一．单项选择题（每小题 2分，共20分）

1．在 C++ 中，数据与操作的封装是借助于 \_\_\_\_\_\_\_\_\_\_ 达到的。

(A) 指针 (B) 类 (C) 数组 (D) 函数

2．下面叙述不正确的是 \_\_\_\_\_\_\_\_\_\_ 。

(A) 基类的保护成员在派生类中仍然是保护的

(B) 基类的保护成员在公有派生类中仍然是保护的

(C) 基类的保护成员在私有派生类中仍然是保护的

(D) 对基类成员的访问必须是无二义性的

3．以下基类中的成员函数中表示纯虚函数的是 \_\_\_\_\_\_\_\_\_\_ 。

(A) virtual void vf(int); (B) virtual void vf(int)=0;

(C) virtual void vf(int=0); (D) virtual void vf(int){ };

4．下面对类的析构函数的定义是 \_\_\_\_\_\_\_\_\_\_ 。

(A) X::~ X( 参数 ); (B) void X::~ X( 参数 );

(C) void X::~ X(); (D) X::~ X();

5．下列关于运算符重载的描述中， \_\_\_\_\_\_\_\_\_\_ 是正确的。

(A) 运算符重载可以改变操作数的个数。

(B) 运算符重载可以改变优先级。

(C) 运算符重载可以改变结合性。

(D) 运算符重载不可以改变原语法规则。

6．编译时的多态性通过使用 \_\_\_\_\_\_\_\_\_\_ 实现。

(A) 构造函数 (B) 虚函数 (C) 重载函数 (D) 析构函数

7．下面对类的构造函数的定义是 \_\_\_\_\_\_\_\_\_\_ 。

(A) void X::X( 参数 ) (B) X::X( 参数 )

(C) int X::X( 参数 ) (D) float X::X( 参数 )

8．已知类 A 中的一个成员函数说明如下： void Set( A & a ) 其中， A & a 的含义是 \_\_\_\_\_\_\_\_\_\_ 。

(A) 指向 A 类的指针为 a

(B) 将 a 的地址值赋给变量 Set

(C) a 是类 A 的引用，用来做函数 Set() 的形参

(D) 变量 A 和 a 是函数 Set() 的两个形参

9．下列函数中， \_\_\_\_\_\_\_\_\_\_ 是对文件进行写操作的。

(A) get() (B) read() (C) seekg() (D) put()

10．写一个 C++ 文件，要创建一个 \_\_\_\_\_\_\_\_\_\_ 流对象。

(A) ifstream (B) ofstream (C) cin (D) cout

二．程序填空题（ 30 分）

1. #include<iostream.h>

class A

{ \_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_

void f(int i){cout<<i<< ' \t ' ; }

void g(){cout<<"g\n"; }

};

class B:private A

{ public:

void h(){cout<<"h\n";}

\_\_\_\_\_\_\_(2)\_\_\_\_\_\_\_

};

void main()

{ B d;

d.f(6); d.h();

}

输出： 6 h

2. #include <iostream.h>

class p\_class

{ int num ;

public :

void set\_num ( int val ) { num = val ; }

void show\_num ( ) ;

} ;

void p\_class :: show\_num ( ) { cout << num << ‘ \t ' ;}

void main ( )

{ p\_class ob[3] , \* p ;

for ( int i = 0; i<3 ; i++ ) ob[i].set\_num((i+1)\*10);

\_\_\_\_\_\_\_\_(3)\_\_\_\_\_\_\_\_ ; p -> show\_num ( ) ;

\_\_\_\_\_\_\_\_(4)\_\_\_\_\_\_\_\_ ; p -> show\_num ( ) ;

\_\_\_\_\_\_\_\_(5)\_\_\_\_\_\_\_\_ ; p -> show\_num ( ) ;

}

输出： 20 30 10

3. #include<iostream.h>

class Tdate

{ int month ; int day ; int year ;

int IsLeapYear(){ return(year%4==0&&year%100!=0)||(year%400==0);}

\_\_\_\_\_\_\_\_(6)\_\_\_\_\_\_\_\_

void Set (int m, int d, int y) { month=m; day=d; year=y; }

\_\_\_\_\_\_\_\_(7)\_\_\_\_\_\_\_\_ ;

} ;

\_\_\_\_\_\_\_\_\_(8)\_\_\_\_\_\_\_ Print()

{ cout << month <<"/"<< day <<"/"<< year<<endl ;

if(IsLeapYear())cout<<"This year Is leapyear.\n" ;

else cout<<"This year Is not leapyear.\n";

}

void main()

{ Tdate a ; a.Set ( 2, 4, 2000 ) ; a.Print() ; }

4. #include<iostream.h>

class Time

{ public:

Time(int h, int m) {hours=h; minutes=m;}

\_\_\_\_\_\_\_\_(9)\_\_\_\_\_\_\_\_ Time12();

\_\_\_\_\_\_\_\_(10)\_\_\_\_\_\_\_\_ Time24(Time time);

private: int hours, minutes;

};

\_\_\_\_\_\_\_\_\_(11)\_\_\_\_\_\_\_\_\_\_ Time12()

{ if(hours>12) { cout<<hours-12<<":"<<minutes<<"PM\n" ; }

else cout<<hours<<":"<<minutes<<"AM\n" ;

}

void Time24(Time time)

{ cout \_\_\_\_\_\_\_\_\_\_\_\_\_(12)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;}

void main()

{ Time T1(20,30), T2(10,45);

T1.Time12(); Time24(T1); T2.Time12(); Time24(T2);

}

输出： 8:30PM

20:30

10:45AM

10:45

5. #include<iostream.h>

class A

{ public:

A(const char \*s){cout<<s<<endl;}

~A(){}

};

class B: \_\_\_\_\_\_\_\_(13)\_\_\_\_\_\_\_\_ A

{ public:

B(const char \*s1, const char \*s2):A(s1) {cout<<s2<<endl;}

};

class C: \_\_\_\_\_\_\_\_(14)\_\_\_\_\_\_\_\_

{ public:

C(const char \*s1, const char \*s2):A(s1) { cout<<s2<<endl;}

};

class D:public B, public C

{ public:

D(const char \*s1,const char \*s2,const char \*s3,const char \*s4);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(15)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{ cout<<s4<<endl;}

};

void main()

{ D \*ptr=new D("class A","class B","class C","class D");

delete ptr;

}

输出： class A

class B

class C

class D

三．读程序写运行结果 （24分）

1．

#include <iostream.h>

class desk

{ public:

desk(){ length=3; width=4; high=5; }

desk(int l,int w,int h) { length=l;width=w;high=h;}

int volume(){ return length\*width\*high;}

int price() { return volume()\*2;}

private:

int length,width,high;

};

class luxury\_desk:public desk

{ public:

luxury\_desk() { m\_price=2.5;}

float price() { return volume()\*m\_price;}

private:

float m\_price;

};

void main()

{ desk da,da1(1,2,3);

luxury\_desk db;

cout<<da.volume()<<" "<<da.price()<<" ";

cout<<db.volume()<<" "<<db.price()<<" ";

cout<<da1.volume()<<" "<<da1.price()<<" ";

}

2．

#include <iostream.h>

class B

{ int b;

public:

B(int i) {b=i;}

virtual void virfun() {cout<< "B::b: "<<b<<" , ";}

};

class D: public B

{ public:

D(int i,int j): B(i) {d=j;}

void virfun() {B::virfun(); cout<<"D::d: "<<d<<endl;}

private: int d;

};

void fun(B \*objp) { objp->virfun(); }

void main() { D \*pd=new D(3,5) ; fun(pd);}

3．

#include <iostream.h>

enum BoolConst { False=0 , True=1 }; //定义枚举类型

class Boolean

{ public:

Boolean(BoolConst x = False) { logic = x; }

void print() const { logic? cout<<" TRUE " : cout<<" FALSE "; }

friend Boolean operator +(const Boolean & obj1, const Boolean & obj2);

friend Boolean operator \*(const Boolean & obj1, const Boolean & obj2);

protected: BoolConst logic;

};

Boolean operator + ( const Boolean & obj1, const Boolean & obj2 )

{ return (obj1.logic || obj2.logic) ? Boolean(True) : Boolean(False); }

Boolean operator \* ( const Boolean & obj1, const Boolean & obj2 )

{ return (obj1.logic && obj2.logic) ? Boolean(True) : Boolean(False); }

void main()

{ Boolean a(False), b(True), c, d ;

c = a \* b; d = a + b;

a.print(); b.print(); c.print(); d.print();

cout<<endl;

}

4．写出data.txt中的结果和屏幕显示的结果。

#include <fstream.h>

int filerror(const fstream &, char \*);

void main()

{ int a=10; double x=50.5;

char str[10], fname[20]= "d:\\data.txt";

fstream iofile(fname,ios::out);

if(!iofile) return;

iofile<<"string\n"<<a+10<<" "<<x<<endl;

iofile.close();

iofile.open(fname, ios::in);

if(!iofile) return;

iofile>>str>>a>>x;

cout<<"x="<<x<<", a= "<<a<<"\nstring="<<str<<"!"<<endl;

}

四．改错题。找出以下程序的错误，并注释错误原因。（10分）

1．

class Time

{ public:

Time(int day){ iMonth=5; iDay=day; iYear=2000; }

Timt(int month, int day=7 ) { iMonth=month; iDay=day; iYear=2000; }

Private:

int iMonth; int iDay; int iYear;

};

void main()

{ Time input(9,1,2005);

input.iMonth=1;

}

2．

Template<class T>

Type func ( T a , T b )

{ return ( a > b ? a : b ); }

#include<iostream.h>

void main()

{ int a=3 ; double x=3.5 ;

cout<<func(a,x)<<endl;

}

五．完成程序。（共16分）

1． 一个类的头文件如下所示：

//test.h

#include<iostream.h>

class test

{ private: int num;

public: test(int); void show();

};

test::test(int n) { num = n; }

void test::show() { cout<<num<<endl;}

编写一个主程序，产生对象TTT，令TTT 数据成员的值为 5，并使用show()函数输出这个对象的数据成员。

2． 根据下面的主程序，补上所缺的类说明文件的最小形式。

#include<iostream.h>

#include ”base.h”

void main()

{ base stry;

stry.init(6); //对数据成员赋值

cout<<stry.Getnum(); //输出数据成员的值

}