

诚信应考，考试作弊将带来严重后果！

# 华南理工大学期末考试

2022-2023-2 学期《高级语言程序设计 C++ (二)》A 卷

- 注意事项：1. 考前请将密封线内各项信息填写清楚；  
2. 所有答案写在答题纸上，答在其它地方无效；  
3. 试卷可做草稿纸，试卷必须与答题纸同时提交；  
4. 考试形式：闭卷；  
5. 本试卷共四大题，满分 100 分，考试时间 120 分钟。

## 一. 单项选择题 (每题 1 分, 共 20 分)

- 关于构造函数和析构函数特点的描述中，错误的是 ( )。  
(A) 析构函数无任何返回类型； (B) 析构函数没有形参；  
(C) 一个类中可定义 0 至多个构造函数； (D) 一个类可以定义 0 至多个析构函数。
- 在哪种派生方式中，派生类可以访问基类中的 private 成员 ( )  
(A). private (B). public (C). protected (D). 以上都不对
- 假定 Ashape 为一个类，则执行 “Ashape circle1, circle2(3), circle[2], \*p=&circle1;” 语句时共调用该类构造函数的次数为 ( )。  
(A). 5 (B). 6 (C). 3 (D). 4
- 假定 Matrix 是一个类名，且 Matrix 中重载了操作符=，可以实现 Matrix 对象间的连续赋值，如 “obj1=obj2=obj3;”。重载操作符=的函数原型最好是 ( )。  
(A). int operaotor=(Matrix); (B). int operator=(Matrix&);  
(C). Matrix operator=(Matrix&); (D). Matrix& operator=(Matrix)。
- 有如下程序：  

```
#include<iostream>
using namespace std;
class Base
{
    char c;
public:
    Base (char n):c(n){ }
    virtual ~Base(){cout<<c;}
};
```

```

class Derived: public Base
{
    char c;
public:
    Derived(char n):Base(n+1),c(n){};
    ~Derived () {cout<<c;};
};

```

```

int main()
{
    Derived ('C');
    return 0;
}

```

执行上面的程序，将输出：

( )

(A) C                      (B) DC                      (C) CD                      (D) D

6. 在下列运算符中，不能被重载的是 ( )。

(A) delete                      (B) ==                      (C) ::                      (D) new

7. 在下列关于运算符重载的描述中，( ) 是正确的。

(A) 可以改变参与运算的操作数个数； (B) 改变运算符原来的优先级；  
(C) 不可以改变运算符原来的结合性； (D) 可以创建新的运算符。

8. 有如下程序：

```

Class MyClass
{
    Private: int id; char gender; char *phone;
    Public:
        MyClass():id(0), gender('#'),phine(NULL){ }
        MyClass(int no, char ge='#',char *ph=NULL)
        { id=no; gender=ge; phone=ph;}
}

```

下列类对象定义语句中错误的是 ( )

(A) MyClass myObj(5, 'f', "13564378234");  
(B) MyClass myObj( "13547331212" );  
(C) MyClass myObj(7, 'm' );  
(D) MyClass myObj(12)。

9. 下列程序中划线处应填入的语句是 ( )。

```
#include <iostream>

class Person
{
public:
    void fun() {cout<<" fun of Person"<<endl; }
};

class Student:public Person
{
    _____; //调用基类的成员函数 fun
    cout<<"fun of Student" <<endl;
};
```

- (A) Person::fun()      (B) Person.fun()      (C) Fun()      (D) Person ->fun()

10. 下列关于 C++ 中类的继承与派生的说法, 正确的是 ( )

- (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不能直接使用;  
(C) 公有派生时, 基类的所有成员访问权限在派生类中保持不变;  
(D) 基类的 `protected` 成员在保护派生类的成员函数中不能直接使用。

11. 下列哪个运算符可以用友元函数重载 ( )

- (A) =      (B) ==      (C) ( )      (D) []

12. 分析下面的 C++ 代码段:

```
class Employee
{
private:
    int a;
protected:
    int b;
public:
    int c;
};

class Manager:public Employee{ };
```

在 `main()` 中, 下列 ( ) 操作是正确的。

- (A) `Employee obj; obj.b=1;`  
(B) `Manager obj; obj.b=10;`  
(C) `Employee obj; obj.c=3;`

(D) Manager8 obj; obj.a=20。

13. 类 Csample 的定义如下:

```
class Csample
{
public:
    Csample(){value=0;}
    void Setvalue(int i){value=i;}
private:
    int value;
};
```

则对语句 Csample \*p, my; p=&my; 正确的描述是 ( )

- (A) 语句 p=&my; 是把对象 my 赋值给指针变量 p;
- (B) 语句 Csample \*p, my; 会调用两次类 Csample 的构造函数;
- (C) 对语句\*p. Setvalue(5)的调用是正确的;
- (D) 语句 p ->Setvalue(3)与 my. Setvalue(3)等价。

14. 下列关于类层次中重名成员的描述, 正确的是 ( )。

- (A) C++不允许派生类的成员与基类成员重名;
- (B) 在派生类中访问重名成员时, 屏蔽基类的同名成员;
- (C) 在派生类中不能访问基类的同名成员;
- (D) 如果要在派生类中访问基类的同名成员, 不可以显式地使用作用域符指定。

15. 在具有继承关系的类层次体系中, 派生类本身的析构函数~C(), 对象成员析构函数~B(), 基类析构函数~A(), 析构函数执行的顺序是 ( )。

- (A) ~B() —~A() —~C();
- (B) ~C() —~B() —~A();
- (C) ~A() —~C() —~B();
- (D) ~A() —~B() —~C()。

16. 下列关于多继承的描述, 不正确的是 ( )。

- (A) 一个类不能被多次说明为一个派生类的直接基类, 可以不止一次地成为间接基类;
- (B) 为了建立唯一的间接基类版本, 应该声明派生类虚继承基类;
- (C) 为了建立唯一的间接基类版本, 应该声明间接基类为虚基类;
- (D) 对于不同基类的同名成员, 派生类对象访问它们时可能会出现二义性。

17. 下面关于虚函数的描述, 正确的是 ( )。

- (A) 在 C++中, 根据 this 指针类型识别类层次中不同类定义的虚函数版本;
- (B) 在 C++中, 根据参数个数和类型识别类层次中不同类定义的虚函数版本;

- (C) 基类的析构函数不可以是虚函数;  
(D) 构造函数可以声明为虚函数。

18. 使用串流类需要包含 ( ) 头文件。

- (A) `iostream`                      (B) `iomanip`                      (C) `fstream`                      (D) `sstream`

19. 下列程序执行中, `MyClass` 类的复制构造函数被调用的次数是( )

```
class MyClass
{
public:
    MyClass(int n){number=n;}
    MyClass(const MyClass &other) {number=other.number;}
    ~MyClass(){ }
private:
    int number;
};

MyClass fun(MyClass p)
{ MyClass temp(p); return temp;}

int main()
{
    MyClass obj1(10),obj2(0);
    MyClass obj3(obj1);
    obj2=fun(obj3);
    return 0;
}
```

- (A)2                                      (B) 3                                      (C)4                                      (D) 5

20. 下列选项中, 能够打开文件 `D:\temp\data1.dat`, 并读出数据的正确语句是 ( )。

- (A) `ofstream infile("D:\\temp\\data1.dat",ios::out);`  
(B) `ofstream infile("D:\\temp\\data1.dat",ios::in);`  
(C) `ifstream infile("D:\\temp\\data1.dat",ios::out);`  
(D) `ifstream infile("D:\\temp\\data1.dat",ios::in);`

二. 阅读下列程序, 写出执行结果 (每题 5 分, 共 30 分)

1.

```
#include<iostream>
using namespace std;
class Func1
```

```

{
public:
    int a1;
    Func1(){a1=10;}
    Func1(int i){a1=i;}
    void Show(){cout<<"a1="<<a1<<"\t"<<"Func1\n";}
};

class Func2
{
public:
    int b1;
    Func2(){b1=30;}
    Func2(int i){b1=i;}
    void Show(){cout<<"b1="<<b1<<"\t"<<"Func2\n";}
};

class Func3:public Func1,public Func2
{
public:
    int b1;
    Func3(int x,int y,int z):Func1(x),Func2(y+30){b1=z;}
    void Show(){cout<<"b1="<<b1<<"\t"<<"Func3\n";}
};

int main()
{
    Func3 my(20,30,40);
    my.b1=10;
    my.Show();my.Func1::Show();my.Func2::Show();
}

```

2.

```

#include <iostream>
using namespace std;
#include<cstring>
class B{
    char str[20];
public:

```

```

        B(char*s="animal"){ strcpy(str,s);cout<<str<<endl;}
};
class Ih1:public virtual B{
    char str1[20];
public:
    Ih1(char*s1,char*s2):B(s1){ strcpy(str1,s2);cout<<str1<<endl;}
};
class Ih2:public virtual B{
    char str2[20];
public:
    Ih2(char*s1,char*s2):B(s1){ strcpy(str2,s2);cout<<s2<<endl;}
};
class Ih3:public Ih1,public Ih2{
    char str3[20];
public:
    Ih3(char*s1,char*s2,char*s3,char*s4):Ih1(s1,s2),Ih2(s1,s3)
    { strcpy(str3,s4);cout<<str3<<endl;}
};
int main()
{
    Ih3 h3("cat","dog","giraffe","monkey");
}

```

3.

```

#include<iostream>
using namespace std;
class A{
public:
    A(){cout<<"A\n";}
    ~A(){cout<<"~A\n";}
};
class B:virtual public A{
public:
    B():A(){cout<<"B\n";}
    ~B(){cout<<"~B\n";}
};
class C:public virtual A{

```

```

public:
    C():A(){cout<<"C\n";}
    ~C(){cout<<"~C\n";}
};
class D:public C,public B{
public:
    D():B(),C(),A(){cout<<"D\n";}
    ~D(){cout<<"~D\n";}
};
void main()
{D d1;}

```

4.

```

#include <iostream>
using namespace std;
class Bclass
{ public:
    Bclass( int i, int j ) { x = i; y = j; }
    virtual int fun() { return (x+y)/2 ; }
protected:
    int x, y ;
};
class Iclass:public Bclass
{ public :
    Iclass(int i, int j, int k):Bclass(i, j) { z = k; }
    int fun() { return ( x*x + y*y + z*z ) / 2; }
private :
    int z ;
};
int main()
{ Iclass obj( 3, 5, 8 );
    Bclass p1 = obj;
    cout << p1.fun() << endl;
    Bclass &p2 = obj ;
    cout << p2.fun() << endl;
    cout << p2.Bclass :: fun() << endl;
    Bclass *p3 = &obj;
}

```



```

        cout << p3 -> fun() << endl;
    }

```

5.

```

#include<iostream>
#include <iomanip>
using namespace std;
int main()
{ double x = 314.15926;
    cout << setiosflags( ios::fixed | ios::showpos ) << setprecision( 4 ) << x << endl;
    cout << setw( 12 ) << setfill('*') << setiosflags( ios::right );
    cout << setprecision( 3 ) << -x << endl;
    cout << resetiosflags( ios::fixed | ios::showpos )
        << setiosflags( ios::scientific );
    cout << setprecision( 5 ) << x << endl;
}

```

6.

```

#include<iostream>
using namespace std;
class Test
{ private:
    static int val;
    int a;
public:
    static int func( );
    void sfunc(Test &r);
};
int Test::val=10;
int Test::func( )
{ return val++; }
void Test::sfunc(Test &r)
{ r.a=20; cout<<"Result3="<<r.a<<endl;}
void main( ){
    cout<<"Result1="<<Test::func( )<<endl;
    Test a;
    cout<<"Result2="<<a.func()<<endl;
}

```

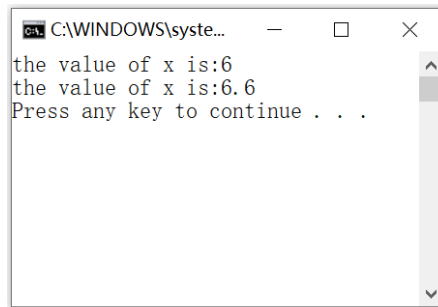
```

        a.sfunc(a);
    }

```

### 三. 将程序补充完整。(每空 2 分, 共 24 分)

1. 以下程序的运行结果如图所示, 请将程序补充完整。



```

#include <iostream>
using namespace std;
_____ (1) _____ //模板说明
class Asample
{
    T x;
public:
    _____ (2) _____ (Asample<T> s1, Asample<T> s2) //友元函数重载加运算符
    {
        Asample<T> tmp;
        tmp.x=(s1.x+s2.x)/2;
        return tmp;
    }
    Asample() {}
    Asample(T i) {x=i;}
    void func() {cout<<"the value of x is:"<<x<<"\n";}
};

int main()
{
    _____ (3) _____ i1(5),i2(7),i3; //定义对象
    _____ (4) _____ d1(5.5),d2(7.7),d3; //定义对象
    i3=i1+i2;
    i3.func();
    d3=d1+d2;
    d3.func();
}

```

}

2. 以下程序的功能是计算圆、矩形和直角三角形的总面积。程序的运行结果如下。请将程序补充完整。

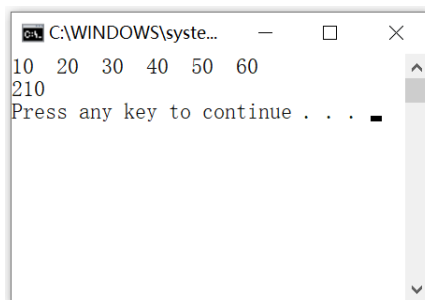
```
#include<iostream>
using namespace std;
class Shape
{
public:
    Shape(double i) {H=i;}
    _____(5)_____=0;
protected:
    double H;
};
class circle:public Shape
{
public:
    circle(double r):Shape(r){ }
    double area(){return H*H*3.14;}
};
class triangle:public Shape
{
protected:double W;
public:
    _____(6)_____:Shape(h){ W=w;}
    double area(){return H*W*0.5;}
};
class rectangle:public Shape
{
```

```

protected:double W;
public:
    rectangle(double h,double w): Shape(h) { W=w;}
    double area(){return H*W;}
};
double total(Shape* s[],int n)
{
    double sum=0.0;
    for(int i=0;i<n;i++)
        sum+=s[i]->area();
    return sum;
}
int main()
{
    Shape *s[3];
    s[0]=new circle(10);
    s[1]=new triangle(5,8);
    s[2]=new rectangle(3,6);
    for(int i=0;i<3;i++)
        cout<<"s["<<i<<"]="<<_____ (7) <<endl;
    double sum=_____ (8) ;
    cout<<"The total area is:"<<sum<<endl;
}

```

3. 程序的运行结果如下。请将程序补充完整。



```

#include<iostream>
_____ (9)
using namespace std;
int main()
{

```

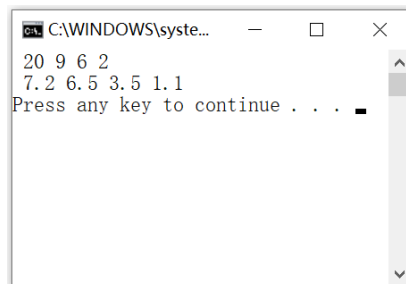
```

int n[6],s=0;
int i;
for(i=0;i<sizeof(n)/sizeof(int);i++) n[i]=(i+1)*10;
_____(10)_____; //写二进制文件
if(!ofile){cout<<"cannot open file.\n";return 1;}
_____(11)_____; //将数据写入文件
ofile.close();
for(i=0;i<sizeof(n)/sizeof(int);i++) n[i]=0;
ifstream ifile("test.bin",ios::in|ios::binary);
if(!ifile){cout<<"cannot open file.\n";return 1;}
ifile.read((char*)&n,sizeof(n));
for(i=0;i<sizeof(n)/sizeof(int);i++)
{
    cout<<n[i]<<" ";
    s+=n[i];
}
cout<<endl<<s<<endl;
_____(12)_____; //关闭文件
return 0;
}

```

#### 四、完成程序。（第 1 小题 12 分，第 2 小题 14 分，共 26 分）

1. 采用函数模板定义排序，如下程序能实现的输出界面如图，请补充程序。



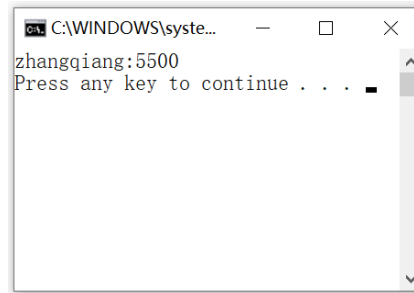
```

#include<iostream>
using namespace std;
const int NUM=4;
...
void main()
{ int a[NUM] = { 13, 4, 7, 20 };
  double b[NUM] = { 6.5, 3.5, 5.2, 1.1 };
  bubbleSort(a);
  traverse(a);
  bubbleSort(b);
}

```

```
    traverse(b);  
}
```

2. 设某公司的工资发放系统通过定义类 Employee 对员工按照职称等级发放基本工资,通过 earning 函数定义 1 级为 3000 元, 2 级为 5000 元。 现该公司下属某一部门, 希望在总公司基本工资的基础上, 加上员工加班工资, 设加班小时数 hours, 加班工资为 hours\*50, 在不修改总公司类 Employee 的情况下, 编写派生的部门类 Employee1。 已知部门一员工 “zhangqiang”, 职称为 2 级, 加班时间为 10 小时, 输出如下图所示。 补充类的定义。



```
#include <iostream>  
#include <cstring>  
using namespace std;  
class Employee  
{  
protected:  
    char *name;  
    int degree;  
    int earning;  
public:  
    Employee(){degree=0;name='\0';}  
    Employee( int d,char *s)  
    {degree=d;name=new char[strlen(s)+1];strcpy(name,s); earnings();}  
    ...  
};  
int main()  
{  
    Employee1 emp("zhangqiang",2,10);  
    emp.print();  
}
```