



§. 基础知识题 – 变量类型

要求:

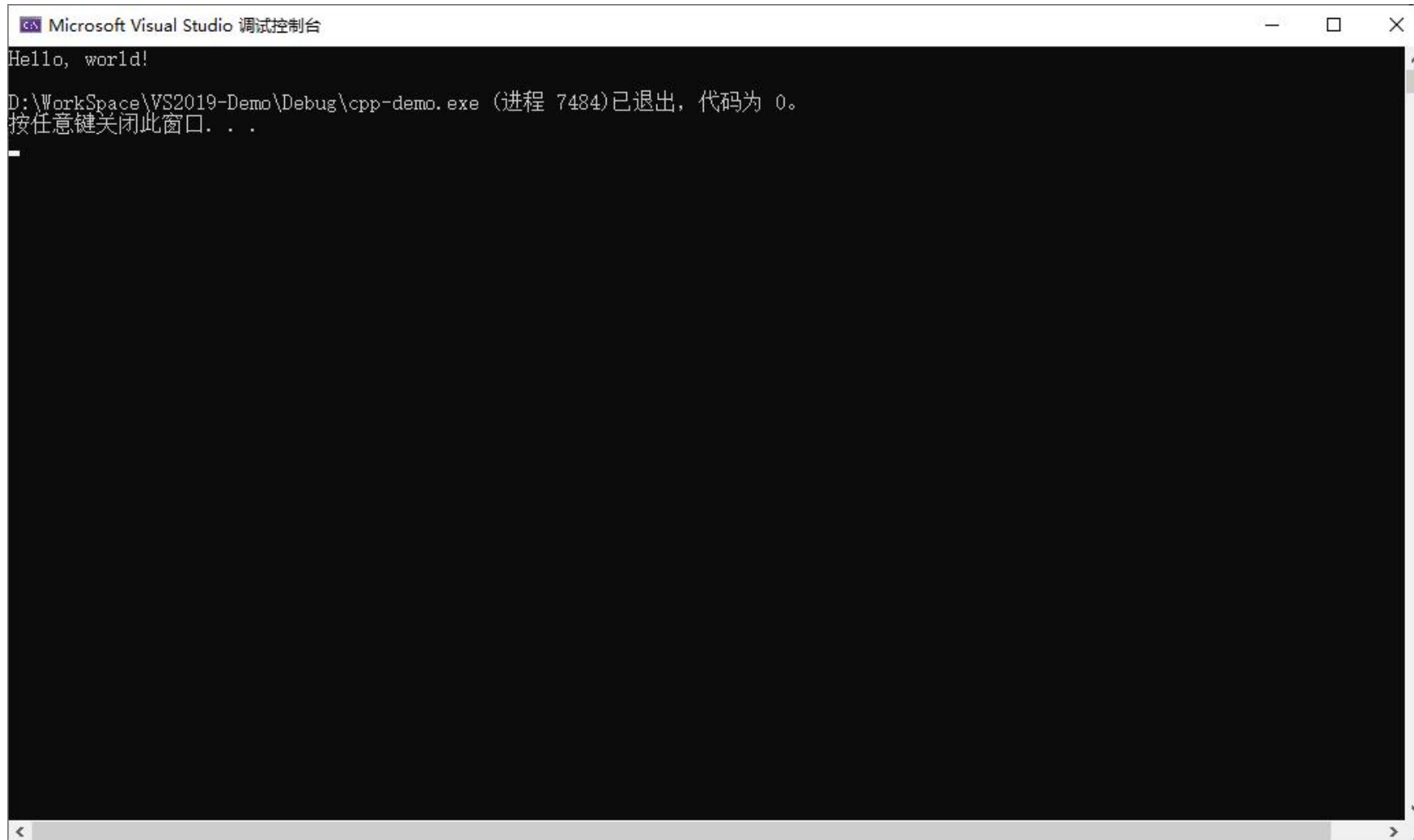
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，再拍照贴图
 - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
 - ★ 如果某题要求VS+Dev的，则如果两个编译器运行结果一致，贴VS的一张图即可，如果不一致，则两个图都要贴
- 4、转换为pdf后提交
- 5、**4月25日前**网上提交本次作业（在“文档作业”中提交）



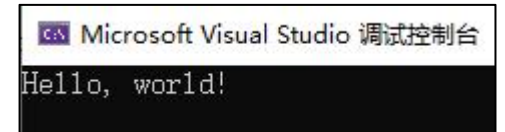
§. 基础知识题 - 变量类型

贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图



例：有效贴图

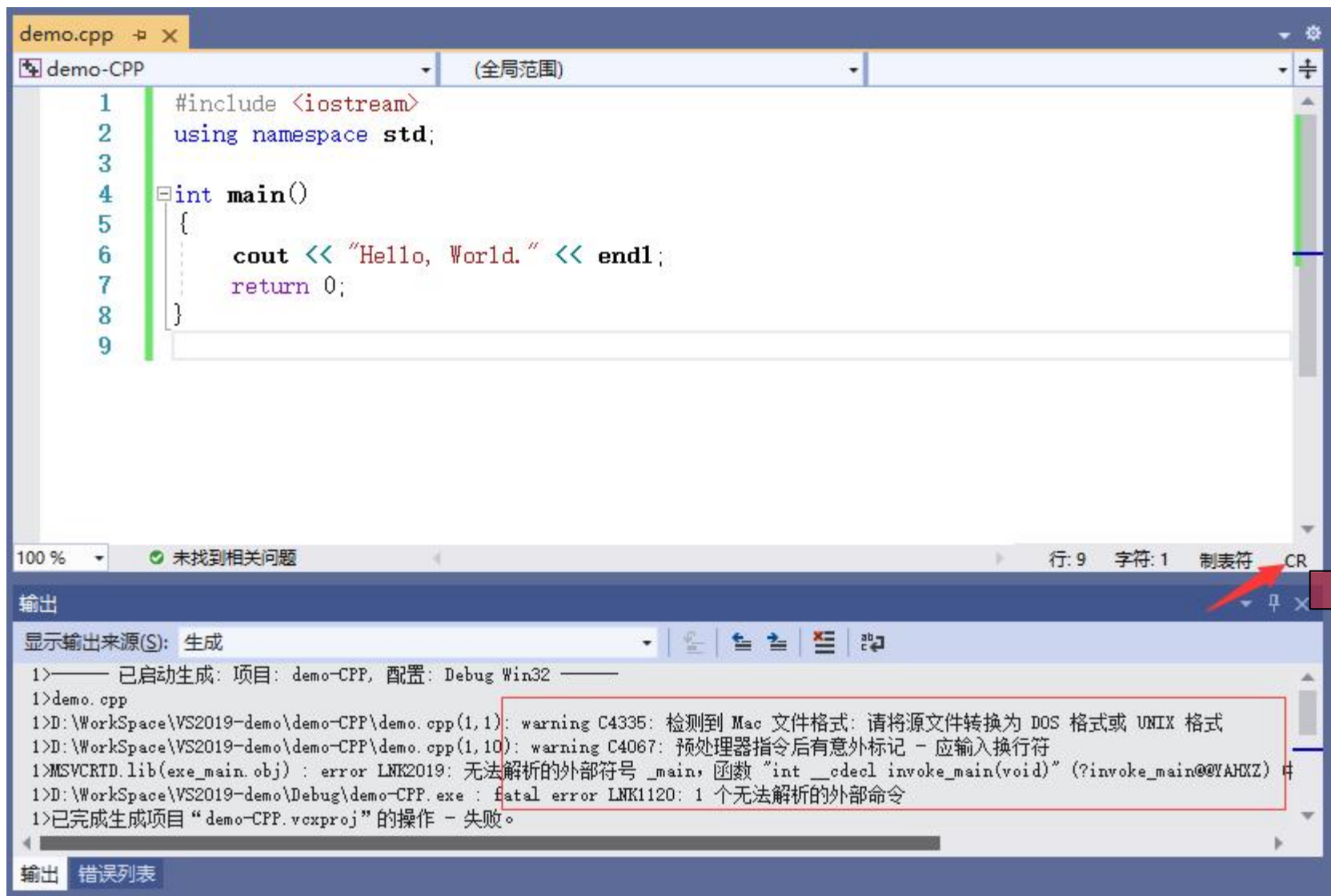




§. 基础知识题 - 变量类型

附：用WPS等其他第三方软件打开PPT，将代码复制到VS2022中后，如果出现类似下面的**编译报错**，则观察源程序编辑窗

的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可





§. 基础知识题 - 变量类型

1、自动变量及形参的分配与释放

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

void f1(int x)
{
    int y;
    cout << &x << ' ' << &y << endl; //打印地址
}

void f2(long p)
{
    float q;
    cout << &p << ' ' << &q << endl; //打印地址
}

int main()
{
    f1(10);
    f2(15L);
    return 0;
}
```

1、运行结果截图及结论

截图:

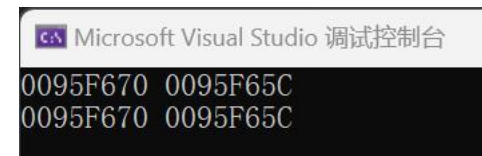


结论:

1.1 __x_和__p_共用了从__0080FE18__开始的4个字节空间

1.2 __y_和__q_共用了从__0080FE04__开始的4个字节空间

2、把f2中float q改为short q，运行结果截图及结论



结论:

2.1 q和__x_共用了从__0095F670__开始的__个字节空间



§. 基础知识题 - 变量类型

1、自动变量及形参的分配与释放

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

void f1(int x)
{
    int y;
    cout << &x << ' ' << &y << endl;
}

int main()
{
    f1(10);
    cout << "..." << endl;

    f1(10);
    cout << "..." << endl;

    f1(10);
    cout << "..." << endl;
    return 0;
}
```

1、运行结果截图



2、结论：

2.1 本示例中，三次调用时分配的x占用__相同__(相同/不同)空间，三次调用时分配的y占用__不同__(相同/不同)空间

2.2 总结形参x和自动变量y的分配和释放规则

形式参数x和自动变量y都是在调用的时候随机分配到一个地址，在函数调用结束之后马上释放。之后调用别的函数的形式参数和自动变量时会被调用同样的地址



§. 基础知识题 - 变量类型

1、自动变量及形参的分配与释放

C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

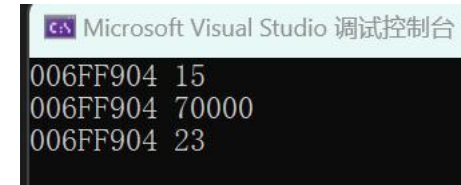
void f1()
{
    int a = 15;
    cout << &a << ' ' << a << endl;
}

void f2()
{
    long a = 70000;
    cout << &a << ' ' << a << endl;
}

void f3()
{
    short a = 23;
    cout << &a << ' ' << a << endl;
}

int main()
{
    f1();
    f2();
    f3();
    return 0;
}
```

1、运行结果截图



2、结论：

2.1 f1/f2/f3中的三个a占用__相同__(相同/不同)空间

2.2 如果当前正在执行f2函数，则f1中的a__已释放__(未分配/已释放)， f3中的a__未分配__(未分配/已释放)



§. 基础知识题 - 变量类型

1、自动变量及形参的分配与释放

D. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

void f3()
{
    short a = 23;
    cout << "f3  " << &a << ' ' << a << endl;
}

void f2()
{
    long a = 70000;
    cout << "f2-1 " << &a << ' ' << a << endl;
    f3();
    cout << "f2-2 " << endl;
}

void f1()
{
    int a = 15;
    cout << "f1-1 " << &a << ' ' << a << endl;
    f2();
    cout << "f1-2 " << endl;
}

int main()
{
    f1();
    return 0;
}
```

1、运行结果截图

Microsoft Visual Studio 调试控制台

```
f1-1 0053FC44 15
f2-1 0053FB60 70000
f3   0053FA7C 23
f2-2
f1-2
```

2、结论:

- 2.1 f1/f2/f3中的三个a占用__不同__(相同/不同)空间
- 2.2 如果当前正在执行f1函数的cout-1语句，则f2中的a__未分配__(未分配/已分配/已释放)，f3中的a__未分配__(未分配/已分配/已释放)
- 2.3 如果当前正在执行f1函数的cout-2语句，则f2中的a__已分配__(未分配/已分配/已释放)，f3中的a__已分配__(未分配/已分配/已释放)
- 2.4 如果当前正在执行f2函数的cout-1语句，则f1中的a__已分配__(未分配/已分配/已释放)，f3中的a__未分配__(未分配/已分配/已释放)
- 2.5 如果当前正在执行f2函数的cout-2语句，则f1中的a__已分配__(未分配/已分配/已释放)，f3中的a__已释放__(未分配/已分配/已释放)
- 2.6 如果当前正在执行f3函数的cout语句，则f1中的a__已分配__(未分配/已分配/已释放)，f2中的a__已分配__(未分配/已分配/已释放)
- 2.7 上述2.2~2.6问题中如果某个a是已分配状态，则此时这个a在何处?

陆续分配在f1中的a之后。

§. 基础知识题 - 变量类型



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 变量类型

2、局部变量的作用范围

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
void fun()
{
    int i, a;
    a=15;
    for(i=0; i<10; i++) {
        int y;
        y=11;
        a=16;
    }
    y=12;
    a=17;
}
int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;
    fun();
    return 0;
}
```

注：如果是error，贴error截图

如果是warning，贴warning截图+运行结果

如果正常，贴运行结果

1、截图



2、解释出现的error/warning的原因

for循环之后int y内存已经释放而下面还是调用了y，因此报错。



§. 基础知识题 - 变量类型

2、局部变量的作用范围

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
```

```
void fun()
{
    int i,a=15;
    {
        int y;
        y=11;
        a=16;
        {
            int w=10;
            y=12;
            a=13;
            w=14;
        }
        w=15;
    }
    y=12;
    a=17;
}
```

```
int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;
    fun();
    return 0;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图

代码	说明	项目	文件
E0020	未定义标识符 "w"	demo	源1.cpp
E0020	未定义标识符 "y"	demo	源1.cpp
C2065	"w": 未声明的标识符	demo	源1.cpp
C2065	"y": 未声明的标识符	demo	源1.cpp

2、解释出现的error/warning的原因

在最内层的大括号之中开辟了int w但是在结束大括号之后也已经释放，同理外层括号之外还有一个y也已经释放了。



§. 基础知识题 - 变量类型

2、局部变量的作用范围

C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

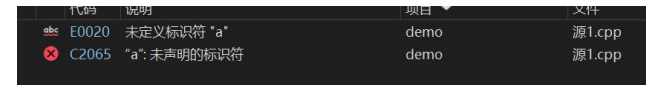
void fun()
{
    a=14;
}

int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;

    int a;
    a=15;
    fun();
    a=16;
    return 0;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图



2、结论：在某个函数(main)中定义的自动变量，在它的调用函数(fun)中____不允许__(允许/不允许)访问

§. 基础知识题 - 变量类型



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 变量类型

3、全局变量的作用范围

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

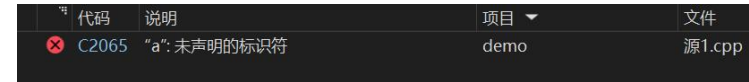
int f1()
{
    a=15;
}

int a;
int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;
    a=16;
    return 0;
}

int f2()
{
    a=17;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图



2、解释出现的error/warning的原因

全局变量的命名在函数int f1之后因此出现问题



§. 基础知识题 - 变量类型

3、全局变量的作用范围

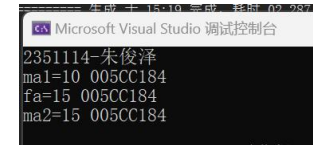
B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
int a;
void f1()
{
    a=15;
    cout << "fa=" << a << ' ' << &a << endl;
}
int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;

    a=10;
    cout << "ma1=" << a << ' ' << &a << endl;
    f1();
    cout << "ma2=" << a << ' ' << &a << endl;
    return 0;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图



2、由运行结果中的地址可以证明，f1和main中访问的变量a____相同__(不同/相同)的a



§. 基础知识题 - 变量类型

3、全局变量的作用范围

C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

int f1(int a)
{
    a=15;
    cout << "fa=" << a << ' ' << &a << endl;
    return a;
}

int main()
{
    /* 注意：输出必须改为自己学号-姓名 */
    cout << "1234567-张三" << endl;

    int a =10;
    cout << "ma1=" << a << ' ' << &a << endl;
    a = f1(a);
    cout << "ma2=" << a << ' ' << &a << endl;
    return 0;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图



2、由运行结果中的地址可以证明，f1和main中访问的变量a___不同___(不同/相同)的a

3、a不是全局变量，解释为什么ma1和ma2两句cout输出的a值不相同？a是如何被改变的？

把main中的a作为形式参数传给f1，之后f1return了一个值给a，也就是15，因此改变了a

§. 基础知识题 - 变量类型



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 变量类型

4、变量同名

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

int a=10, b;
void f1()
{
    int a=5, b;
    cout << "a1=" << a << ' ' << &a << endl;
    cout << "b1=" << b << ' ' << &b << endl;
}
void f2()
{
    cout << "a2=" << a << ' ' << &a << endl;
    cout << "b2=" << b << ' ' << &b << endl;
}
int main()
{
    f1();
    f2();
    return 0;
}
```

注：如果是error，贴error截图

如果是warning，贴warning截图+运行结果

如果正常，贴运行结果

1、截图



2、由b可知，局部变量不初始化，初值为__随机的值__；
全局变量不初始化，初值为__0__

3、由截图可知，全局变量a/b的起始地址差_320_个字节；
局部变量a/b之间差_12_个字节；全局和局部之前差
__9mb__(单位KB/MB均可)，说明这是两个不同的存储区，
全局变量在__静态__存储区，局部变量在__动态__存储区。



§. 基础知识题 - 变量类型

4、变量同名

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;
```

```
int a=10;
short a;
```

```
void f1()
{
```

```
    int x=5;
```

```
    double x=1.2;
```

```
    short p=1, p=2;
```

```
}
```

```
int main()
```

```
{
```

```
    f1();
```

```
    return 0;
```

```
}
```

注：如果是error，贴error截图

如果是warning，贴warning截图+运行结果

如果正常，贴运行结果

1、截图

代码	说明	项目	文件
✖ C2371	"a": 重定义; 不同的基类型	demo	源1.cpp
✖ C2371	"x": 重定义; 不同的基类型	demo	源1.cpp
✖ C2374	"p": 重定义; 多次初始化	demo	源1.cpp

2、结合4. A/4. B可以得知：__不同__(相同/不同)级别的变量允许同名； __相同__(相同/不同)级别的变量不允许同名；变量同名是的使用规则是____优先使用本级别的变量_____。

§. 基础知识题 - 变量类型



此页不要删除，也没有意义，仅仅为了分隔题目



§. 基础知识题 - 变量类型

5、自动变量与静态局部变量

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
using namespace std;

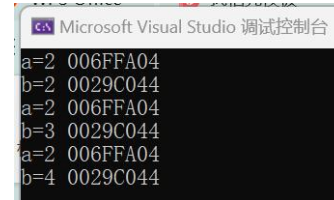
void f1()
{
    int a=1;
    a++;
    cout << "a=" << a << ' ' << &a << endl;

    static int b=1;
    b++;
    cout << "b=" << b << ' ' << &b << endl;
}

int main()
{
    f1();
    f1();
    f1();
    return 0;
}
```

注：如果是error，贴error截图
如果是warning，贴warning截图+运行结果
如果正常，贴运行结果

1、截图



2、结合a/b各自的地址和值，得到结论为：

自动变量a多次调用，则__每次__(每次/仅第一次)进行初始化，函数运行结束后__会__(会/不会)释放空间，下次进入时__继续使用上次的空间__(再次分配/继续使用上次的空间)

静态局部变量a多次调用，则__仅第一次__(每次/仅第一次)进行初始化，函数运行结束后__不会__(会/不会)释放空间，下次进入时__继续使用上次的空间__(再次分配/继续使用上次的空间)

根据上面的分析结果，自动变量应该放在__动态数据区__(动态数据区/静态数据区)，静态局部变量应该放在__静态数据区__(动态数据区/静态数据区)

§. 基础知识题 - 变量类型



此页不要删除，也没有意义，仅仅为了分隔题目