

杭州电子科技大学学生考试卷（A）卷

考试课程	面向对象程序设计		考试日期	06 年 6 月     日			成绩		
课 程 号	B1002100	教 师 号		任课教师姓名			楼永坚		
考生姓名		学号（8 位）		年级	05	专业	050511/2/3	座位号	

一、判断题（15 分）（对的打√，错的打×）

- 友元函数用于允许一个函数访问不相关类的私有部分。
- 构造函数可以被继承。
- 动态绑定的多态性是通过虚函数实现的。
- 在 c++中，传引用调用等同于传地址调用。
- 重载函数必须有不同的参数列表。
- 可以用 delete 释放不是用 new 运算符分配的内存。
- 类成员的默认访问模式是 private。
- 在类 Time 中的析构造函数可以声明为：void ~Time(int);
- const 对象必须初始化。
- 在 c++中，只能重载已有的运算符。

二 、选择题（20 分）

- 关键字 \_\_\_\_\_ 说明对象或变量初始化后不会被修改。  
a. static    b. public    c. const    d. inline
- 如果调用带有默认参数的函数时缺少一个参数，则\_\_\_\_\_参数就作为这一参数。  
a. 第一个    b. 最后一个    c. 中间一个    d. 以上都不是
- 成员函数可声明为静态的，条件是它不访问 \_\_\_\_\_类成员。  
a. 静态    b. 常数    c. 非静态    d. 公共
- 内联函数执行起来比标准函数\_\_\_\_\_。  
a. 更慢    b. 更快    c. 次数更多    d. 以上都不是
- 默认参数的值由\_\_\_\_\_提供。  
a. 该函数    b. 调用程序    c. 上述二者    d. 以上都不是
- 在 C++ 中，混合类型表达式\_\_\_\_\_。  
a. 允许存在    b. 为一错误    c. 从 int 到 float    d. 从 float 到 int
- 表达式 long(intVar) 也可表示为 \_\_\_\_\_。  
a. intvar = long;    b. intVar(long)    c. (long)intVar    d. 以上都不是
- 静态数据成员的生存期\_\_\_\_\_。  
a. 与整个程序相同    b. 不长于类的生存期

- c. 取决于创建的对象数    d. 以上都不是

9. 要让一个类中的所有对象具有共同的数据，请使用\_\_\_\_\_。

- a. 常数成员变量    b. 私有数据成员  
c. 静态数据成员    d. 以上都是

10. 设置虚基类的目的是：

- a.简化程序    b.消除二义性    c.提高运行效率    d.减少目标代码

三 、指出下列程序片段中的错误标号，写出正确语句或解释错在何处。（20 分）

1) ①int index=675;

②const int \*ptr=&index;

③int \*const ntptr=&index;

④\*ptr=555;

⑤\*ntptr=666;

⑥int another=8;

⑦ptr=&another;

⑧ntptr=&another;

2) ①int arrp;

②arrp=new int[15];

③delete arrp;

3) 下面程序为什么会编译错误，并改正错误（提出解决办法）。

```
class window
{
    protected:
    int basedata;
};
class border: public window
{ };
class menu: public window
{ };
class border_and_menu: public border, public menu
{
    public:
    int show()
    { return basedata;
    }
```

4) 改正下面程序段中的错误，写出整个正确的程序段

template<T>

```
void print(T *a)
{ cout<<a<<'\\n';}
void main( )
{
    const int x=0;
    cout<<y<<'\\n';
    int y;
    x=5;
    int* p
    p=&y;
    print(p);
    return 0;
}
```

四 、写出下面程序的执行结果：（15 分）

```
1) #include <iostream>
using namespace std;
class A
{friend double count(A&);
public:
A(double t, double r):total(t),rate(r){}
private:
    double total;
double rate;
};
double count(A& a)
{
a.total+=a.rate*a.total;
return a.total;
}
int main(void)
{
    A x(80,0.5),y(100,0.2);
    cout<<count(x)<<','<<count(y)<<'\\n';
    cout<<count(x)<<'\\n';
    return 0;
}
```

执行结果：

```
2)
#include<iostream>
using namespace std;
class Count{
    private:
        static int counter;
        int obj_id;
    public:
        Count(); //constructor
        static void display_total(); //static function
        void display();
        ~Count(); //destructor
};
int Count::counter; //definition of static data member
Count::Count() //constructor
{
    counter++;
    obj_id = counter;
}
Count::~~Count() //destructor
{
    counter--;
    cout<<"Object number "<<obj_id<<" being destroyed\\n";
}
void Count::display_total() //static function
{
    cout <<"Number of objects created is = "<<counter<<endl;
}
void Count::display()
{
    cout << "Object ID is "<<obj_id<<endl;
}
int main(void)
{
    Count a1;
    Count::display_total();
    Count a2, a3,a4;
```

<pre>Count::display_total(); a2.display(); a4.display(); return 0; } 3) #include &lt;iostream&gt; using namespace std; class BASE {     char c; public:     BASE(char n):c(n){}     virtual ~BASE(){cout&lt;&lt;c;} }; class DERIVED:public BASE{     char c; public:     DERIVED(char n):BASE(n+1),c(n){}     ~DERIVED(){cout&lt;&lt;c;} }; int main(void) {     DERIVED('X');     return 0; }</pre> <p>五、 程序填空：（10 分）</p> <pre>#include &lt;iostream&gt; using namespace std; class A {     _____ (1) _____     char name[80]; public:     A(  ____ (2)  ____  ) {____ (3)  ____ } }; class B_____ (4) _____ { public:</pre>	<pre>    B(const char*n)_____ (5) _____{}     void PrintName( ) {cout&lt;&lt;"name:"&lt;&lt;name&lt;&lt;endl;}; }; void main( ) {     B b1("Ling Li");     b1.PrintName( ) ; } //    执行结果：    name: Ling Li</pre> <p>六、编程题（20 分）</p> <p>1. 编写程序：定义抽象基类 Shape，由它派生出五个派生类：Circle（圆形）、Square（正方形）、Rectangle（长方形）、Trapezoid（梯形）和 Triangle（三角形），用虚函数分别计算各种图形的面积，并求出它们的和。要求用基类指针数组。使它的每一个元素指向一个派生类的对象。</p> <p>注：主函数中定义如下对象</p> <pre>Circle circle(12.6);     Square square(3.5); Rectangle rectangle(4.5,8.4); Trapezoid trapezoid(2.0,4.5,3.2); Triangle triangle(4.5,8.4);</pre>
--	--

杭州电子科技大学学生考试卷（A）答卷

考试课程	面向对象程序设计		考试日期	06 年 6 月    日			成绩		
课 程 号	B1002100	教 师 号		任课教师姓名			楼永坚		
考生姓名		学号（8 位）		年级	05	专业	050511/2/3	座位号	

一、判断题（15 分）

1)      2)      3)      4)      5)      6)      7)      8)      9)      10)

二、选择题（20 分）

1)      2)      3)      4)      5)      6)      7)      8)      9)      10)

三 、指出下列程序片段中的错误，并解释错在何处。（20 分）

1)

2)

3)

4)

四 、写出下面程序的执行结果：（15 分）

1)

2)

3)

五、程序填空：（10 分）

六、编程题（20 分）

杭州电子科技大学学生考试卷（A）答案

考试课程	面向对象程序设计		考试日期	06 年 6 月    日			成绩		
课 程 号	B1002100	教 师 号		任课教师姓名			楼永坚		
考生姓名		学号（8 位）		年级	05	专业	050511/2/3	座位号	

一、判断题（15 分）

1) ✓    2) ✗    3) ✓    4) ✗    5) ✓    6) ✗    7) ✓    8) ✗    9) ✓    10) ✓

二、选择题（20 分）

1) c    2) b    3) c    4) b    5) c    6) a    7) c    8) a    9) c    10) b

三 、指出下列程序片段中的错误，并解释错在何处。（20 分）

1) ④\*ptr=555;    ptr 是指向整数常量的指针  
⑧ntptr=&another;    ntptr 是常量指针，不能指向别的的变量

2) ①int    arrp;    应改为: int \*arrp;  
③delete arrp;    应改为: delete [ ]arrp;

3) return basedata;// 在 border\_and\_menu 中引用 basedata 时产生二义性，应使用虚基类  
应改为：  
class border:virtual public window  
{ };  
class menu: virtual public window  
{ };

4) 整个正确的程序段（参考）：  
#include <iostream.h>    //加本句  
template<typename T>    //加 typename  
void print(T \*a)  
{    cout<<a<<"\n";}  
void main( )  
{    int y=10;    //y 应先声明后使用，并给初值  
      const int x=0;  
      cout<<y<<"\n";  
      //x=5;    x 为 const，去掉该句

```
int* p;  
p=&y;  
print(p);  
// return 0; main 返回为 void ， 去掉该句  
}
```

四 、写出下面程序的执行结果：（15 分）

1)

```
120,120  
180  
Print out 120
```

2)

```
Number of objects created is = 1  
Number of objects created is = 4  
Object ID is 2  
Object ID is 4  
Object number 4 being destroyed  
Object number 3 being destroyed  
Object number 2 being destroyed  
Object number 1 being destroyed
```

3) XY

五、程序填空：（10 分）

（1）protected:或 public    （2）const char \*n    （3）strcpy(name,n);  
（4）: public A 或: protected A    （5）:A(n)

六、编程题（20 分）

1.

```
#include <iostream>  
using namespace std;  
class Shape  
{public:  
    virtual double area() const =0;  
};  
  
class Circle:public Shape  
{public:  
    Circle(double r):radius(r){  
        virtual double area() const {return 3.14159*radius*radius;};  
    protected:  
        double radius;  
};
```

<pre>class Square:public Shape {public:     Square(double s):side(s){}     virtual double area() const {return side*side;} protected:     double side; };  class Rectangle:public Shape {public:     Rectangle(double w,double h):width(w),height(h){}     virtual double area() const {return width*height;} protected:     double width,height; };  class Trapezoid:public Shape {public:     Trapezoid(double t,double b,double h):top(t),bottom(t),height(h){}     virtual double area() const {return 0.5*(top+bottom)*height;} protected:     double top,bottom,height; };  class Triangle:public Shape {public:     Triangle(double w,double h):width(w),height(h){}     virtual double area() const {return 0.5*width*height;} protected:     double width,height; };  int main() {     Circle circle(12.6);     Square square(3.5);     Rectangle rectangle(4.5,8.4);</pre>	<pre>Trapezoid trapezoid(2.0,4.5,3.2); Triangle triangle(4.5,8.4);     Shape *pt[5]={&amp;circle,&amp;square,&amp;rectangle,&amp;trapezoid,&amp;triangle};     double areas=0.0;     for(int i=0;i&lt;5;i++)         {areas=areas+pt[i]-&gt;area();}     cout&lt;&lt;"totol of all areas="&lt;&lt;areas&lt;&lt;endl;     return 0; }</pre>
--	--