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)**

1：作业调度程序从处于\_\_\_ D \_\_\_\_状态的队列中选择适当的作业投入运行。(3分)

A.运行   B.提交    C.完成    D.后备

2. SQL语言中，删除一个表的命令是 （ B ）(3分)

A.CLEAR TABLE    B.DROP TABLE    C.DELETE TABLE    D.REMOVE TABLE

3：ATM采用的线路复用方式为   C    。 (3分)

A. 频分多路复用

B. 同步时分多路复用

C. 异步时分多路复用

D. 独占信道

4：数据库中只存放视图的 （ C ）(3分)

A．操作         B.对应的数据

C．定义         D.限制

5：什么是虚拟设备？为什么在操作系统中引入虚拟设备？(10分)

虚拟设备是通过虚拟技术将一台独占设备变换为若干台逻辑设备，供若干个用户进程同时使用，通常把这种经过虚拟技术处理后的设备称为虚拟设备。(5分)

在操作系统设备管理中，引入虚拟设备是为了克服独占设备速度较慢、降低设备资源利用率的缺点，从而提高设备的利用率。(5分)

6：TCP为何采用三次握手来建立连接，若采用二次握手可以吗,请说明原因？(16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。 (6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(8分)。

7：什么是分布式数据库？(12分,每点4分)

数据库分布在计算机网络的不同计算机上，

网络中的每个节点具有独立处理的能力（场地自治），可以执行局部应用，

同时，每个节点也能通过网络通信子系统执行全局应用。

**C++部分(50分)**

1. 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；(3分)

1. 要使引用pr代表变量“char \* p；”，则pr的初始化语句为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_       。

char \* &pr=p；(3分)

3.“零值”可以是0, 0.0 , FALSE或者“空指针”。例如 int 变量 n 与“零值”比较的 if 语句为：if ( n == 0 )，则BOOL flag 与“零值”比较的 if 语句为：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；float x 与“零值”比较的 if 语句为：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。(6分)

if ( flag )

const float EPSINON = 0.00001;if ((x >= – EPSINON) && (x <= EPSINON))

4. 设有如下语句：(3分)

int x；

(    ）

\*px=0；

则选择以下哪条语句可将x值置为0。C

A  int \* px；              B  int const \* px=&x；

C  int \* const px=&x；      D  const int \* px=&x；

5. 设void f1(int \* m，long & n)；int a；long b；则以下调用合法的是（    ）。B(3分)

A  f1(a，b)；              B  f1(&a，b)；

C  f1(a，&b)；            D  f1(&a，&b)；

6. 请写出如下代码的运行结果(6分)

int main()

{

int a,b,c,d;

a=0;

b=1;

c=2;

d=3;

printf(“%d”,a+++b+c+++d++);

}

6

7. 写出下面函数的功能(12分)

template <class Type>

void WE(Type a[],Type b[],int n) {

for (int i=0; i<n; i++)

b[n-i-1]=a[i];

}

模板函数，把数组a的每个元素按逆序放入数组b中。

8. 写一段代码判断一个单向链表中是否有环。(14分)

给出如下结构：

struct node

{

struct \*next;

};

typedef stuct node Node;

bool getCycle(){

Node\* temp1 = head;//(假设head就是这个链表的头)

Node\* temp2 = head;

while(head->next!=NULL)

{

temp1 = temp1->next;//步长为1

temp2 = temp2->next->next;//步长为2

if(temp1 == temp2)

return ture;

head=head->next;

}

return false;

}

**JAVA部分(50分)**

1. 判断下列对的是哪个（）b(3分)

A、short s1 = 1; s1 = s1 + 1;

B、short s1 = 1; s1 += 1;

C、short s1 = 1; s1 = s1 – 1;

D、short s1 = 1; s1 = s1 \* 1;

2. main方法是Java  Application程序执行的入口点，关于main方法的方法头以下哪项是合法的（     ）B(3分)

A、public  static  void  main（ ）

B、public  static  void   main（ String  args[] ）

C、public static int  main（String  [] arg ）

D、public  void  main（String  arg[] ）

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

A.    3.5                       B.    3

C.    4                        D.    5

4、Math.round(11.5)=\_\_\_\_\_\_\_\_ (4分)

Math.round(-11.5)=\_\_\_\_\_\_\_\_

12，-11

5. 假设x= 10, y= 20, z= 30; 计算下列表达式的值(6分)

A) x <10 || x > 10 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    B) !(x < y + z) || (x + 10  <= 20) \_\_\_\_\_\_\_\_\_\_\_\_\_

C) z-y == x && Math.abs(y-z) != x

false   true  false

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? (10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， 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。(3分)

重载Overloading是一个类中多态性的一种表现。 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)。(3分)

Overloaded的方法是可以改变返回值的类型。(2分)

7. 谈谈HashMap和Hashtable的区别？(6分)

都属于Map接口的类，实现了将惟一键映射到特定的值上。(2分)  
HashMap 类没有分类或者排序。它允许一个 null 键和多个 null 值。 (2分)  
Hashtable 类似于 HashMap，但是不允许 null 键和 null 值。它也比 HashMap 慢，因为它是同步的。(2分)

8. 构造一个类来描述屏幕上的一个点，该类的构成包括点的x和y两个坐标，以及一些对点进行的操作，包括：取得点的坐标值，对点的坐标进行赋值，编写应用程序生成该类的对象并对其进行操作。(15分)

import java.io.\*;

public class Point{  
int x,y;  
}  
public Point(int x,int y){  
this.x = x;  
this.y = y;  
}  
public Point getPoint(){  
Point tempPoint = new Point(x,y);  
return tempPoint;  
}  
public void setPoint(Point point){  
this.x = point.x;  
this.y = point.y;  
}  
public static void main(String args[]){  
Point Point1 = new Point(3,4);  
System.out.println(“Point1:”+”(“+Point1.x+”,”+Point1.y+”)”);

Point Point2 = Point1.getPoint();  
System.out.println(“Point2:”+”(“+Point2.x+”,”+Point2.y+”)”);

Point Point3 = new Point(5,6);  
Point1.setPoint(Point3);  
System.out.println(“Point1:”+”(“+Point1.x+”,”+Point1.y+”)”);  
}  
}