浙江理工大学 2017 —2018 学年第二学期
《面向对象程序设计 A》期末试卷（ A ）卷

本人郑重承诺：本人已阅读并且透彻地理解《浙江理工大学考场规则》，愿意在考试中自觉遵守这些规定，保证按规定的程序和要求参加考试，如有违反，自愿按《浙江理工大学学生违纪处分规定》有关条款接受处理。

承诺人签名：_____ 学号：_____ 班级：_____

一．选择题（每小题 2 分，共 20 分）

1. 面向对象程序设计思想的主要特征中不包括(　　)。

A. 封装性　　B. 多态性　　C. 继承性　　D. 功能分解，逐步求精

2. 已知X类，则当程序执行到语句：X array[3], *p[2];时，调用了(　　)次构造函数。

A. 5　　B.3　　C. 2　　D. 1

3. 关于类的构造函数和析构函数,下面说法中正确的是(　　)。

A. 一个类只能定义一个构造函数，但可以定义多个析构函数

B. 一个类只能定义一个析构函数，但可以定义多个构造函数

C. 构造函数与析构函数同名，只要名字前加了一个求反符号(~)

D. 构造函数可以指定返回类型，而析构函数不能指定任何返回类型，即使是void类型也不可以

4. 下面关于 C++中类的继承与派生的说法错误的是(　　) 。

A．基类的 protected 成员在公有派生类的成员函数中可以直接使用

B．基类的 protected 成员在私有派生类的成员函数中可以直接使用

C．派生类可以访问基类的所有数据成员，调用基类的所有成员函数

D．继承描述类的层次关系，派生类可以具有与基类相同的方法

5. 考虑下面的函数原型声明：void testDefaulParam(int a,int b=7,char z='*');
下面函数调用中，不合法的是(　　)。

A. testDefaulParam(5);　　B. testDefaulParam(5,8);

C. testDefaulParam(5,"#");　　D. testDefaulParam(0,0,'#');

6. 在一个函数中，要求通过函数来实现一种不太复杂的功能，并且要求加快执行速度，选用(　　)。

A. 内联函数　　B. 重载函数　　C. 递归调用　　D. 嵌套调用

7.下列关于静态数据成员的说法，不正确的是(　　)。

A. 类中定义的公用静态数据成员，可以通过类的对象来访问

B. 类中定义的所有静态数据成员，都必须在类外初始化

C. 静态数据成员不是所有对象所共用的

D. 普通的成员函数可以直接访问类中的静态数据成员

8.下面对模板的声明，正确的是(　　)。

A. template<T>　　B. template<class T1, T2>

C. template<class T1, class T2>　　D. template<class T1; class T2>

9. 在派生类中重新定义虚函数时，除了(　　)方面，其他方面都必须与基类中相应的虚函数保持一致。

A．参数个数　B．参数类型　C．函数名称　D．函数体

10. 友元运算符_left>_right 被 C++编译器解释为(　　) 。

A.operator >(_left, _right)　　B. r >(_left, _right)

C._right.operator >(_left)　　D. _left.operator >(_right)

二、改错题, 指出错误语句说明原因并改正之(共 10 分)

要求：程序能够输出 1 2 3。

```
#include<iostream>
using namespace std;
class X{
private:
    int a=0,&b=a;
    const int c=10;
public:
    X(int i,int j=3,int k){a=i;b=j;c=k;}
    static void display( ){cout<<a<<' '<<b<<' '<<c<<endl;}
};
int main( ){
     X x1(1,2,3);
     const int r;
     int i, &d;
     const char *pc1="hello";
     pc1[2]='t';
     x1.display();
     return 0;
}
```

三．阅读程序，写出程序的运行结果(共 20 分)

1. (8 分)

```
#include<iostream>
using namespace std;
class A{
    int x;
public:
    A():x(0){cout<<"constructor A() called..."<<endl;}
    A(int i):x(i){cout<<"X"<<x<<"\tconstructor..."<<endl;}
    ~A(){cout<<"X"<<x<<"\tdestructor..."<<endl;}
};
class B{
    int y;
    A X1,X2[2];
public:
    B(int j):X1(j),y(j){cout<<"B"<<j<<"\tconstructor..."<<endl;}
    ~B(){cout<<"B"<<y<<"\tdestructor..."<<endl;}
};
int main(){
    B B1(3);
    return 0;
}
```

2．(7 分)

```
#include <iostream>
using namespace std;
class A {
    int a;
public:
    A(int i){ cout<<"Constructing A "<<i<<endl; }
};

class B {
public:
```

```
    B(){ cout<<"Constructing B "<<endl;}
};

class B1: public B ,virtual public A{
public:
    B1(int i):A(i){ cout<<"Constructing B1 "<<i<<endl; }
};

class B2:virtual public A,public B {
public:
    B2(int j):A(j){ cout<<"Constructing B2 "<<j<<endl; }
};

class D: public B1, public B2 {
public:
    D(int m,int n): B1(m),B2(n),a(3),A(4){ cout<<"Constructing D"<<endl; }
    A a;
};
void main(){
    D d(1,2);
}
```

3.（5分）

```
#include<iostream>
#include<cstdlib>
using namespace std;
struct student{
    int id,score;
};
template<class T>
class buffer
{
private:
    T a;
    int empty;
```

```
public :
    buffer();
    T get();
    void put(T x);
};
template <class T>
buffer<T>::buffer():empty(0){}
template <class T>
T buffer<T>::get()
{
    if(empty == 0)
    {
        cout << "the buffer is empty!" << endl;
        exit(1);
    }
    return a;
}
template<class T>
void buffer<T>::put(T x)
{
    empty ++;
    a = x;
}
int main()
{
    student s = {2011, 80};
    buffer<int> ibuf1, ibuf2;
    buffer<student> sbuf;
    buffer<double> dbuf;
    ibuf1.put(15);
    ibuf2.put(-202);
    cout << ibuf1.get() << " " << ibuf2.get() << endl;
    sbuf.put(s);
    cout << "the student's id is " << sbuf.get().id << endl;
```

```
    cout << "the student's score is " << sbuf.get().score << endl;
    cout << dbuf.get() << endl;
    return 0;
}
```

四．程序填空题（每空 2 分，共 30 分）

1．下列程序在构造函数和析构函数中申请和释放类的私有成员指向的内存空间，在横线处填上适当的语句，完成该类的实现。

```
class MClass{
public:
    MClass(int d);
    ~MClass();
private:
    int *p;
};
MClass::MClass(int d){
    _____(1)_____;
}
MClass::~MClass(){
    _____(2)_____;
}
```

2．在横线处填上适当的语句，完成下列程序，使得本程序的运行结果为 (1.2 , 7.8)

```
#include<iostream>
using namespace std;
class Location{
public:
    Location(double x,double y){
        _____(3)_____
    }
    double getx(){return x;}
    double gety(){return y;}
private:
    ___(4)___
protected:
```

```
        (5)
};
class Point: Location{
public:
    Point(double x,double y)    (6)    {
        (7)
    }
    void draw(){
        if(is_visible == false){
            cout << "(" << getx() << " , " << y << ")" << endl;
        }
        else
            is_visible = true;
    }
    void hide(){
        is_visible = false;
    }
private:
    bool is_visible;
};
int main()
{
    Point p(1.2, 7.8);
    p.draw();
    p.hide();
    return 0;
}
```

3．在横线处填上适当的字句，完成下面复数类的定义。

```
#include<iostream>
class Complex{
    double real;
    double image;
public:
    Complex(double r=0.0, double i=0.0){
```

```
        (8)    ;
        image=i;
    }
    Complex(    (9)    com1){
        real=com1.real;
        image=com1.image;
    }
    (10)    Complex operator+(Complex c1, Complex c2);
    Complex operator-(          (11)          ) const{
        return Complex((real-c.real), (image-c.image));
    }
    friend void print(Complex c);
};
Complex operator+(Complex c1, Complex c2){
    return          (12)          ;
}
```

4．在横线处填上适当的语句，利用异常处理机制合理得处理由主函数的两条调用语句导致的异常，使得：

当调用语句为 cout<<Divide(3,0)<<endl;时，输出结果为：除数不能为 0!

当调用语句为 cout<<Divide(2,1000)<<endl;时，输出结果为：除数太大!

当调用语句为 cout<<Divide(2,4)<<endl;时，输出结果为：0.5

```
#include <iostream>
using namespace std;
enum errs{error0,error1};
double Divide(double a, double b)
{
        if(b == 0)    (13)    ;
        else if(b>= 1000)    throw error1;
        else          (14)          ;
}
void main()
{
    try{
        cout<<Divide(3,0)<<endl;
```

```
        cout<<Divide(2,1000)<<endl;
        cout<<Divide(2,4)<<endl;
    }catch(errs er )
    {
        switch(er)
        {
            case error0 :
                cout<<"除数不能为 0!"<<endl;
                break;
            (15)    :
                cout<<"除数太大!"<<endl;
                break;
        }
    }
}
```

五．**程序设计题**（20 分）

1．(6 分)修改下面给出的程序，但不允许对 main()函数作任何修改，使程序能够在屏幕上输出唐诗：

床前明月光，
疑是地上霜。
举头望明月，
低头思故乡。

原来的程序为：

```
#include<iostream>
void main()
{
    cout<<“举头望明月，”<<endl;
}
```

2、(14分)下列Shape类是一个表示形状的抽象类，area( )为求图形面积的函数，total( )则是一个通用的用来求不同形状的图形面积总和的函数。

```
class Shape{
public:
    virtual double area()=0;
};
```

```
double total(Shape *s[ ], int n) {
    double sum=0.0;
    for(int i=0; i<n; i++)      sum+=s[i]->area( );
    return sum;
}
```

要求：

(1)从 Shape 类派生出圆类(Circle)、正方形类(Square)，圆类新增数据成员半径(radius)，正方形类新增数据成员边长(a)，圆类和正方形类都有构造函数，修改、显示数据成员值的函数，求面积函数。

(2)写出 main()函数，计算半径为 5.5 的圆和边长为 9.9 的正方形的面积和(必须通过调用 total 函数计算)。

浙江理工大学 2017 —2018 学年第二学期
《面向对象程序设计 A》期末试卷（A）卷

本人郑重承诺：本人已阅读并且透彻地理解《浙江理工大学考场规则》，愿意在考试中自觉遵守这些规定，保证按规定的程序和要求参加考试，如有违反，自愿按《浙江理工大学学生违纪处分规定》有关条款接受处理。

承诺人签名：_____ 学号：_____ 班级：_____

答题纸

一	二	三	四	五	总分

一．选择题（每小题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10

二．程序改错题，指出错误语句说明原因并改正之(共 10 分)

1．

2．

3．

4．

5．

三．程序阅读题(共 20 分)

1．（8 分）

2. （7 分）

3. （5 分）

四．程序填空题（每空 2 分，共 30 分）

(1)_____ (2)_____

(3)_____ (4)_____

(5)_____ (6)_____

(7)_____ (8)_____

(9)_____ (10)_____

(11)_____ (12)_____

(13)_____ (14)_____

(15)_____

五．程序设计题（20 分）

1.（6 分）

2.（14 分）