Question 1: Infinite loop

```
int a;
std:: cin >> a;
```

如果输入 a 是字符或者 double，无法读取正确的输入；返回值是 0 或者取舍整数，最严重会出现死循环现象！

Hint: 明白 cin 的返回值和几个状态!

Question 2: Different headers between .h and .hpp?

Question 3: A byte has 8 bits?

1) A single character (such as letter or other keyboard symbols) is 8 bits
2) 8 is a power of 2; The machine code is binary.

Question 4: High-level language, low-level language and machine language?

Question 5: compiler and linker?

Hint: before compiler is source program/code; after compiler is object program/code

Question 6: OOP has three main characteristics?

1) Encapsulation: information hiding or abstraction
2) Inheritance: reuse program codes
3) Polymorphism: A single name can have multiple meanings in the context of inheritance

Question 7: C and C++ language

Question8: Three main errors?

1) Syntax
2) Run-time
3) Logic

Question 9: Understand using namespace std;?

Hints: scope,

Question 10: How to display decimal point in C++?

Hints: #include <iomainp>

```
cout.setf(ios::fixed);   //setf means set flags
cout.setf(ios::showpoint);
cout.precision(2);
cout.set(ios::showpos); //show + signs eg. +5.0 or -3.4
cout.width(5); //limit the space
```

Question 11: C++11 new features?

1) auto x = expression;

Question 12: String and whitespace

Hint: cin reads until it encounters a whitespace character

Question 13: Type Compatibilities or type mismatch (int vs double), (int vs char), (char vs string), and bool;

Question 14: Type cast and Type conversion?

Question 15: Overloading in c++? Hints: different types and numbers

Question 16: return for void function!

Question 17: Call by value and Call by reference?

Hints: & is used for the other name. Understand std::swap function!

Question 18: Formal Parameters and Arguments

Question 19: Class ifstream and Class ofstream

Hints: dot operator: a.open();

Hints: ofstream outFile; outFile.open(“a.txt”, ios::app); ios::app appends the content to the end or create an empty file.

Hints: stream parameters must be call by reference

Question 20: The exit statement?

Hints: exit(1); #include <cstdlib> and using namespace std;

Question 21: What does compile check in makefile?

Hints: variables, functions, and grammars.

Question 22: Difference between compiler error and linker error?

Question 23: Character I/O: Member function get, put, and putback

Hints:

1) Input_Stream.get(Char_Variable);
2) Char 的数据类型
3) Accept blank, newline, and one char!

Compare cin:

1) Cin rejects blank, and newline!

Question 24: default arguments

Question 25: Array: C++11: range-based for loop

Question 26: Array argument (like call by reference) Book p.395;

Answer: an array formal parameter is neither a call-by-value parameter nor call-by-reference parameter, it is likely to call by reference.

Question 27: Sort an array

Hint:

1) Selection Sort: Place the indexth smallest element
2) Bubble Sort
3) Insertion Sort

Question 28: String is an array with base type char; '\0' is called the null character and is used as an end marker. String has '\0'!

Question 29: cin, get, getline machine: (p.338)

标准输入缓冲区，换行符（回车键）依然会存入缓冲区

cin 分隔符（输入的结束）：tab, space, and return ; 如果使用输入 cin，最好清除缓冲区；

cin 可以截取前面正确的数字，不正确的数字继续留在缓冲区

getline() 属于 string 流 不会忽略换行符，换行符会转换'\0',代表结束; #include<string>

cin.get ()不会糊流分隔符; #include <string>

cin.getline() 属于 istream 流，读取一串字符串，以指定的结束符结束！有 3 个参数

cin.ignore() 清空输入缓冲区的当前行，使上次残留下的数据没有影响下次输入，numeric_limits::max() 使用的最大值，可以自己用足够大的整数代替

cin.ignore(): 当输入缓冲区没有数据时，会阻塞等待数据到来

default arguments for function

Question 30: String class: member function (p.483)

1) Str.substr(position,length)
2) Str.find(str1); str.find(str1,pos)
3) Str.insert(pos,str2); str.erase(pos,length);

Question 31: Size and Capacity?

Question 32: C-string value and C-string variables

Question 33: * and & operator (p.511)

Question 34: new operator (dynamic variables)

The new operator creates a new dynamic variable of a specified type and returns a pointer that points to this new variable.

Question 34: Basic memory management:

Freestore = heap

Question 35: dangling pointers (undefined pointer variables) means: *p the result is unpredictable and usually.

Question36: Why do we use dynamic array? (p.524)

Question 37: Structures and classes? (p.575)

Question 38: Private; Public; Protected?

Question 39: virtual function is late binding. (Polymorphism, override, redefine). p,869

Question 40: Why should all destructors be virtual?

Question 41: Throw (Try-Catch). Throw a value of any type?

Catch(type e)…. It is like a function, but not.

Multiple throws and catches

Question 42: typedef keyword?

Question 43: #ifndef, #define, #endif mechanism to prevent multiple file inclusion of all the files .

Question 44: friend function refers to outside the class; Friends can access private members! but Friends are not member functions! In other words, a friend function has the same access privileges as a member function.

Hint: don't use qualifier ::

Question 45: Exception handle try{throw1;thow2;…}catch(throw1){}catch(throw2){}…

Question 46: Throw an Exception inside a function

Question 47: Template in compiler

Question 48: STL iterator refers to generalization of a pointer.

Question 49: STL? String, vector, list, stack, queue, set, map

Hints: set and map have order

Question 50: Look at keywords or predefined:

Hint: aligns, alignof, as, constexpr, decltype, mutable, dynamic_cast, noexcept, register, reinterpret_cast, static_assert, thread_local, typeid, volatile, wchar_t

Question 51: Precedence of Operators:

Hints: x = y = z means x = (y = z); x+ y+ z means (x+y) +z

Question 52: Libaraies: Character function ~ int function

Hints: <cstring> <cstdlib> <cctype>

Eg. 1) int atoi(const char a[]); //string -> int

2) int rand();

3) void srand(unsigned int);

Question 53: inline function in class

Question 54: oveload the square brackts

Eg. char& operator[](int index);

Question 55: this pointer points to the calling object

Question 56: 返回值是指针，什么时候需要 delete，什么时候不能 delete？

Question 57: delete 代表释放内存空间，那片区域暂时保存了以前的值；

Makefile notes:

https://blog.csdn.net/haoel/article/details/2886/ 中文参考目录 1
https://blog.csdn.net/haoel/article/details/2887 zho2
https://blog.csdn.net/haoel/article/details/2888 zong 3

Makefile simple rules:

Target:Prerequisites

<Tab>Command …. #! Tab is important

1) Target is an object file(.o or .obj), or executable file(.exe), or a lable.
2) Prerequisites: Dependent files (.h or .cpp..)

If prerequisites are updated, target will be updated.

Makefile has:

1) Explicit rules
2) Implicit rules
3) Define variables
4) Comment: #
5) Include file

Makefile Names:

1) "GNUmakefile"-> "makefile"->"Makefile" # find in order
2) New name: make -f <newname> eg. make -f makefile.arr

Makefile reference:

1) Include <filename>eg. include foo.make *.mk $(bar) # many makefiles and variables, separate by space

2) “-I” or “--include-dir” find different directory.

Makefile has only one final target that must be in the top.

Makefile symbles:

1) “~” eg. ~/test means $Home directory
2) “*” eg. *.c all .c files

Return pointer:

1)引用总是指向一个对象,没有所谓的 null reference .所有当有可能指向一个对象也由可能不指向对象则必须使用指针.由于没有所谓的 null reference 所以所以在使用前不需要进行测试其是否有值.,而使用指针则需要测试其的有效性.