

座位号：

杭州电子科技大学学生考试卷（ A ）卷

考试课程	C++面向对象程序设计(甲)		考试日期	2017 年 6 月 日		成 绩	
课程号	A0507070	教师号		任课教师姓名			
考生姓名		学号 (8 位)		年 级		专 业	

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、单项选择题（每题 2 分，总计 20 分）

题 号	1	2	3	4	5
答 案	C	D	A	D	A
题 号	6	7	8	9	10
答 案	C	C	C	B	D

二、程序填空题（每空 2 分，总计 20 分）

- (1) _____date d,int day_____
- (2) _____day>dys[d.mo-1]_____
- (3) _____temp.n=n+s.n_____
- (4) _____s3.disp()_____
- (5) _____A(i+10)_____或 A（6*i）_____
- (6) _____dispa()_____
- (7) _____【virtual】 void f()_____
- (8) _____f()_____
- (9) ____bool operator()(int x) _____
- (10) ____count_if(a.begin(),a.end(),Pass())_____

三、阅读程序题（每题 6 分，总计 30 分）

1. 阅读该程序，给出程序的输出结果。

n=10,k=3
n=20,k=3
n=30,k=3

2. 阅读该程序，给出程序的输出结果。

n=9

3. 阅读该程序，给出程序的输出结果。

5
4.12
China

4. 阅读该程序，给出程序的输出结果。

anihCevol

5. 阅读该程序，给出程序的输出结果。

8/3 = 2
【6/0=】 6 is divided by zero!
That is ok.

座位号:

四、编程题（每题 15 分，总计 30 分）

1. 按照要求，编写程序。

评分标准：要求采用抽象类和类继承，若不采用则扣分。

```
#include <iostream>
using namespace std;

//雇员类
class Employee {
protected :
    int _id;
    double _salary;
    string _name;
public :
    Employee (int id, string name)
    : _id(id), _name (name)
    {}
    virtual ~Employee () {}
    virtual double    CalcSalary () = 0;
};

//经理类
class Manager : public Employee {
public :
    Manager (int id,  string name) : Employee (id,name) {}
    virtual double    CalcSalary () {
        _salary = 8000;
        return _salary;
    };
};

//兼职技术员类
class Technician : public Employee {
    int _workHour;
public :
    Technician (int id, string name) : Employee (id,name) {}
    void    SetWorkHour (int workHour) {
        _workHour = workHour;
    };
};
```

```
    }
    virtual double    CalcSalary () {
        _salary = 100 * _workHour;
        return _salary;
    };
};

//销售经理类
class Salesman : public Employee {
    double _salesMount;
public :
    Salesman (int id, string name) : Employee (id,name) {}
    void    SetSalesMount (double salesMount) {
        _salesMount = salesMount;
    }
    virtual double    CalcSalary () {
        _salary = 0.04 * _salesMount;
        return _salary;
    };
};

class SalesManager : public Employee {
    double _salesTotalMount;
public :
    SalesManager (int id, string name) : Employee (id,name) {}
    void    SetSalesTotalMount (double salesTotalMount) {
        _salesTotalMount = salesTotalMount;
    }
    virtual double    CalcSalary () {
        _salary = 5000 +0.005 * _salesTotalMount;
        return _salary;
    };
};

int main()
{
    Manager zhang (1000, "张三");
    Technician li (1000, "李四");
    Salesman wang (1000, "王五");
    SalesManager zhao (1000, "赵六");
};
```

```
li.SetWorkHour(50);  
wang.SetSalesMount(175000);  
zhao.SetSalesTotalMount(1500000);  
Employee *pEmployeeA [] = {&zhang, &li, &wang, &zhao};  
  
for (int i = 0; i <4; i++) {  
    cout << pEmployeeA [i] ->CalcSalary() << endl;  
}  
}
```

座位号：

<div>2. 按照要求，编写程序。</div> <div>评分标准：不可存在内存泄露。</div> <div><pre>#include <iostream> #include <vector> #include <cstdlib> using namespace std; class Matrix { public: Matrix (int row, int col); //构造 row x col 大小矩阵 Matrix operator + (const Matrix &rhs) const; //相同大小矩阵相加 Matrix operator - (const Matrix &rhs) const; //相同大小矩阵相减 void Display () const; //显示矩阵 //矩阵元素，可作为右值 const int & at (int row, int col) const { return _datas [row][col]; } //矩阵元素，可作为左值 int& at (int row, int col) { return _datas [row][col]; } //矩阵行数 int GetRowCount () const { return _datas.size (); } //矩阵列数 int GetColCount () const { return _datas [0].size (); } private: //存放矩阵元素， 本类没有直接动态分配，无需重载拷贝构造和赋值 vector<vector<int>> _datas; }; Matrix::Matrix (int row, int col) { for (int i = 0; i < row; i++) { vector<int> dataRow (col);</pre></div>	<div><pre> _datas.push_back (dataRow); } } Matrix Matrix::operator + (const Matrix &rhs) const { Matrix result (this->GetRowCount(), this->GetColCount()); for (unsigned int i = 0; i < _datas.size (); i++) { for (unsigned int j = 0; j < _datas [i].size (); j++) { result.at(i,j) = this->at(i, j) + rhs.at (i, j); } } return result; } Matrix Matrix::operator - (const Matrix &rhs) const { Matrix result (this->GetRowCount(), this->GetColCount()); for (unsigned int i = 0; i < _datas.size (); i++) { for (unsigned int j = 0; j < _datas [i].size (); j++) { result.at(i,j) = this->at(i, j) - rhs.at (i, j); } } return result; } void Matrix::Display () const { for (unsigned int i = 0; i < _datas.size (); i++) { for (unsigned int j = 0; j < _datas [i].size (); j++) { cout.width (5); cout << at (i,j); } cout << endl; } cout << endl; }</pre></div>
--	--

```
int main()
{
    Matrix  A (5, 6), B (5, 6);
    int     i, j;
    for (i = 0; i < A.GetRowCount(); i++)
        for (j = 0; j < A.GetColCount(); j++)
            A.at(i, j) = rand () % 100 ;
    for (i = 0; i < A.GetRowCount(); i++)
        for (j = 0; j < A.GetColCount(); j++)
            B.at(i, j ) = rand () % 100;

    A.Display();
    B.Display();

    Matrix C = A + B;
    C.Display();

    Matrix D = A - B;
    D.Display();

    return 0;
}
```

杭州电子科技大学学生考试卷 (A) 卷

考试课程	C++面向对象程序设计(甲)		考试日期	2017 年 6 月 日		成 绩	
课程号	A0507070	教师号		任课教师姓名			
考生姓名		学号 (8 位)		年 级		专 业	

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、单项选择题（每题 2 分，总计 20 分）

1. 若定义语句为 `const char *ptr; , ptr` 应该是_____。
A. 指向字符变量的指针 B. 指向字符的常量指针
C. 指向字符串常量的指针 D. 指向字符串的常量指针
2. 对定义重载函数的下列要求中, _____是错误的。
A. 要求参数的个数不同
B. 要求参数中至少有一个类型不同
C. 要求参数个数相同时, 参数类型不同
D. 要求函数的返回值不同
3. 有以下类的说明, 请指出 A、B、C、D 四个语句错误的是_____。

```
class CSample
{    int a=2.5;           //A
    CSample( );           //B
public:
    CSample(int val);      //C
    ~CSample( );          //D
};
```

4. 一个类的友元函数能够访问该类的_____。
A. 只有私有成员
B. 只有保护成员
C. 只有公有成员
D. 所有成员
5. 下列运算符中，_____运算符在 C++中不能重载。
A. ? : B. + C. - D. <=
6. 以下对模板的说明，正确的是_____。
A. `template<T>` B. `template<class T1,T2>`
C. `template<class T1,class T2>` D. `template<class T1;class T2>`
7. 关于多继承二义性的描述中，_____是错误的。
A. 一个派生类的两个基类中都有某个同名成员，在派生类中对这个成员的访问可能出现

二义性

- B. 解决二义性的最常用的方法是对成员名的限定
- C. 基类和派生类中出现同名函数，也存在二义性问题
- D. 一个派生类是从两个基类派生来的，而这两个基类又有一个共同的基类，对该基类成员进行访问时，也可能出现二义性

8. 以下基类中的成员函数，哪个表示纯虚函数_____。
- A. `virtual void vf(int);` B. `void vf(int)=0;`
C. `virtual void vf()=0;` D. `virtual void vf(int){ }`
9. 在 C++ 中，打开一个文件就是将这个文件与一个_____建立关联；关闭一个文件就取消这种关联。
- A. 类 B. 流 C. 对象 D. 结构

- 10 现有语句 `int iarray[] = { 0, 1, 2, 3, 4, 5, 6, 6, 6, 7, 8 };`
`vector<int> ivector(iarray, iarray + sizeof(iarray) / sizeof(int));`
 请问要找出 `ivector` 之中小于 7 的元素个数, 采用下面哪个算法_____。
 A. `find_if` B. `count` C. `sort` D. `count_if`

二、程序填空题（每空 2 分，总计 20 分）

1. 请在下面程序的横线处填上适当内容,以使程序完整,并使运行结果为: 2/10/2004

```
#include <iostream.h>

static int dys[]={31,28,31,30,31,30,31,31,30,31,30,31};

class date
{
    int mo,da,yr;

public:
    date(int m,int d,int y) {mo=m;da=d;yr=y;}
    date() {}
    void disp() {cout<<mo<<"/"<<da<<"/"<<yr<<endl;}
    void leap(int a){
        if(a%4==0&&a%100!=0||a%400==0)
            dys[1]=29;
        else dys[1]=28;}
    friend date operator+(_____(1)_____)
    {
        day+=d.da;
        d.leap(d.yr);
        while(_____(2)_____)
        {
            d.leap(d.yr);
```

```
        day-=dys[d.mo-1];
        if(++d.mo==13)
        {   d.mo=1;
            d.yr++;
        }
    }
    d.da=day;
    return d;
}
```

```
};
int main()
{   date d1(2,10,2003),d2;
    int n;
    cin>>n; //从键盘输入 365
    while(n<0){cin>>n;}
    d2=d1+n;
    d2.disp();
}
```

2. 请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为： n=30

```
#include <iostream.h>
template <class T>
class Sample
{   T n;
public:
    Sample(){}
    Sample(T i){n=i;}
    Sample<T> operator+(const Sample<T> &);
    void disp(){cout<<"n="<<n<<endl;}
};
template <class T>
Sample<T> Sample<T>::operator+(const Sample<T> &s)
{   Sample<T> temp;
    _____(3)_____;
    return temp;
}
int main()
{   Sample<int> s1(10),s2(20),s3;
    s3=s1+s2;
    _____(4)_____;
```

```
}
```

3. 请在下面程序的横线处填上适当字句，以使程序完整,并使程序的输出为

```
A Constructor2
B Constructor2
x1=12
x2=2
```

```
#include <iostream.h>
class A
{
public:
    A(){cout<<"A Constructor1"<<endl;}
    A(int i){x1=i;cout<<"A Constructor2"<<endl;};
    void dispa(){cout<<"x1="<<x1<<endl;}
private:
    int x1;
};
class B:public A
{
public:
    B(){cout<<"B Constructor1"<<endl;}
    B(int i): _____(5)_____ {x2=i;cout<<"B Constructor2"<<endl;}
    void dispb()
    {   _____(6)_____;
        cout<<"x2="<<x2<<endl;
    }
private:
    int x2;
};
int main()
{   B b(2);
    b.dispb();
}
```

4. 请在下面程序的横线处填上适当字句，以使程序完整,并使程序的输出为
构造 base 对象
调用 base::f()
构造 derive 对象
调用 derive::f()

```
#include <iostream.h>
class base
{
public:
    base()
    {   cout<<"构造 base 对象"<<endl;
        f();
    }
    _____(7)_____ {cout<<"调用 base::f()"<<endl;}
};
class derive:public base
{
public:
    derive()
    {   cout<<"构造 derive 对象"<<endl;
        _____(8)_____;
    }
    void f(){cout<<"调用 derive::f()"<<endl;}
};
int main()
{
    derive d;
}
```

5. 下面程序用 STL 的条件计数算法和自定义的函数对象对一个存放在整数向量类对象中的学生成绩进行统计及格人数并显示结果。请在下面程序的横线处填上适当字句，以使程序完整。

```
#include<iostream>
#include<vector>
#include<algorithm>
#include<functional>
using namespace std;
class Pass
{
public:
    _____(9)_____
    {
        return x>=60;
    }
};
```

```
int main()
{
    vector<int> a;
    int count;
    a.push_back(78);
    a.push_back(92);
    a.push_back(52);
    count = _____(10)_____;
    cout<<"count="<<count<<endl;
}
```

三、程序阅读题（每题 6 分，总计 30 分）

```
1. #include <iostream.h>
class CSample
{
private:
    int n;
    static int k;
public:
    CSample(int i){n=i;k++;}
    void disp();
};
void CSample::disp()
{   cout<<"n="<<n<<"k="<<k<<endl;}
int CSample::k=0;
int main()
{   CSample a(10),b(20),c(30);
    a.disp();
    b.disp();
    c.disp();
}
```

```
2. #include <iostream.h>
class Sample
{   int n;
public:
    Sample(){}
    Sample(int m){n=m;}
    Sample operator--(int)
```


<pre> { Sample old=*this; n--; return old; } void disp(){cout<<"n="<<n<<endl;} }; int main() { Sample s(10); s--; s.disp(); } 3. #include <iostream.h> #include <string.h> template <class T> T max(T x, T y) { return (x > y ? x : y); } char *max(char *x, char *y) { if (strcmp(x, y) >= 0) return x; else return y; } int main() { cout << max(2, 5) << endl; cout << max(3.14,4.12) << endl; cout << max("China", "Beijing") << endl; } 4. #include <stack> #include <iostream> #include <string> using namespace std; int main() { stack<char> s; string str;</pre>	<pre> cin >> str; //从键盘输入字符串 LoveChina for (string::iterator iter = str.begin(); iter != str.end(); ++iter) s.push(*iter); while (!s.empty()) { cout << s.top(); s.pop(); } cout << endl; } 5. #include <iostream> using namespace std; int divide(int x, int y) { if (y == 0) throw x; return x / y; } int main() { try { cout << "8/3 = " << divide(8, 3) << endl; cout << "6/0 = " << divide(6, 0) << endl; cout << "9/2 = " << divide(9, 2) << endl; } catch (int e) { cout << e << " is divided by zero!" << endl; } cout << "That is ok." << endl; } 四、编程题（每题 15 分，总计 30 分） 1. 编程实现小型公司的工资管理。该公司主要有 4 类人员：经理（manager）、兼职技术人员（technician）、销售员（salesman）和销售经理（salesmanager）。要求存储这些人员的编号、姓名和月工资，计算月工资并显示全部信息。月工资计算办法是：经理拿固定月薪 8000 元，兼职技术人员按每小时 100 元领取月薪，销售员按当月销售额的 4%提成，销售经理既拿固定月工资也领取销售提成，固定月工资为 5000 元，销售提成为所管辖部门当月销售总额的千分之五。兼职技术人员一个月工作小时数、销售员一个月销售额、销售经理所管辖部门一个月销售总额由各个类的成员函数完成设置。（要求用抽象类和类继承） 2. 定义一个矩阵类（Matrix），设计相关的构造函数、析构函数等，采用运算符重载方式实现矩阵的加、减运算。编写 main 函数，对以上所有功能进行测试。注意：按照实际情况大小使用内存空间，矩阵的加、减运算不要造成内存浪费。</pre>
---	--