
using namespace std;  
  
int main(){  
  
    String sentence("A new day has come finally");  
    char ch = 'c';  
  
    size_t pos1;  
    size_t pos2;  
  
    //Write a single line of code to get the first occurrence of character 'c'  
    pos1 = .....  
  
    //Write a single line of code to get the position of character 'a' after  
pos1    pos2 = .....  
  
    //Write an if statement to check if the searches were successful or not  
    .....  
  
}
```

The first argument of both these functions can be a character array or a string itself. These functions facilitates searching any character from a group of characters.

c) **File Operations:** *tellg* is a function that returns the absolute position of the get pointer for the ifstream. Hence if the get pointer is at present at the end of the file, tellg will return the size (or the number characters) of the file. It is a member function of ifstream and does not take any arguments

seekg is used to set the position of the get pointer. In is again a member function of ifstream and commonly takes the position as the argument (for other variances see references).

Please fill in the dotted sections:

```
#####  
#include<iostream>  
#include<fstream>  
#include<string>  
  
using namespace std;  
  
int main(){  
  
    const *char const filename("myfile.txt");
```

```

    ifstream::pos_type size;

    //Write a line of code to open it for reading and set the start pointer to
the end
    ifstream myfile(filename, .....);

    //Write a line of code to get the size of the file
    size = .....

    //Write a line of code to set the pointer to the beginning of the file
    .....
}
#####

```

d) **Namespaces:** What problem might occur in the following code snippet ? How to resolve it ?

```

#####
#include<iostream>

int main(){
    cout << "Hello World" << endl;
}
#####

```

2. Compilation:

a) What are the meaning of flags -Wall and -pedantic in g++ command ?

3. Make:

Make is a very useful tool for program compilation when there are dependencies. In the most basic case the syntax of a make file is the following:

```

<Target>: <dependencies>
<Tab > <command>

```

Here is a complete example of a make file with macros

```

#####
HOME = /home/courses/cop4530/fall05/recitation
CC = g++ -Wall -pedantic
PROJ = $(HOME)/rect2/makeutil
INCL = -I $(PROJ)

all: main.x

main.x: largest.o print.o main.o
< TAB >$(CC) -o main.x print.o largest.o main.o

largest.o: $(PROJ)/largest.h $(PROJ)/largest.cpp
< TAB >$(CC) -c $(INCL) $(PROJ)/largest.cpp

```

```
print.o: $(PROJ)/print.h $(PROJ)/print.cpp
< TAB >$(CC) -c $(INCL) $(PROJ)/print.cpp
```

```
main.o: $(PROJ)/main.cpp
< TAB >$(CC) -c $(INCL) $(PROJ)/main.cpp
```

```
clean:
```

```
< TAB >rm -rf *.o *~ *.x
```

```
#####
```

what will happen if the name of the file is changed to “abc.txt” ? How to resolve that problem ?

References:

1. tellg : <http://www.cplusplus.com/reference/iostream/istream/tellg/>
2. seekg: <http://www.cplusplus.com/reference/iostream/istream/seekg/>
3. ifstream: <http://www.cplusplus.com/doc/tutorial/files/>
4. string : <http://www.cplusplus.com/reference/string/string/>
5. make : http://developers.sun.com/solaris/articles/make_utility.html
6. g++ warings: <http://gcc.gnu.org/onlinedocs/gcc-3.4.6/gcc/Warning-Options.html>