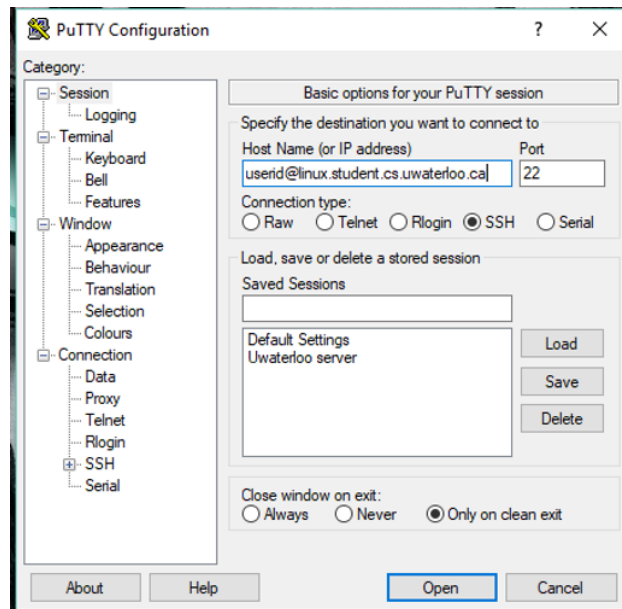


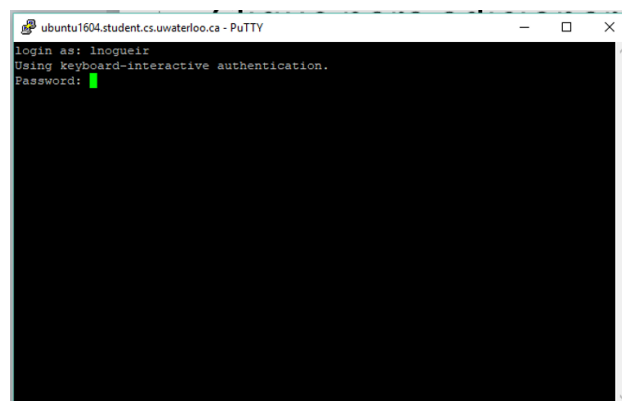
Tutorial 1

1 student environment set up

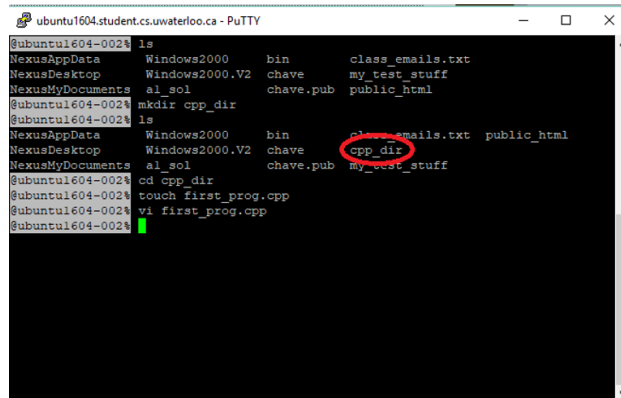
1. First download Putty



2. Login with the information from your CS account



3. create a new directory to store your files, strat editing your file using vim

A screenshot of a terminal window titled 'ubuntu1604.student.cs.uwaterloo.ca - PuTTY'. The terminal shows a series of commands and their outputs. The first command is 'ls', which lists the contents of the current directory: 'NexusAppData', 'NexusDesktop', 'NexusMyDocuments', 'bin', 'class_emails.txt', 'chave', 'my_test_stuff', 'al_sol', and 'public_html'. The second command is 'mkdir cpp_dir', which creates a new directory named 'cpp_dir'. The third command is 'ls', which lists the contents of the current directory again, showing 'cpp_dir' has been added. The fourth command is 'cd cpp_dir', which changes the current directory to 'cpp_dir'. The fifth command is 'touch first_prog.cpp', which creates a new file named 'first_prog.cpp'. The sixth command is 'vi first_prog.cpp', which opens the file in the vi editor. The terminal shows the prompt '\$' at the end of each command line.

```
ubuntu1604.student.cs.uwaterloo.ca - PuTTY
$ ls
NexusAppData  Windows2000  bin          class_emails.txt
NexusDesktop  Windows2000.V2  chive       my_test_stuff
NexusMyDocuments  al_sol        chive.pub   public_html
$ mkdir cpp_dir
$ ls
NexusAppData  Windows2000  bin          class_emails.txt  public_html
NexusDesktop  Windows2000.V2  chive       my_test_stuff
NexusMyDocuments  al_sol        chive.pub   my_test_stuff
$ cd cpp_dir
$ touch first_prog.cpp
$ vi first_prog.cpp
$
```

2 Compiling your programs

command: `g++ -std=c++14 -wall XXX.cc -o YYY`

`-std=c++14`: compile using version c++14

`-wall`: display warning

`-o`: output file(default as a.out)

3 set of problems

1. What is the output of this function?

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

int main() {
    string str;
    vector<string> v;
    while(!cin.eof()) {
        getline(cin, str);
        v.push_back(str);
        if(str == "") {
            cout << "empty" << endl;
        }
    }
    cout << v.size() << endl;
}
```

(a) Input: abcd\n\nEOF
 Output:
 empty
 empty
 3

(b) Input: abcd\nEOF
 Output:
 empty
 2

What about

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

int main() {
    string str;
    vector<string> v;
    while(getline(cin, str)) {
        v.push_back(str);
        if(str == "") {
            cout << "empty" << endl;
        }
    }
    cout << v.size() << endl;
}
```

(a) Input: abcd\n\nEOF
 Output:
 empty
 2

(b) Input: abcd\nEOF
 Output:
 1

2. Create a function that reads a text file and store the values of each line in a vector.

3. Create a function that gets the entire of a vector and stores them in a text file.
4. Create a function that given a text file with names, creates another text file with the names sorted in alphabetical order.

```
#include <string>
#include <vector>
#include <fstream>
#include <iostream>
#include <algorithm>
using namespace std;

void storeVals(vector<string> &vals, string &file) {
    ifstream input;
    string str;
    input.open(file);
    if(!input.good()) {
        return;
    }
    while(input >> str) {
        vals.push_back(str);
    }
    input.close();
}

void OutputVals(vector<string> & vals, string &file) {
    ofstream output;
    if(file.find(".txt") == string::npos) {
        cerr << "File should be .txt" << endl;
        exit(1);
    }
    output.open(file);
    int size = vals.size();
    for(int i = 0; i < size; ++i) {
        output << vals[i] << endl;
    }
    output.close();
}

int findMin(vector<string> &vals, int begin) {
    int min = begin, size = vals.size();
    for(int i = begin + 1; i < size; ++i) {
        if(vals[min] > vals[i]) {
```

```

        min = i;
    }
}
return min;
}

void swap(vector<string> &vals, int i, int j) {
    string temp = vals[i];
    vals[i] = vals[j];
    vals[j] = temp;
}

void sortNames(vector<string> &vals) {
    //sort(vals.begin(), vals.end());
    int size = vals.size();
    for(int i = 0; i < size; ++i) {
        int min = findMin(vals, i);
        swap(vals, i, min);
    }
}

int main() {
    string nameIn ,nameOut;
    cin >> nameIn >> nameOut;
    vector<string> names;
    storeVals(names, nameIn);
    sortNames(names);
    OutputVals(names, nameOut);
    return 0;
}

```