

天津大学 2014-2015 学年 C++期末考试样卷 及答案

一. 单项选择题(每题 1 分, 共 20 分)

1. C++源程序文件的扩展名是 。 B

A) .C B) .CPP C) .DLL D) .EXE

2. 下列属于 C++合法的字符常量是 C 。

A) '\97' B) "A" C) '\t' D) "\0"

3. 设 `const int B=6; int a[12];` 则下列错误的是 _____。 A

A) `B+=3;` B) `a[5]=B+3;` C) `int x[B+3];` D) `a[B+3]=25;`

4. a、b 均为 int 型且 $b \neq 0$, 则表达式 $a-a/b*b$ 的值为 C 。

A) 0 B) a C) a 被 b 除的余数部分 D) a 被 b 除商的整数部分

5. 设 `int a=5,b=6,c=7;` 则 `cout<<((a+b)<c&&b!=c)<<","<<((a+b)<c,b=c);` 的输出是 B 。

A) 0,0 B) 0,7 C) 1,0 D) 1, 7

6. 以下关于编译预处理的叙述中, 错误的是 C 。

A) 预处理命令必须以“#”开始 B) 一条预处理命令行必须独占一行

C) 一条预处理命令行必须以分号结束 D) 预处理命令是在正式编译之前先被处理的

7. 设 int 型数据占 4 个字节, 若有定义: `int x[10]={0,2,4};` 则 `sizeof(x)` 的值为 C 。

A) 3 B) 12 C) 40 D) 10

8. 下列关于 C++函数的叙述中, 正确的是 。

A) 每个函数至少都要有一个形参 B) C 每个函数都必须有返回值

C) 函数在被调用之前必须先声明 D) 函数不能自己调用自己

9. 下面有关重载函数的说法中正确的是 C 。
- A) 重载函数必须具有不同的返回值类型 B) 重载函数形参个数必须不同
- C) 重载函数必须有不同的形参表 D) 重载函数名可以不同
10. 表达式 `strcmp("box","boss")` 的值是 A 。
- A) 正整数 B) 负整数 C) 0 D) 不确定的数
11. 设 `int a[3][4], *p=a[0];` 以下不能正确访问数组元素 `a[i][j]` 的是 A 。
- A) `*(a+i+j)` B) `*(*(a+i)+j)` C) `*(a[i]+j)` D) `p[4*i+j]`
12. 当一个派生类公有继承一个基类时，基类的所有 `public` 成员成为派生类的 D 。
- A) 友元 B) `private` 成员 C) `protected` 成员 D) `public` 成员
13. 设 `int a=3, *p=&a;` 则 `*p` 的值是 D 。
- A) 变量 `a` 的地址值 B) 无意义 C) 变量 `p` 的地址值 D) 3
14. 设 `int i=5, f=15;` 则执行语句 `if(i=0) f=5;` 后 `i` 和 `f` 的值分别是 C 。
- A) 5, 15 B) 5, 10 C) 0, 15 D) 0, 10
15. 设 `int *p;` 以下对 `new` 运算符的正确使用形式是 C 。
- A) `p=new a[10];` B) `*p=new int[10];`
- C) `p=new int[10];` D) `p=new int[5]={1,2,3,4,5,6};`
16. 下列关于动态联编的描述错误的是 C。
- A) 动态联编在运行时选择联编函数 B) 动态联编也称为动态多态性
- C) 函数重载和运算符重载属于动态联编 D) 动态联编可用来选择虚函数
17. 设二叉树的高度为 $k(k \geq 0)$ ，若其包含的结点总数为 $2^{k+1}-1$ ，则该二叉树是 B 。
- A) 完全二叉树 B) 满二叉树 C) 二叉排序树 D) 普通的二叉树
18. 设用一个大小为 6 的一维数组存储顺序循环队列，若当前 `front` 和 `rear` 的值分别为 3 和 5，当从队列中删除一个元素，再加入两个元素后，`front` 和 `rear` 的值分别是 D 。
- A) 5 和 0 B) 4 和 7 C) 无法完成要求的操作 D) 4 和 1

19. 以下关于关键字的描述中错误的是 C 。

A) 关键字可由元素的一个或多个数据项组成 B) 可以按关键字进行查找

C) 不同元素的关键字值可以相同 D) 每个元素的关键字必须是唯一的

20. 直接插入排序在被排序的数据 B 时效率较高。

A) 完全无序 B) 基本有序 C) 逆序排列 D) 最大值与最小值相差不大

二. 写出以下程序的运行结果（每题 4 分，共 20 分）

1. #include <iostream.h>

```
void main(){
```

```
int a[8]={12,25,33,5,8,16,51,22},i;
```

```
int *p=a;
```

```
for(i=0;i<4;i++)
```

```
cout<<*(p+i*2)<<endl;
```

```
}
```

程序运行的结果是：

12 33 8 51

2. #include <iostream.h>

```
int fun(){
```

```
static int d2=0,d1=1;
```

```
int t=d2+d1;
```

```
d1=d2;
```

```
d2=t;
```

```
return d2;
```

```
}
```

```
void main(){
```

```

for(int i=0;i<6;i++)

cout<<fun()<<" ";

cout<<endl;

}

```

程序运行的结果是： 1 1 2 3 5 8

3. #include <iostream.h>

```

class A{

public:

A(){ a=b=10; }

A(int i,int j) { a=i; b=j; }

void operator+=(const A &x){ a+=x.a; b+=x.b; }

void print();

private:

int a,b;

};

void A::print(){ cout<<"a="<< a<<" ,b="<< b<<endl; }

void main(){

A m,n(15,18);

m.print(); n.print();

m+=n;

m.print(); n.print();

}

```

程序运行的结果是：

a=10, b=10

a=15,b=18

a=25,b=28

a=15,b=18

4. #include <iostream.h>

void main(){

int x=3,y=3;

switch(x%2) {

case 1: switch (y%x){

case 0:cout<<"first\n";

case 1:cout<<"second\n";break;

default: cout<<"hellow\n";

}

case 2:cout<<"third\n";

}

}

程序运行的结果是：

first

second

third

5. #include <iostream.h>

#include <string.h>

void fun(char *str){

char t;

int i,j;

```

for(i=1;i<strlen(str);i++){

t=str[i];

for(j=i-1;j>=0&& t<str[j];j--)

str[j+1]=str[j];

str[j+1]=t;

}

}

```

```

void main(){

char s[]="CFBEAD";

fun(s);

cout<<s<<endl;

}

```

程序运行的结果是：ABCDEF

```

6 .#include <iostream.h>

class point{

private:

int x,y;

public:

point(int x1=0,int y1=0){

x=x1;

y=y1;

}

virtual void move(int x1,int y1){

x+=x1;

```

```

y+=y1;

}

virtual void disp(){

cout<<"点: ("<<x<<','<<y<<');

}

};

class line:public point{

private:

int xx,yy;

public:

line(int x1,int y1,int x2,int y2):point(x1,y1){

xx=x2;

yy=y2;

}

void move(int x1,int y1){

point::move(x1,y1);

xx+=x1;

yy+=y1;

}

void disp(){

point::disp ();

cout<<"-点: ("<<xx<<','<<yy<<');

}

};

```

```

void main(){

line l(2,4,6,8);

point p(l),*pp=&l;

p.move (3,3);

pp->move (4,4);

p.disp ();

cout<<endl;

pp->disp ();

cout<<endl;

}

```

结果：

点： (5,7)

点： (6,8)-点： (10,12)

Press any key to continue

三. 程序填空（每空 2 分，共 32 分）：

1. 设一个数列的第 1、2 项分别为 0 和 1，以后每个奇数编号的项是前两项之和，偶数编号项是前两项之差的绝对值。下面程序的功能是生成该数列的前 20 项并存入一个数组 x，然后将这 20 个数存入文本文件 data.txt。

```

#include <fstream.h>

#include <math.h>

void main() {

int x[20],i;

x[0]=0;x[1]=1

i=2;

do{

```



```

x[i]= ; //x[i-1]+x[i-2]

x[i+1]=abs(x[i-1]-x[i]);

i= ; //i+2

}while( ); //i<19

;

if(!fout) return;

for( ;i<20 ; j++){ //i=0

fout<<x[i]<<" ";

fout.close();

}

```

2. 以下程序实现将任意正整数转换成相应字符串。例如，将 1234 转换为“1234”。

```

#include <iostream.h>

class Cstack{ //定义字符栈

char s[80];

int top; //栈顶指示器

public:

Cstack() { top=-1 ;} //把栈置为空

void push(char ch){ //将 ch 入栈

if(top==79) return;

s[++top]= ch ;

}

char pop(){ //出栈并返回栈顶元素值（栈空时返回 0）

if(top== -1)return 0;

return ; //s[top--]

```

```

}

};

void itos(int num, char *str){ //将 num 转换为字符串并存入 str

Cstack s1;

while(num!=0){

s1.push( ); // '0'+num%10

num/=10;

}

char ch;

for(int i=0;(ch=s1.pop())!=0;i++) ; //str[i]=ch

str[i]=0;

}

void main(){ char s[80]; itos(1234,s); cout<<s<<endl; }

```

3. 以下函数的功能是删除字符串最前面和最后面的所有空格。

```

#include <string.h>

void DelSpace(char *s){

for(int i=strlen(s)-1;i>=0;i--) if(s[i]!=' ') break; //删除后面的空格

s[ ]=0; //i+1

char *p=s;

while(*s!=0 ) s++; //删除前面的空格

strcpy( ); //p,s

}

```

4. 以下程序输出数组 a 中重复出现最多的那个数以及该数重复出现的次数和首次出现的位置。

```

#include <iostream.h>

void main(){

int i,j,k,t, n(0); //M=20

int a[20]={1,2,3,4,5,6,7,8,4,5,4,5,4,6,7,1,2,3,4,5};

for(i=0;i<M-1;i++){

for( k=1;j=i+1 ;j<M;j++)

if( ) k++; //a[i]==a[j]

if(n<k){ ;t=i;} //n=k

}

cout<<"重复最多的数是: "<<a[t]<<" 重复次数: "<<n<<" 首次位置: "<<t<<endl;

}

```

5. 在 Chain 类中增加成员函数 Insertx，函数的功能是：在表中查找有无值为 x 的元素，若有，则显示“已存在”，否则，将值为 x 的元素插到表头

```

void insertx(char x){

Node *p = head->next , *q;

while( ) p = p -> next; //p!=NULL&&p->data!=x

if( p==NULL){

q= ; //new Node

q->data=x;

= head->next; //q->next

head->next=q;

}

else

cout<<x<<"已存在!"\n;

```

}

四. 简答题 (12 分)

1. 假设 S 和 X 分别表示进栈和出栈操作, 由输入序列 “ABC” 得到输出序列 “BCA” 的操作序列为 SSXSXX, 则由 “a*b+c/d” 得到 “ab*cd/+” 的操作序列什么? (4 分)。

SXSSXXSSXSXX

2. 已知一个散列表如下图所示:

51	64		29				20		35			38
0	1	2	3	4	5	6	7	8	9	10	11	12

其散列函数为 $h(\text{key}) = \text{key} \% 13$, 处理冲突的方法为现行地址序列探查法。

请回答下列问题:

(1) 将关键字 35, 20, 33, 38, 51, 64 存入哈希表。(3 分)

(2) 对表中关键字 29 和 64 进行查找时, 所需进行的比较次数各为多少? (1 分)

29 1 次, 64 3 次

3. 以下是两个重载函数的原型声明:

```
void overload(int a,int b,double c=5.5);
```

```
void overload(int x,int y);
```

请问此函数重载正确吗? 如果不正确请说明原因。(4 分)

不正确, 因为第一个函数的第三个参数有缺省值, 调用时也可以给两个实参; 而第 2 个函数与第 1 个函数形成重载, 调用时也需要 2 个参数, 此时, 将会产生二义性, 系统无法判断应该调用哪一个函数。

五. 编写程序。(12 分)

编写函数用辗转相除法求整数 a 和 b 的最大公约数。算法为: 将较大的数放在变量 a 中, 较小的数放在 b 中。然后求 a 除以 b 的余数 r。如果 r 为 0, 则除数 b 即为最大公约数; 否则, 将 b 存入 a, 将 r 存入 b, 反复求 a 和 b 的余数, 直到余数为 0。

在主函数中从键盘输入两个整数, 调用上述函数求出它们的最大公约数并输出。