Student Name: Gao Jinliang
Student ID: 2022141450043

C++ Programming
Assignment 1

Section 1: Configure the corresponding docker environment

Part1 The screenshot of docker run hello-world,docker images,docker ps running in terminal.(Verdict as following)

```
PS C:\Users\GAO> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

```
PS C:\Users\GAO> docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
scucpphw_test_img   latest    f16f8f5276f8   3 days ago       1.35GB
<none>              <none>    1f28488c05b0   4 days ago       1.3GB
hello-world         latest    feb5d9fea6a5   17 months ago    13.3kB
```

```
PS C:\Users\GAO> docker ps
CONTAINER ID   IMAGE                      COMMAND       CREATED          STATUS          PORTS     NAMES
fcf0e3c9bc0f   scucpphw_test_img:latest   "/bin/bash"   13 seconds ago   Up 12 seconds   22/tcp    gracious_clarke
```

Part2 The screenshot of cd /ws/code/ after entering docker with docker exec -it ¡container id¿ /bin/bash.(Verdict as following)

```
root@fcf0e3c9bc0f:/# cd /ws
root@fcf0e3c9bc0f:/ws# g++ test.cpp
```

Section 2: Compiling single files with g++

Part1 Compile the test.cpp file in single step and step by step, print out the corresponding process file with the ls command and take a screenshot, execute the a.out executable obtained in single step, and the test executable obtained in step by step, and give the corresponding running results.(Verdict as following)

```
root@fcf0e3c9bc0f:/# cd /ws
root@fcf0e3c9bc0f:/ws# g++ test.cpp
root@fcf0e3c9bc0f:/ws# g++ -E test.cpp -o test.i
root@fcf0e3c9bc0f:/ws# g++ -S test.i -o test.s
root@fcf0e3c9bc0f:/ws# g++ -c test.s -o test.o
root@fcf0e3c9bc0f:/ws# g++ test.o -o test
root@fcf0e3c9bc0f:/ws# g++ -std=c++17 test.cpp
root@fcf0e3c9bc0f:/ws# g++ -Wall test.cpp
root@fcf0e3c9bc0f:/ws# g++ -g test.cpp -o test
root@fcf0e3c9bc0f:/ws# readelf -S test | grep -i debug
  [27] .debug_aranges    PROGBITS         0000000000000000  00006146
  [28] .debug_info       PROGBITS         0000000000000000  000065a6
  [29] .debug_abbrev     PROGBITS         0000000000000000  0000f6a9
  [30] .debug_line       PROGBITS         0000000000000000  00010235
  [31] .debug_str        PROGBITS         0000000000000000  00010f2b
  [32] .debug_rnglists   PROGBITS         0000000000000000  0001afa6
  [33] .debug_line_str   PROGBITS         0000000000000000  0001b25f
root@fcf0e3c9bc0f:/ws# g++ -O2 test.cpp
g++: error: unrecognized command-line option '-O2'
root@fcf0e3c9bc0f:/ws# g++ -O2 test.cpp
```

The picture above here is how these files are made without "ls", and the picture below is that with command "ls".

```
root@fcf0e3c9bc0f:/ws# g++ test.cpp
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test.cpp  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# g++ -E test.cpp -o test.i
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test.cpp  test.i  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# g++ -S test.i -o test.s
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test.cpp  test.i  test.s  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# g++ -c test.s -o test.o
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test.cpp  test.i  test.o  test.s  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# g++ test.o -o test
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test  test.cpp  test.i  test.o  test.s  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# ./a.out
vector v after call to generate_n() with lambda: 1 1 2 3 5 8 13 21 34
x: 1 y: 1
vector v after 1st call to fillVector(): 1 2 3 4 5 6 7 8 9
vector v after 2nd call to fillVector(): 10 11 12 13 14 15 16 17 18
root@fcf0e3c9bc0f:/ws# readelf -S test | grep -i debug
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test  test.cpp  test.i  test.o  test.s  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws# g++ -O2 test.cpp
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test  test.cpp  test.i  test.o  test.s  with_o.o  without_o.o
root@fcf0e3c9bc0f:/ws#
```

Part2 Gives information about the execution of the optimized and non-optimized compilation of inefficency.cpp using the time command.(Verdict as following)

```
root@fcf0e3c9bc0f:/ws# ls
1.cpp  a.out  inefficency.cpp  test  test.cpp  test.i  test.o  test.s
root@fcf0e3c9bc0f:/ws# g++ inefficency.cpp -O2 -o with_o.o
root@fcf0e3c9bc0f:/ws# g++ inefficency.cpp -o without_o.o
root@fcf0e3c9bc0f:/ws# time ./with_o.o
result = 100904034

real    0m0.014s
user    0m0.006s
sys     0m0.000s
root@fcf0e3c9bc0f:/ws# time ./without_o.o
result = 100904034

real    0m1.873s
user    0m1.864s
sys     0m0.000s
root@fcf0e3c9bc0f:/ws#
```

Part3 Select 3 to 4 other test files and try to compile them with different g++ parameters, giving screenshots of the corresponding generated files and the final execution.(Verdict as following)

File1 test_class.cpp(C++11 standard)

```
root@fcf0e3c9bc0f:/ws# g++ test_class.cpp
root@fcf0e3c9bc0f:/ws# ls
1.cpp           test      test.o          test_class_size.cpp         test_noexcept.cpp  with_o.o
a.out           test.cpp  test.s          test_default_parameter.cpp  test_ptr.cpp       without_o.o
inefficency.cpp  test.i    test_class.cpp  test_move.cpp               test_raii.cpp
root@fcf0e3c9bc0f:/ws# g++ -E test_class.cpp -o test_class.i
root@fcf0e3c9bc0f:/ws# g++ -S test_class.i -o test_class.s
root@fcf0e3c9bc0f:/ws# g++ -c test_class.o -o test_class
g++: warning: test_class.o: linker input file unused because linking not done
g++: error: test_class.o: linker input file not found: No such file or directory
root@fcf0e3c9bc0f:/ws# g++ -c test_class.s -o test_class.o
root@fcf0e3c9bc0f:/ws# g++ test_class.o -o test_class
root@fcf0e3c9bc0f:/ws# ls
1.cpp           test      test.o      test_class.cpp  test_class.s               test_move.cpp      test_raii.cpp
a.out           test.cpp  test.s      test_class.i    test_class_size.cpp        test_noexcept.cpp  with_o.o
inefficency.cpp  test.i    test_class  test_class.o    test_default_parameter.cpp  test_ptr.cpp       without_o.o
root@fcf0e3c9bc0f:/ws# g++ -std=c++11 test_class.cpp
root@fcf0e3c9bc0f:/ws# g++ -Wall test_class.cpp
root@fcf0e3c9bc0f:/ws# g++ -g test_class.cpp -o test_class
root@fcf0e3c9bc0f:/ws# readelf -S test_class | grep -i debug
  [26] .debug_aranges    PROGBITS         0000000000000000  00003082
  [27] .debug_info       PROGBITS         0000000000000000  000030d2
  [28] .debug_abbrev     PROGBITS         0000000000000000  0000596d
  [29] .debug_line       PROGBITS         0000000000000000  00005fc2
  [30] .debug_str        PROGBITS         0000000000000000  0000623b
  [31] .debug_rnglists   PROGBITS         0000000000000000  000075f2
  [32] .debug_line_str   PROGBITS         0000000000000000  0000761e
root@fcf0e3c9bc0f:/ws# g++ -O test_class.cpp
root@fcf0e3c9bc0f:/ws# ./a.out
1       #21325302 is created
1       #58320212 is created
5       #21325302       5000    5000
25      #58320212       10000   10000
45      #21325302       5500    10500
60      #58320212       -4000   6000
90      #21325302       27.64   10527.6
90      #58320212       21.78   6021.78
#21325302       Balance: 10527.6
#58320212       Balance: 6021.78
root@fcf0e3c9bc0f:/ws# ls
1.cpp  inefficency.cpp  test.cpp  test.o  test_class      test_class.i  test_class.s         test_default_parameter.cpp  test_noexcept.cpp  test_raii.cpp  without_o.o
a.out  test             test.i    test.s  test_class.cpp  test_class.o  test_class_size.cpp  test_move.cpp               test_ptr.cpp       with_o.o
```

File2 test_move.cpp(C++98 standard)

```
root@fcf0e3c9bc0f:/ws# g++ test_move.cpp
root@fcf0e3c9bc0f:/ws# g++ -E test_move.cpp -o test_move.i
root@fcf0e3c9bc0f:/ws# g++ -S test_move.i -o test_move.s
```

```
root@fcf0e3c9bc0f:/ws# g++ -c test_move.s -o test_move.o
root@fcf0e3c9bc0f:/ws# g++ test_move.o -o test_move
root@fcf0e3c9bc0f:/ws# g++ -std=c++98 test_move.cpp
In file included from /usr/local/include/c++/11.2.0/initializer_list:36,
                 from test_move.cpp:2:
/usr/local/include/c++/11.2.0/bits/c++0x_warning.h:32:2: error: #error This file requires compiler and library support f
or the ISO C++ 2011 standard. This support must be enabled with the -std=c++11 or -std=gnu++11 compiler options.
   32 | #error This file requires compiler and library support \
root@fcf0e3c9bc0f:/ws#
test_move.cpp:16:31: warning: variadic templates only available with '-std=c++11' or '-std=gnu++11'
   16 | template <typename T, typename... Types>

test_move.cpp:17:48: warning: variadic templates only available with '-std=c++11' or '-std=gnu++11'
   17 | void printX(const T &firstArg, const Types &...args) {

test_move.cpp:25:19: warning: variadic templates only available with '-std=c++11' or '-std=gnu++11'
   25 | template <typename... Args> int maximum(int n, Args... args) {
```

```
test_move.cpp:49:15: error: expected ';' at end of member declaration
   49 |     auto head() -> decltype(m_head) { return m_head; } //使用decltype进行类型推导
                       ;
test_move.cpp:49:27: error: 'm_head' is not a type
   49 |     auto head() -> decltype(m_head) { return m_head; } //使用decltype进行类型推导

test_move.cpp:49:18: error: ISO C++ forbids declaration of 'decltype' with no type [-fpermissive]
   49 |     auto head() -> decltype(m_head) { return m_head; } //使用decltype进行类型推导

test_move.cpp:54:37: warning: variadic templates only available with '-std=c++11' or '-std=gnu++11'
   54 | template <int IDX, int MAX, typename... Args> struct PRINT_TUPLE {

test_move.cpp:56:40: error: 'tuple' does not name a type; did you mean '_tuple'?
   56 |     static void print(ostream &os, const tuple<Args...> &t) {
                                             _tuple
test_move.cpp:56:45: error: expected ',' or '...' before '<' token
   56 |     static void print(ostream &os, const tuple<Args...> &t) {

test_move.cpp: In static member function 'static void VariadicTemplates::PRINT_TUPLE<IDX, MAX, Args>::print(std::ostream
&, int)':
test_move.cpp:57:11: error: 'get' was not declared in this scope
   57 |       os << get<IDX>(t) << (IDX + 1 == MAX ? "" : ",");

test_move.cpp:57:20: error: 't' was not declared in this scope; did you mean 'tm'?
   57 |       os << get<IDX>(t) << (IDX + 1 == MAX ? "" : ",");
                        tm
```

Here is the omission of hundreds of lines are basically the standard incompatibility of the issue of warnings.

```
root@fcf0e3c9bc0f:/ws# g++ -Wall test_move.cpp
root@fcf0e3c9bc0f:/ws#
test_move.cpp:174:46: warning: typedef 'iType' locally defined but not used [-Wunused-local-typedefs]
  174 |     typedef typename decltype(obj)::value_type iType;

test_move.cpp: In function 'void Decltype::test()':
test_move.cpp:170:11: warning: 'elem' is used uninitialized [-Wuninitialized]
  170 |     cout << elem << endl;
```

```
root@fcf0e3c9bc0f:/ws# g++ -g test_move.cpp -o test_move
root@fcf0e3c9bc0f:/ws# readelf -S test_move | grep -i debug
  [27] .debug_aranges    PROGBITS         0000000000000000  0000308a
  [28] .debug_info       PROGBITS         0000000000000000  0000321a
  [29] .debug_abbrev     PROGBITS         0000000000000000  000079fa
  [30] .debug_line       PROGBITS         0000000000000000  000083ed
  [31] .debug_str        PROGBITS         0000000000000000  00008903
  [32] .debug_rnglists   PROGBITS         0000000000000000  0000be5f
  [33] .debug_line_str   PROGBITS         0000000000000000  0000bf53
root@fcf0e3c9bc0f:/ws# g++ -O0 test_move.cpp
root@fcf0e3c9bc0f:/ws# ./a.out
Process(int&):0
Process(int&&):1
Process(int&&):0
forward(int&&):2
Process(int&):2
forward(int&&):0
Process(int&):0
```

File3 test_ptr.cpp(C++14standard)

```
root@fcf0e3c9bc0f:/ws# g++ test_ptr.cpp
root@fcf0e3c9bc0f:/ws# g++ -E test_ptr.cpp -o test_ptr.i
root@fcf0e3c9bc0f:/ws# g++ -S test_ptr.i -o test_ptr.s
root@fcf0e3c9bc0f:/ws# g++ -c test_ptr.s -o test_ptr.o
root@fcf0e3c9bc0f:/ws# g++ test_ptr.s -o test_ptr
root@fcf0e3c9bc0f:/ws# ls
1.cpp            test.i          test_class.i               test_move      test_noexcept.cpp  test_ptr.s
a.out            test.o          test_class.o               test_move.cpp  test_ptr           test_raii.cpp
inefficency.cpp  test.s          test_class.s               test_move.i    test_ptr.cpp       with_o.o
test             test_class      test_class_size.cpp        test_move.o    test_ptr.i         without_o.o
test.cpp         test_class.cpp  test_default_parameter.cpp  test_move.s    test_ptr.o
root@fcf0e3c9bc0f:/ws# g++ -std=c++14 test_ptr.cpp
root@fcf0e3c9bc0f:/ws# g++ -Wall test_ptr.cpp
test_ptr.cpp: In function 'int main()':
test_ptr.cpp:11:8: warning: unused variable 'ref' [-Wunused-variable]
   11 |     int* ref = ptr;
      |

root@fcf0e3c9bc0f:/ws# g++ -g test_ptr.cpp -o test_ptr
root@fcf0e3c9bc0f:/ws# g++ -O2 test_ptr.cpp
root@fcf0e3c9bc0f:/ws# g++ -Wall test_ptr.cpp
test_ptr.cpp: In function 'int main()':
test_ptr.cpp:11:8: warning: unused variable 'ref' [-Wunused-variable]
   11 |     int* ref = ptr;
      |

root@fcf0e3c9bc0f:/ws# ./a.out
weak1 is expired
5
```

Section 3: Debugging files with GDB

Part1 For code fragment 1, make sure the stack frames and hierarchy of function calls are traced for each of the two calls to sumOfSquare(), and screenshots of the process are given.(Verdict as following)

```
root@fcf0e3c9bc0f:/ws# gdb test_function_overload
GNU gdb (Debian 10.1-1.7) 10.1.90.20210103-git
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from test_function_overload...
(gdb) list
1       #include <iostream>
2
3       using namespace std;
4
5       int sumOfSquare(int a, int b) {
6           return a * a + b * b;
7       }
8
9       double sumOfSquare(double a, double b) {
10          return a * a + b * b;
```

```
(gdb) break sumOfSquare
Breakpoint 1 at 0x4011c0: sumOfSquare. (2 locations)
(gdb) run
Starting program: /ws/test_function_overload
warning: Error disabling address space randomization: Operation not permitted
warning: File "/usr/local/lib64/libstdc++.so.6.0.29-gdb.py" auto-loading has been declined by your `auto-load safe-path' set to "$debugdir:$datadir/auto-load".
Enter two integer: bt

Breakpoint 1, sumOfSquare (a=0, b=4198608) at test_function_overload.cpp:6
6           return a * a + b * b;
(gdb) up
#1  0x0000000000401262 in main () at test_function_overload.cpp:17
17          cout << "Their sum of square: " << sumOfSquare(m, n) << endl;
(gdb) up
Initial frame selected; you cannot go up.
(gdb) down
#0  sumOfSquare (a=0, b=4198608) at test_function_overload.cpp:6
6           return a * a + b * b;
(gdb) down
Bottom (innermost) frame selected; you cannot go down.
(gdb) continue
Continuing.
Their sum of square: 1763354880

Breakpoint 1, sumOfSquare (a=2.0747120802178963e-317, b=0) at test_function_overload.cpp:10
10          return a * a + b * b;
(gdb)
```

Part2 For code snippet two, we want you to be able to trace the specific variable value of the variable y for each loop and think about the meaning of the final printed value of j.(Verdict as following)

(gdb) start The program being debugged has been started already. Start it from the beginning? (y or n) y Temporary breakpoint 3 at 0x401222: file test_range_based.cpp, line 9. Starting program: /ws/test_range_based warning: Error disabling address space randomization: Operation not permitted warning: File "/usr/local/lib64/libstdc++.so.6.0.29-gdb.py" auto-loading has been declined by your 'auto-load safe-path' set to "*debugdir* :datadir/auto-load". Temporary breakpoint 3, main () at test_range_based.cpp:9 9 int x[10] = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;
(gdb) n
11 for(int y : x) // Access by value using a copy declared as a specific type.
1: y = 4214944
(gdb) n
13 cout ¡¡ y ¡¡ " ";
1: y = 1
(gdb) n
11 for(int y : x) // Access by value using a copy declared as a specific type.
1: y = 1
(gdb) n
13 cout ¡¡ y ¡¡ " ";
1: y = 2
(gdb) n
11 for(int y : x) // Access by value using a copy declared as a specific type.
1: y = 2
.....
(gdb) n
13 cout ¡¡ y ¡¡ " ";
1: y = 10
(gdb) n
11 for(int y : x) // Access by value using a copy declared as a specific type.
1: y = 10
(gdb) n
15 cout ¡¡ endl;
(gdb) n
1 2 3 4 5 6 7 8 9 10
17 for(auto y : x) // Copy of 'x', almost always undesirable
(gdb) display y
2: y = 0
(gdb) n
18 cout ¡¡ y ¡¡ " ";
2: y = 1
(gdb) n
17 for(auto y : x) // Copy of 'x', almost always undesirable
2: y = 1
(gdb) n
18 cout ¡¡ y ¡¡ " ";
2: y = 2

```
(gdb) n
17 for( auto y : x )  // Copy of 'x', almost always undesirable
2: y = 2
......
(gdb) n
18 cout ¡¡ y ¡¡ " ";
2: y = 10
(gdb) n
17 for( auto y : x )  // Copy of 'x', almost always undesirable
2: y = 10
(gdb) n
20 cout ¡¡ endl;
(gdb) n
1 2 3 4 5 6 7 8 9 10
22 for( auto y : x )  // Type inference by reference.
(gdb) n
23 cout ¡¡ y ¡¡ " ";
(gdb) display y
3: y = (int ) @0x7ffd547622d0: 1
(gdb) n
22 for( auto y : x )  // Type inference by reference.
3: y = (int ) @0x7ffd547622d0: 1
(gdb) n
23 cout ¡¡ y ¡¡ " ";
3: y = (int ) @0x7ffd547622d4: 2
(gdb) n
22 for( auto y : x )  // Type inference by reference.
3: y = (int ) @0x7ffd547622d4: 2
......
(gdb) n
23 cout ¡¡ y ¡¡ " ";
3: y = (int ) @0x7ffd547622f4: 10
(gdb) n
22 for( auto y : x )  // Type inference by reference.
3: y = (int ) @0x7ffd547622f4: 10
(gdb) n
25 cout ¡¡ endl;
(gdb) n
1 2 3 4 5 6 7 8 9 10
27 for( const auto y : x )  // Type inference by const reference.
(gdb) n
28 cout ¡¡ y ¡¡ " ";
(gdb) display y
4: y = (const int ) @0x7ffd547622d0: 1
(gdb) n
27 for( const auto y : x )  // Type inference by const reference.
4: y = (const int ) @0x7ffd547622d0: 1
(gdb) n
```

```
28 cout ¡¡ y ¡¡ " ";
4: y = (const int ) @0x7ffd547622d4: 2
(gdb) n
27 for( const auto y : x )  // Type inference by const reference.
4: y = (const int ) @0x7ffd547622d4: 2
.....
(gdb) n
28 cout ¡¡ y ¡¡ " ";
4: y = (const int ) @0x7ffd547622f4: 10
(gdb) n
27 for( const auto y : x )  // Type inference by const reference.
4: y = (const int ) @0x7ffd547622f4: 10
(gdb) n
30 cout ¡¡ endl;
(gdb) n
1 2 3 4 5 6 7 8 9 10
31 cout ¡¡ "end of integer array test" ¡¡ endl;
(gdb) n
end of integer array test
32 cout ¡¡ endl;
34          vector<double> v;
(gdb) n
35          for (int i = 0; i < 10; ++i) {
(gdb) n
36              v.push_back(i + 0.14159);
(gdb) n
35          for (int i = 0; i < 10; ++i) {
(gdb) n
36              v.push_back(i + 0.14159);
(gdb) n
35          for (int i = 0; i < 10; ++i) {
(gdb) n
36              v.push_back(i + 0.14159);
(gdb) n
35          for (int i = 0; i < 10; ++i) {
(gdb) n
36              v.push_back(i + 0.14159);
(gdb) n
35          for (int i = 0; i < 10; ++i) {
(gdb) continue
Continuing.
0.14159 1.14159 2.14159 3.14159 4.14159 5.14159 6.14159 7.14159 8.14159 9.14159
end of vector test
```

Each value of j represents each value in the vector.

Part3 For code snippet three, we would like you to modify the code to work according to the hints in the debug and code, and tell us which of const, enum, define has the address and print out its address.(Verdict as following)

```
(gdb) run
Starting program: /ws/test_const
warning: Error disabling address space randomization: Operation not permitted
warning: File "/usr/local/lib64/libstdc++.so.6.0.29-gdb.py" auto-loading has been declined by your `auto-load safe-path'
 set to "$debugdir:$datadir/auto-load".
To enable execution of this file add
        add-auto-load-safe-path /usr/local/lib64/libstdc++.so.6.0.29-gdb.py
line to your configuration file "/root/.gdbinit".
To completely disable this security protection add
        set auto-load safe-path /
line to your configuration file "/root/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
hello
3
3
6
10
6
[Inferior 1 (process 39) exited normally]
```

The picture above shows how it run.

```
Breakpoint 1, main () at test_const.cpp:24
24              cout << p << endl;
(gdb) print p
No symbol "p" in current context.
(gdb) c
Continuing.
hello

Breakpoint 2, main () at test_const.cpp:27
27              cout << x << endl;
(gdb) print x
$1 = (const int &) @0x402004: 3
(gdb) c
Continuing.
3
(gdb)
Breakpoint 3, main () at test_const.cpp:29
29              cout << C::NUM1 << endl;
(gdb) print C::NUM1
$2 = C::NUM1
```

This picture shows that only const has its address.