OOP1: Basic C++

Created	@May 21, 2024 9:51 AM
Type	Lecture
Reviewed	☐
Projects	Professor Interaction, Object Oriented Programming
Resources	Object Oriented Programming

@May 21, 2024 1:40 PM

Contents:

1. Class Ideas → Not really important so whatever
2. C++
    a. Basic Info
        i. Who C++
        ii. Why
    b. Hello World and related
    c. Type
    d. Auto
    e. some useful commands
    f. if-else
    g. switch
    h. Loop
    i. Strings
    j. Some vector ideas

C++

Basic Information:

Creator: Bjarne Stroustrup（不会真的考吧...）

Why C++:

- Manual
- Usually faster and smaller

C++ VS Python:

C++

- Compile(Check Error and turn it into an object file) first, then interpret(run object)

Python

- Compile and interpret together, one line after another

Hello World

- #include <iostream>
  - You can see it as import a class(std) into the file so that you can do certain things like
    - cin
    - cout
    - ...

    - if you have things like GUI, you may not need it
- Write in main
    - you still need to specify the return type
    - usually return 0, so that you know your file can be compiled if there is a 0 at the end
    - but you don't really necessarily do so as it will still give you an exit code 0
- std::cout <<
    - meaning: character output
    - pronounced "c-out"
    - a part in the std(standard) package
    - The similar function to "print"
    - << here mean put "..." into the console named cout
        - GPT: This operator takes the data on its right side and inserts it into the stream on its left side.
    - It will not change to the next line itself, so you need a \n
    - If you have multiple things to input in a line, you can use multiple <<
        - std::cout << "Hello" << name
- endl
    - meaning endline
    - will switch to the next line
- std::
    - a direction lead you to the standard "package"
    - you can use "using namespace std; " to omit all the std::

Don't Forget to Add ';' at the end of each sentence!

Type

In C++, a variable can only hold 1 type, and you have to initialize it first

Type name	Example	Storage Requirement
bool	true/false(small case)	
char	‘a’,’1’	1 byte = 8 bit
int	integers	4 byte
double	3.14	4+4 = 8 byte
…		

E.g. Int a = 9

00001001 ⟶ 1001 = bin(9)
00000000
8 0s
8 0s

Therefore, Max Int = 2^31 -1 (altogether 4 ∗ 8 = 32 bits but 1 is for the sign)

Auto

If you don’t know what type to use, you can initialize the variable as “auto”

or

you can cast it with decltype

E.g.

```
int a = 0;
auto b:int = a;
//decltype(a) b;//declare the type of b with the same type of a
```

Some useful commands

- Const: set a var constant and cannot be modified
- \b: backslash
- \\: \
- \t: tab
- ++x & x++
- +=...
-

```
const bool status = true;
//status = false u cannot modify as it is const
```

```
cout << "------------------------------" << endl;
cout << "Hello\b\b\b\\";
cout << "Wor\tld";
```

If-else statement

```
if(c > 5){
std::cout << c <<std::endl}
else if(c == 5){
std::cout << c << std::endl}
else{
}
```

still use “==” here

Switch

```
switch(statement:thing consider){
case 1(check if statement = case):
case 2:
...
default:
}

Once a cased is met, all the underlying case will be outputed
will always run default case even if there is some case being me
```

Loop:

For:

```
for(initial; control(not end step); step){
}
```

While:

```
while(control){}
```

do...while():

```
do{}while();
remind the ';'
```

Function

Declaration:

at the beginning

```
returntype func-name(type1 name1,type2 name2);
remind the ';'
```

Define: at the end

```
returntype func-name(type1 name1,type2 name2){}
remind the '{}'
```

We can also use auto here

However, use auto for a function means you ask the compiler to deduce a type, so it must have a type but not null

Strings:

- mutable-unlike python
- see it like a list

File I/O

1. [type]stream name
2. .open()
3. getline(from, named as)

4. .close() for garbage collection

Vector

Problems Left to be solved

auto function? when will go wrong → see example in the file