```
huangs3@penguin:redefine cat main.c
#include "alib.c"
#include "blib.c"
int main(){
return a+b;
}
huangs3@penguin:redefine cat alib.c
int a=100;
huangs3@penguin:redefine cat blib.c
int b=200;
huangs3@penguin:redefine gcc main.c alib.c blib.c
```

Guess what’s will happen in this attempt to compile

```
huangs3@penguin:redefine gcc main.c alib.c blib.c
/tmp/ccNh9y1f.o:(.data+0x0): multiple definition of
`a'
/tmp/ccEcTiTd.o:(.data+0x0): first defined here
/tmp/ccSETFVj.o:(.data+0x0): multiple definition of
`b'
/tmp/ccEcTiTd.o:(.data+0x4): first defined here
collect2: ld returned 1 exit status
huangs3@penguin:redefine
```

Multiple definition occurs

```
huangs3@penguin:redefine gcc -E main.c alib.c blib.c
# 1 "main.c"
# 1 "<built-in>"
# 1 "<command line>"
# 1 "main.c"
# 1 "alib.c" 1
int a=100;
# 2 "main.c" 2
# 1 "blib.c" 1
int b=200;
# 3 "main.c" 2
```

To be continued

With “gcc –E” command we can see why it happens.

Continuing

```
int main(){
return a+b;
}
# 1 "alib.c"
# 1 "<built-in>"
# 1 "<command line>"
# 1 "alib.c"
int a=100;
# 1 "blib.c"
# 1 "<built-in>"
# 1 "<command line>"
# 1 "blib.c"
int b=200;
huangs3@penguin:redefine
```

```
huangs3@penguin:redefine gcc main.c
huangs3@penguin:redefine gcc -E main.c
# 1 "main.c"
# 1 "<built-in>"
# 1 "<command line>"
# 1 "main.c"
# 1 "alib.c" 1
int a=100;
# 2 "main.c" 2
# 1 "blib.c" 1
int b=200;
# 3 "main.c" 2
int main(){
return a+b;
}
huangs3@penguin:redefine
```

No problem if we do this

```
huangs3@penguin:bad cat main.c
#include "flib.h"
#include "ilib.h"
int main(){
float f;
int i;
f = getf();
i = geti();
return 0;
}
huangs3@penguin:bad cat flib.c
float x = 1.1;
float getf(){
return x;
}
```

```
huangs3@penguin:bad cat ilib.c
int x = 2;
int geti(){
return x;
}
huangs3@penguin:bad cat flib.h
float getf();
huangs3@penguin:bad cat ilib.h
int geti();22
```

What's going wrong?

```
huangs3@penguin:bad gcc main.c flib.c ilib.c
/tmp/cc4K9Q3P.o:(.data+0x0): multiple
definition of `x'
/tmp/ccXHzNZD.o:(.data+0x0): first defined
here
collect2: ld returned 1 exit status
huangs3@penguin:bad
```

We can use “static” to solve the problem

```
huangs3@penguin:good cat flib.c
float getf(){
static float x = 1.1;
return x;
}
huangs3@penguin:good cat ilib.c
int geti(){
static int x=2;
return x;
}
huangs3@penguin:good
```