1： 讲解基本数据类型：

Int long float double char

它们和java 的使用方式基本相同。大家可以将这些基本类型当成Java 使用就可以了。

2： 同时讲解sizeof 操作符：

这个并不是函数，而是一个单目运算符。

以字节的形式给出了其操作数的存储大小。操作数可以是一个表达式或者是类型名，且必须有括号，操作数的存储大小由操作数的类型决定。

用途： 主要用途是与存储分配

运算符不是函数

1，运算符是内置于语言的，函数是所带的库里面的

2. 函数的名字可以随便你起，但是运算符去不能，运算符不能自定义，但是能重载

2： 讲解基本的for switch while if break contiue控制流 与java 的一致，所以大家就可以把他们当成简单的Java使用就可以了。

3： 函数定义，头文件，引用

3： 讲解printf & scanf getchar

输入输出函数。这个类似于java 中的System.out.printf 和

而输入就麻烦很多：

1. 基本输入流是InputStream，这是一个抽象类。利用他的子类BufferedInputStream可以读取键盘输入数据。  
   BufferedInputStream keyboard = new BufferedInputStream(System.in);  
   2、Readers流是以字符流的方式读写字节流的对象。他也是抽象类，利用其子类InputStreamReader可以读取键盘数据  
   InputStreamReader keyboard = new InputStreamReader(System.in);  
   3、你还可以利用标准流定义自己的流操作。就是定义换行等！

4： 数组

关于数组的了解。用很大不同来形容。

第一个： 表达式写法不同

Java Int[] array;

C int[10] array;

二： 大小的确定性，

Java 定义的时候可以不确定大小，用的时候动态申请

C 语音定义的时候必须先确定大小。

C语言中 数组是占用一段联系的存储空间的。

定义的时候，java的数组的大小可以不确定，但是C 语言必须知道数组大小，是确定的。

调试内存时： 调试-窗口-内存

4： 讲解指针。

1.

& 地址操作符只能应用于内存中的对象，即变量与数组元素，它不能作用于表达式、常量或者register 类型的变量。

\*一元运算符，是间接寻址或间接引用运算符。当作用于指针时，是指针所指向的对象。

举例说明，

2.

指针只能指向某种特定类型的对象。也就是说，每个指针都指向某个特定的数据类型。

例外： void 类型的指针可以存放指向任何类型的指针。

写成 int \*p 的目的是凸显 \*p 的结果是int 类型数据。

3.运算符的优先级和结合顺

\*p = \*p = 1;

\*p +=1;

++\*p;

(\*p) ++ ;

等价，且，最后一个的括号必须要有，因为++右结合的单目运算符

例如： 2 + 5 \* 4

4. 指针可以赋值给指针变量,

Int \* ip；

Int \* ip2;

Ip = ip2;

因为指针也是变量

讲解形参用指针和普通值传参数和java 的区别

总结： 指针有点像Java 的引用，对不对，但是更加的灵活，复杂，而java 就更容易使用更简单，但是没有这么灵活。

Java 引用和C语言指针的区别

<http://blog.csdn.net/guo18735177413/article/details/42705961>

5： 讲解外挂（jason课程）