

试卷代号:1008

座位号

中央广播电视大学 2002—2003 学年度第二学期“开放本科”期末考试(开)

## 计算机专业面向对象程序设计试题

2003 年 7 月

|     |   |   |   |   |   |     |
|-----|---|---|---|---|---|-----|
| 题 号 | 一 | 二 | 三 | 四 | 五 | 总 分 |
| 分 数 |   |   |   |   |   |     |

|     |     |
|-----|-----|
| 得 分 | 评卷人 |
|     |     |

### 一、单项选择题(每小题 2 分,共 20 分)

说明:题后给出一组答案,分别标识为 A~D,请从中选择出一个正确的答案,将其标识填入括号中。

1. 下列的常量名中,( )是非法的。
  - A. PI
  - B. 2002YEAR
  - C. Li\_Si
  - D. \_KINE\_2\_5
2. 在程序中,自动变量定义语句正确位置是( )。
  - A. 随便任何位置
  - B. 在所有函数之前
  - C. 该变量使用前的任何位置
  - D. 局部范围内该变量使用前的任何位置
3. 在内存中所存储的含 L 个字符的一个字符串常量是( )。
  - A. 一串相应的 ASCII 码值,占用 L 个字节
  - B. 一串相应的 ASCII 码值,占用 L+1 个字节
  - C. 一组相应的正整数,占用 L 个字节
  - D. 一组相应的整数,占用 2L 个字节

4. C++中整数可用十进制、八进制和十六进制三种方式来表示,但负整数( )。
- A. 不能用十六进制方式来表示,可用十进制和八进制方式来表示
  - B. 不能用八进制方式来表示,可用十进制和十六进制方式来表示
  - C. 既不能用八进制也不能用十六进制方式来表示
  - D. 不能表示
5. C++中的实数可以写成不同表示形式,下列表示形式中( )是正确的。
- A. 1.234E5
  - B.  $1.234 \times 10^6$
  - C. E5
  - D. 3E4.5
6. 函数之间的调用是( )。
- A. 高层的函数调用低层的函数
  - B. 低层的函数不能调用高层的函数
  - C. 具有层次关系,不能随意进行相互调用
  - D. 没有层次关系,函数之间完全可以随意进行相互调用
7. 被非静态成员函数访问的类的数据成员( )。
- A. 可以是非静态数据成员也可以是静态数据成员
  - B. 不可能是类的静态数据成员
  - C. 只能是类的非静态数据成员
  - D. 只能是类的静态数据成员
8. 当将一个类 A 或函数 f()说明为另一个类 B 的友元后,类 A 或函数 f()就可以直接访问类 B 的成员( )。
- A. 只能是公有成员
  - B. 只能是保护成员
  - C. 是除私有成员之外的任何成员
  - D. 是任何成员
9. 从一个基类派生出的各个类的对象之间( )。
- A. 共享所有数据成员,还包含了由同一个基类生成的对象
  - B. 共享部分数据成员,并包含了由同一个基类生成的对象
  - C. 不共享任何数据成员,但包含了由同一个基类生成的对象
  - D. 共享部分数据成员
10. 在定义一个派生类时,若不显式说明是哪种继承方式,那么就默认为( )方式。
- A. 私有继承
  - B. 非私有继承
  - C. 保护继承
  - D. 公有继承

|     |     |
|-----|-----|
| 得 分 | 评卷人 |
|     |     |

二、填空题(每小题 2 分,共 20 分)

说明:下列各题有 1-2 处空项号,请在表格中对应的题号——空项号右边依次填写应有的内容。

| 题号—空项号 | 应有的内容 |
|--------|-------|
| 1—A,B  |       |
| 2—C    |       |
| 3—D,E  |       |
| 4—F,G  |       |
| 5—H    |       |
| 6—I    |       |
| 7—J    |       |
| 8—K    |       |
| 9—L    |       |
| 10—M   |       |

1. 表达式是由  A  按照一定语法规则组成的符号序列,表达式加上  B  构成表达式语句。
2. 大多数表达式都具有值,它是将操作符施加于操作数的计算结果。若表达式是一个  C  的函数调用,则它没有值。
3. continue 语句的作用是结束  D ,跳转到  E  处,继续执行下一次循环。
4. 在函数定义中的参数称为  F ,在调用函数的地方传递给被调函数的参数称为  G 。
5. 如果定义一个函数的返回类型为 void,就表示该函数  H 。
6. 定义在  I  之外的变量或常量为全局量,它可以在其后的所有地方使用。
7. 当函数的参数是指针时,它的实在参数的值必须是一个  J 。
8. 函数返回值类型带有 const 标记,表示不能将函数调用表达式作为  K  使用。
9. C++ 编译器处理内联函数比较特殊;在遇到调用内联函数的地方会用  L  来替换函数的调用。
10. 假定一个变量定义语句为 "int x;" 则定义一个指针变量 p 并初始化为 x 的地址的语句为  M 。

|    |     |
|----|-----|
| 得分 | 评卷人 |
|    |     |

三、程序填充题(每个空项 2 分,共 16 分)

说明:下面程序的每行前面加有行号,不完整部分有空项号,按程序功能要求,在下列表格中对应的行——空项号右边填充程序中的应有的内容。

| 行—空项号 | 应有的内容 |
|-------|-------|
| 1—A   |       |
| 3—B   |       |
| 10—C  |       |
| 13—D  |       |
| 15—E  |       |
| 16—F  |       |
| 23—G  |       |
| 24—H  |       |

// \_\_\_\_\_

//从键盘读入 80 名职工年龄,并记录到一个数组中。

//然后,计算职工年龄平均值并调用函数 top()计算最高年龄值,并输出结果。

// \_\_\_\_\_

1. \_\_\_\_\_ A \_\_\_\_\_
2. int top(int[],int);
3. \_\_\_\_\_ B \_\_\_\_\_ employeeNum=80;
4. \_\_\_\_\_
5. void main()
6. {
7. int employeeAge[employeeNum];
8. int i;
9. int top\_Age,sum\_Age=0;
10. for( \_\_\_\_\_ C \_\_\_\_\_ )
11. {
12. cin>>employeeAge[i];

```

13.     sum_Age _____ D _____ ;
14. }
15. cout<<"The average Age is"<<_____ E _____<<endl;
16. _____ F _____ ;
17. cout<<"The top of Age:"<<top_Age<<endl;
18. }
19.
20. int top(int iArray[],int len)
21. {
22.     int iTop=iArray[0];
23.     for(_____ G _____)
24.         _____ H _____ ;
25.     return iTop;
26. }

```

|    |     |
|----|-----|
| 得分 | 评卷人 |
|    |     |

#### 四、编程题(共 20 分)

说明:正弦函数  $\cos x$  的近似值计算式如下:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + \frac{(-1)^n x^{2n}}{(2n)!}$$

其中,  $x$  为弧度,取值范围:  $-\infty < x < \infty$ ,  $n$  为大于等于 0 的整数值。

根据各个小题程序功能要求,写出函数的完整定义。

1. (本小题 10 分)利用上述近似值计算式,实现余弦函数  $\cos \alpha$  的近似值计算,其中,  $\alpha$  为角度数,取值范围:  $[0^\circ, 90^\circ]$ ,要求误差小于给定值  $\Delta = 0.000001$ 。

```
double con_alpha_90(double dAlpha)
```

```
{//在下面写上程序的实现
```

```
    const double Delta=0.000001;
```

```
    const double PI=3.1415926;
```

```

double x, con_d, con_alpha;;
int n=0;
x=dAlpha * PI/180;      //角度单位转化成以弧度为单位
con_alpha=0; //con_alpha 用于保存累加和
con_d=1; //con_d 用于保存当前累加项的值
//请向下编写剩余代码,完善这个算法,它为一个 do 或 while 循环,以及一条 return 语
句。
}

```

2. (本小题 10 分)根据下面所给的函数原型用递归算法实现上一小题余弦函数  $\cos\alpha$  的近似值计算。其中  $n$  的值为需要累加的项数减 1,  $dAlpha$  的值为一个角度数。

```
double con_alpha(int n, double dAlpha);
```

提示:可使用递归表示,即:

当  $n=0$  时,取值为 1;

当  $n>0$  时,取如下递归表达式:

$$\cos\_alpha(n) = \cos\_alpha(n-1) + \delta(n)$$

其中:

$$\delta(n) = \frac{(-1)^n x^{2n}}{2n!}$$

$\delta(n)$  也可用递归表示,即:

$$\delta(n) = \delta(n-1) * \frac{(-1)^n x^2}{2n(2n+1)}$$

其中递归终止条件为  $\delta(0)$  等于 1

```
double con_d(int n ,double x)      //递归算法实现增量函数的绝对值
```

```
{//在下面写上程序的实现
```

```
}
```

```
double con_alpha(int n ,double dAlpha) //递归算法计算余弦函数展开式中前 n+1 项
```

之和

```
{//在下面写上程序的实现
```

```
}
```

|     |     |
|-----|-----|
| 得 分 | 评卷人 |
|     |     |

五、理解问答题(本题共 24 分)

说明:读懂各个小题程序,为了便于提问,程序的每行前面加有行号,请对所提的问题作出准确的相应解答。

(1)(本小题 10 分)读懂如下程序:

```

1. #include<iostream. h>
2. double SectorArea()
3. {
4.     const float PI=3. 14;
5.     double * pdr=new double(6);
6.     float * pfa=new float[3];
7.     if(! pdr||! pfa)
8.     {
9.         cout<<"Error memory allocation!"<<endl;
10.        return 1;
11.    }
12.    double& dRad= * pdr;
13.    float& fAng=pfa[0];
14.    cout<<"the sector start is:";
15.    cin>>pfa[1];
16.    cout<<"the sector end is:";
17.    cin>>pfa[2];
18.    if(pfa[2]>=pfa[1])fAng=pfa[2]-pfa[1];
19.    else fAng=pfa[1]-pfa[2];
20.    cout<<"the area of sector is"<<PI * dRad * dRad * fAng/360<<endl;
21.    delete &dRad;
22.    delete[]pfa;
23.    return 0;
24. }
```

```

25. void main()
26. {
27.     if(SectorArea())
28.         cout<<"The program failed!"<<endl;
29.     else
30.         cout<<"The program succeeded!"<<endl;
31. }

```

请写出下列问题答案：

①第 5 行起何作用？

答：

②第 7—11 行可否省去？并说明原因。

答：

③第 21—23 行起何作用？

答：

④此程序功能是什么？若对"the sector start is;"和"the sector end is;"提示回答分别为 40.0 和 70.0, 写出执行输出结果。

答：

(2)(本小题 14 分)读懂如下程序：

```

/ ***** stack. h ***** /
class Stack {
    public:
        Stack(int size=20);
        ~Stack();
        void Push(int newElem);
        int Pop();
        int Depth(){return top;}
        bool Empty(){return top==0;}
        bool Full(){return top==stackSize;}
        void Print()const;

```



```

private:
    const int stackSize;
    int * elem;
    int top;
};
/ ***** stack. cpp ***** /
#include<iostream. h>
#include"stack. h"

Stack::Stack(int size):stackSize(size)
{
    elem=new int[size];
    top=0;
}
Stack::~~Stack()
{
    delete[]elem;
}
void Stack::Push(int newElem)
{
    if(Full())
    {
        cout<<"nStack is overfilled!!!"<<endl;
        return;
    }
    elem[top]=newElem;
    top++;
}

int Stack::Pop()
{
    if(Empty())

```

```

    {
        cout<<"\nStack is empty!!! Top==";
        return 0;
    }
    int ret=elem[top-1];
    top--;
    return ret;
}

void Stack::Print()const
{
    if(! top)cout<<"Stack is empty!";
    else cout<<"stack: ";
    for(int i=0;i<=top-1;i++) cout<<elem[i]<<" ";
    cout<<endl;
}

```

```

/ ***** main. cpp *****/
#include<iostream. h>
#include"stack. h"
void main()
{
    Stack s(10);
    int i;
    cout<<"Pushing 10 elements: ";
    for(i=0;i<10;i++)
    {
        cout<<i<<" ";
        s. Push(i);
    }
    cout<<"\nStack length="<<s. Depth()<<endl;
}

```

```

cout << "\nstack full?" << s.Full() << endl;
s.Print();

cout << "\nPopping elements:";
for(i=0; i<12; i++) cout << s.POp() << " ";
cout << endl;

cout << "\nPushing 6 elements:" << endl;
for(i=6; i>0; i--) s.Push(i);
cout << "Stack length=" << s.Depth() << endl;
s.Print();

cout << "\npushing 5 elements:";
for(i=0; i<5; i++)
{
    cout << i << " ";
    s.Push(i);
}

cout << "\nstack full?" << s.Full() << endl;
s.Print();
}

```

请写出此程序执行所显示的输出结果。

试卷代号:1008

中央广播电视大学 2002—2003 学年度第二学期“开放本科”期末考试(开)

## 计算机专业面向对象程序设计

### 试题答案及评分标准

(供参考)

2003 年 7 月

#### 一、单项选择题(每小题 2 分,共 20 分)

1. B            2. D            3. B            4. C            5. A  
6. D            7. A            8. D            9. C            10. A

评分标准:共 20 分,每答对一小题得 2 分。

#### 二、填空题(每小题 2 分,共 20 分)

| 题号—空项号 | 应有的内容       |      |
|--------|-------------|------|
| 1—A,B  | 操作符和操作数     | 分号   |
| 2—C    | 没有返回值       |      |
| 3—D,E  | 当前一次循环      | 循环开始 |
| 4—F,G  | 形式参数        | 实在参数 |
| 5—H    | 没有返回值       |      |
| 6—I    | 任何函数(或所有函数) |      |
| 7—J    | 地址(或内存地址)   |      |
| 8—K    | 左值          |      |
| 9—L    | 函数体中的代码     |      |
| 10—M   | int * p=&x; |      |

评分标准:共 20 分,每题 2 分,填错一空则不得分。

#### 三、程序填空题标准答案

| 行—空项号 | 应有的内容                                  |
|-------|--|
| 1—A   | #include<iostream. h>                  |
| 3—B   | const int                              |
| 10—C  | i=0; i<employeeNum; i++                |
| 13—D  | + = employeeAge[i];                    |
| 15—E  | sum_Age/employeeNum                    |
| 16—F  | top_Age=top(employeeAge, employeeNum); |
| 23—G  | int i=1; i<len ; i++                   |
| 24—H  | if(iTop<iArray[i]) iTop=iArray[i]      |

评分标准:共 16 分,每填对一空项得 2 分。

#### 四、编程题标准答案

##### 1. [参考答案解法一]

```
do{  
    if(n%2==0) con_alpha+=con_d;  
    else con_alpha-=con_d;  
    n++;  
    con_d*=x*x/(2*n*(2*n-1));  
}while(con_d>=Delta);    //逐步逼近,直到一个累加项的值小于给定允许误差值
```

为止

```
return son_alpha;
```

##### [参考答案解法二]

```
while(con_d>=Delta){  
    if(n%2==0)con_alpha +=con_d;  
    else con_alpha-=con_d;  
    n++;  
    con_d*=x*x/(2*n*(2*n-1));  
}  
return con_alpha;
```

评分标准:本小题共 10 分,具体评分要点如下

①正确的循环条件和循环增量占 2 分。

②循环体占 6 分。

③返回结果占 2 分。

##### 2. [参考答案]

```
double con_d(int n double x)    //递归算法实现增量函数  
{  
    if(n==0) return 1;  
    else return con_d(n-1,x)*x*x/(2*n*(2*n-1));  
}
```

```

double con_alpha(int n ,double dAlpha) //递归算法实现余弦函数
{
    const double PI=3.1415926;
    double x,d;
    x=dAlpha * PI/180; //角度单位转化成以弧度为单位
    //检查参数是否合理,若不合理,给出提示信息,并立即退出,返回-1。
    if(n<0||x>PI/2)
    {
        cout<<"Error!"<<endl;
        return-1;
    }

    //递归算法计算
    if(n==0) return 1;
    else
    {
        d=con_d(n,x);
        if(n%2==0)return con_alpha(n-1,dAlpha)+d;
        else return con_alpha(n-1,dAlpha)-d;
    }
}

```

评分标准:本小题共 10 分,其中 con\_d 算法占 5 分,con\_alpha 算法占 5 分。请酌情给分。

## 五、理解问答题(共 24 分)

1. 10 分,标准答案及评分标准:

①答:申请分配一个与 double 类型数据相应的内存空间,并赋予初始值为 6。(2 分)

②答:不能省,若系统已无内存空间可分配,仍继续执行程序可能发生破坏性后果。

(2 分)

③答:分别释放相应于前面申请两类数据获准所占用动态内存空间。(2 分)

④答:根据输入的初始和终止角计算扇形区的面积。(2 分)

执行输出结果为:(2分)

the sector start is:40.0

the sector end is:70.0

The area of sector is 9.42

The program succeeded!

2. 14分,标准答案及评分标准如下:

输出结果为:

Pushing 10 elements 0 1 2 3 4 5 6 7 8 9

Stack length=10

stack full? 1

Stack:0 1 2 3 4 5 6 7 8 9

Poping elements:9 8 7 6 5 4 3 2 1 0

Stack is empty!!! Top==0

Stack is empty!!! Top==0

Pushing 6 elements:

Stack length=6

Stack:6 5 4 3 2 1

Pushing 5 elements:0 1 2 3 4

Stack is overfilled!!!

stack full? 1

Stack:6 5 4 3 2 1 0 1 2 3

评分标准:共14分,输出共分6段,每段正确得2分,全对得14分。