本试卷分三部分，公共部分、C++部分和JAVA部分，其中公共部分为必考部分，C++与JAV

部分，两者选一完成。试卷满分100分。

**公共部分(50分)**

5、在TCP/IP层次模型中与OSI参考模型第四层相对应的主要协议有\_\_\_\_\_\_\_\_\_\_\_\_        和            ，其中后者提供无连接的不可靠传输服。TCP（传输控制协议）和UDP（用户数据报协议） (顺序可交换) （4分）

6、在OSI中，完成相邻节点间流量控制功能的层次是              。数据链路层（2分）

5、IP地址为 140.111.0.0 的B类网络，若要切割为9个子网，而且都要 连上Internet，请问子网掩码设为（D ）。（4分）

A. 255.0.0.0

B. 255.255.0.0  
C. 255.255.128.0

D. 255.255.240.0

6、以下哪一个不是关于千兆位以太网的正确描述（ C ）。（4分）

A. 数据传输速率为1000MBit/S

B. 支持全双工传送方式

C. 只能基于光纤实现

D. 帧格式与以太网帧格式相同

3. 设有98个已排序元素,采用二分法查找时,最大比较次数是（ D ）（4分）

1. 49
2. 15
3. 20
4. 7

4. 高度为h的完全二叉树结点数最多有（ A ）（4分）

1. 2h-1
2. 2h+1
3. 2h-1
4. 2h+1

5. Hash表是用于数据存储的一种有效的数据结构，Hash表的查找复杂度依赖于Hash值算法的有效性，在最好的情况下，Hash表的查找复杂度为（ A ）；在最坏的情况下，Hsah表的查找复杂度为（ C ）（4分）

1. O(1)
2. O(logn)
3. O(n)
4. O(nlogn)

1. 写出进程的主要状态。（12分）

运行态（running）（2分），就绪态（ready）（2分），等待态（blocked）（2分），挂起（suspend）（2分），新建态（new）（2分），终止态（exit）（2分）

1. TCP为何采用三次握手来建立连接，若采用二次握手可以吗？为什么？（12分）

三次握手是为了防止已失效的连接请求再次传送到服务器端。 （4分）

二次握手不可行，（2分）

因为：如果由于网络不稳定，虽然客户端以前发送的连接请求以到达服务方，但服务方的同意连接的应答未能到达客户端。则客户方要重新发送连接请求，若采用二次握手，服务方收到重传的请求连接后，会以为是新的请求，就会发送同意连接报文，并新开进程提供服务，这样会造成服务方资源的无谓浪费。（6分）

**C++部分（50分）**

1. 设一维整型数组data已进行了初始化，则其元素个数可由操作符sizeof通过表达式\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_计算出来。sizeof(data/data[0]) （2分）
2. 在继承机制下，当对象消亡时，编译系统先执行\_\_\_\_\_\_\_\_的析构函数，然后再执行\_\_\_\_\_\_\_的析构函数。子类（2分）   父类（2分）
3. 设有“int w[3][4]；”，pw是与数组名w等价的数组指针，则pw的初始化语句为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ int (\* pw)[4]=w; （2分）

4.  要使引用pr代表变量“char \* p；”，则pr的初始化语句为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_       。char \* &pr=p；（2分）

5. 引用与指针有什么区别？（12分）

1) 当引用被创建时，它必须被初始化。（指针则可以在任何时候被初始化。）（4分）

2) 一旦一个引用被初始化为指向一个对象，它就不能被改变为对另一个对象的引用。（指针则可以在任何时候指向另一个对象。）（4分）

3) 不可能有N U L L引用。必须确保引用是和一块合法的存储单元关连。（4分）

6. 请问以下程序将输出什么结果？（16分）

char \*RetMemory(void)

{

char p[] = “hello world”;

return p;

}

void Test(void)

{

char \*str = NULL;

str = RetMemory ();

printf(str);

}

可能是乱码。

因为GetMemory返回的是指向“栈内存”的指针，该指针的地址不是 NULL，但其原现的内容已经被清除，新内容不可知。

7. 程序员规范中要求不要写出类似(++i)+(i++) 或 f(++i,i++)这样的代码，请说明原因。（12分）

计算子表达式的顺序由编译器决定的，虽然参数的压栈顺序在给定的调用方式下式固定的， 但参数表达式的计算顺序也由编译器决定的。不同的编译器或不同的表达式计算的顺序可能不一致。

**JAVA部分（50分）**

1.               方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。            方法是不能被当前类的子类重新定义的方法。

抽象(abstract)方法（2分）、最终(final)方法（2分）

2. 以下那个方法声明能够被加入到Child类中，编译正确？（ ）B（4分）

public class Parent {

public int addValue( int a, int b) { return a+b; }

}

class Child extends Parent {

}

1. int addValue( int a, int b ){// do something…}
2. public void addValue (){// do something…}
3. public void addValue( int b, int a ){// do something…}

D.  public int addValue( int a, int b )throws MyException {//do something…}

3. 设 float x = 1 , y = 2 , z = 3，则表达式  y＋＝z－－/＋＋x  的值是(      )。A（4分）

A.    3.5                       B.    3

C.    4                        D.    5

4、在Java  Applet程序用户自定义的Applet子类中，一般需要重载父类的(      )方法来完成一些画图操作。D（4分）

A.   start( )                    B.   stop( )

C.   init( )                     D.   paint( )

E. 以上都不对

5. 如果一个对象仅仅声明实现了cloneable接口，但是不声明clone方法，外部能够调用其clone方法吗？（ ）B（4分）

1. 能
2. 不能

C.  不确定

6. 方法重载Overload和方法的重写Overriding的区别。Overloaded的方法是否可以改变返回值的类型? （12分）

方法的重写Overriding和重载Overloading是Java多态性的不同表现。（2分）

重写Overriding是父类与子类之间多态性的一种表现，（3分）

重载Overloading是一个类中多态性的一种表现。（3分）

如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。（4分）

7. 写个程序，要求创建一个文件，写入从键盘输入的一串字符，然后再读该文件并将文件内容显示在屏幕上。（18分）

import java.io.\*;

class MyFileIo{

public static void main(String args[]){

FileInputStream fin;

FileOutputStream fout;

char ch;

int data;

try{

fin = new FileInputStream(FileDescriptor.in);

fout = new FileOutputStream(“myfile1”);

System.out.println(“请输入一串字符以#结束：”);

while((ch = (char)fin.read()) != ‘#’)

Fout.write(ch);

fin.close();

fou.close();

System.out.print(“”);

fin = new FileInputStream(“myfile1”);

fout = new FileOutpurStream(FileDescriptor.out);

while(fin.available() > 0){

data = fin.read();

fou.write(data);

}

fin.close();

fout.close();

}catch(FileNotFoundException e){

System.out.println(“File not found!”);

}carch(IOException e){}

}

}