

VE280 Mid Review

Linux

- cd, ls, mkdir, rmdir, touch, cp, mv, rm
- nano/gedit/vim/emacs, cat, less, diff
- >, <
- sudo, apt-get install/autoremove, man

Exercises:

- Append the content of `a.txt` to `b.txt`
- Rename `p1.cpp` to be `p2.cpp`
- Use c++11 standard to compile the program `p1` with `p1.cpp`

Developing Programs

- Write code -> Compile -> Run
- Header Guard
- Makefile

```
all: run_add

run_add: run_add.o add.o
    g++ -o run_add run_add.o add.o

run_add.o: run_add.cpp
    g++ -c run_add.cpp

add.o: add.cpp
    g++ -c add.cpp

clean:
    rm -f run_add *.o
```

C++ Basics

- x++ & ++x
- Call-by-Value & Call-by-Reference
- References & Pointers

```
int main1() {
    int x = 3; // the first x in main() scope
    if (x == 3) {
        int x = 5; // the second x in if scope
        cout << x; // use the x in the same scope (x ==
5)
    }
    cout << --x; // use the first x (same scope),
decrement before print it out
    return 0;
}
```

```
int x = 4;
void f(int &x) { // this x is reference
    x++; // changes the value of argument
}
int main2() {
    f(x); // main2 can only use global x (4)
    cout << x; // global x has been changed by f(x)
    int x = 3; // second x in main2() scope
    f(x); // use the second x (same scope)
    cout << x; // local x has been changed by f(x)
    return 0;
}
```

```
int x = 4;
void g(int *x) { // the address is copied to argument x
    x++; // increment the address, nothing happens to
caller
}
int main3() {
    g(&x); // send the address of global x
    cout << x; // nothing happens to global x
    int x = 3;
    g(&x); // send the address of local x
    cout << x; // nothing happens to local x
    return 0;
}
```

```
#include <iostream>
```

```

using namespace std;

struct MyStruct { // note typedef is not needed */
    int length;
    int values[3];
};

void change(MyStruct t) {
    t.length++;
    t.values[1] = 3;
}

int main() {
    MyStruct t = {3, {0, 1, 2}};
    change(t);
    cout << t.length << " " << t.values[1] << endl;
}

```

Const Qualifier

- const int MAXSIZE = 256;
- const Reference
- const Pointers

Procedural Abstraction

- REQUIRES
- MODIFIES
- EFFECTS