

CAAM420: Daily Notes

Chang Tian

10/18/13

1 Daily announcement

Chapter four note is now online.

Project 3 is due 25th Oct.

Use piazza for any questions

Vigrind will be very useful to determine any memory-related errors

Today's main content: C++: a better version of C

2 Input and output features

Compare the usual i/o commands of C and C++:

C VS C++

Stdout VS cout

Stderr VS cerr

Stdin VS cin

C print statement: `Printf("n=%d\n",n);`

C++ print statement:

`cout << "n=" << n << "\n";`

We do not have to tell the program how to convert the integer

cout can detect the type and do appropriate conversion

We can also modify the program to decide how many digits to print out

More examples will be shown later

3 Operator overloading

The operator overloading is a nice feature of C++ as you can give special meanings to operators when they are used with user-defined types

It will redefine an assignment operator for a new type of object, for example, we can put two streams together using a plus

We will encounter more examples in the future.

4 Default parameters

Some parameters of a function are set as default values.

Therefore, we do not need to type in the value again unless we want to use other values for the parameter.

5 Name mangling

We will use `ex42_vector.c` as a good example for the remaining of this lecture.

In `c`, we cannot have two functions with the same name, so the linker assumes a unique name for each function

in `c++`, however, we can have two functions with the same name as long as they have different parameter types. `C++` combines the name and the types of parameters to let the linkers see the functions with different names.

Therefore, we need use some device to tell the linker which is `c` and which is `c++` function

6 Source file

In `C`, the source file can only end with `.c` or `.h`, which can only include declaration.

In `C++`, however, the file can end with `.C`, `.H`, `.cc`, `.hh`, `.cpp`, `.hpp`, `.cxx` or `.hxx`

Example: `ex42_vectorpp.cc`

7 struct VS class

Instead of using “struct”, `C++` uses “class” for type building.

8 Private, protected and public

In `C++`, we have these three access types:

private: only accessible directly by instances (“objects”) of the class, values not visible outside the file, not accessible for other functions.

protected: only accessible directly by instances of the class and its subclasses.

public: accessible by other class or function.

9 Default constructor and copy constructor

Any class allocating memory dynamically must be provided with its proper copy constructor and destructor to avoid duplicated reference to the same memory.

Post construction: if you want to modify member data after construction, you must declare the data mutable.

For example, when we copy a vector, it does not share data with the previous vector. The new vector will simply have the same value.

We do not want the pointer to be copied. We want the data to be copied because if we copy the pointer, the two variables will share the same data and consequently the same allocated memory. Then when we want

to free the second vector after freeing the original vector, the compiler will give an error that the memory has been freed already.

More details about this section can be seen on L59 from ex42_vectorpp.cc.

10 C++ version of malloc and free

Instead of malloc and free in C, C++ uses new and delete.

With reference from L125 of ex42_vectorpp.cc, note that the destructor erases the data from the stack. Therefore we don't need to care about the state, it does not exist anyway. The destructor is automatically called when the vector goes out of scope.

Delete will to get rid of the dynamic memory the data allocated to.

11 Declare the variable at the point of use

In C++, the programmer can declare the loop variable inside the for statement.

If we declare an integer `i` in the for statement, then the scope of the variable `i` is only in the for statement.

We can declare the variable at the point of use so that we do not need to page to the top to see the type of a variable.

We will finish talking about the example on Monday.