**《C++面向对象程序设计》答案**

一、每个2分

B, A, B, C, B A, A, B, D, A

二、每空2分

1、封装、继承、多态

2、Point(){}

3、初始化列表

4、friend void f();

5、2

6、Point(const Point& a);

7、指针类型，指针所指向对象类型

8、class/typename

9、类成员

三、每个修改1分

1、#include<iostream>

using namespace std;

int f(int m=0, int n){ //n缺少默认参数

return x\*y; //未定义x，y

}

void main(){

int tmp=10;

const int i; //常量没有初始化

int &rr1=tmp;

int &&rr2=tmp; //右值引用赋值号右侧是数值或表达式

rr1=i; //引用不可修改

cout<<f(2,3);

}

2、#include<iostream>

void main(){

char \*a=”computer”;

const char \*pointer1=”computer”;

char \*const pointer2=”computer”;

char const \*pointer3=”computer”;

const char const \*pointer4=”computer”;

pointer1[2]=”a”; //pointer1是指向常量的指针，所指向内容不能修改

pointer2[2]=”a”; // pointer2是指向常量的指针，所指向内容不能修改

pointer3[2]=”a”;

pointer4[2]=”a”; // pointer4是指向常量的指针，所指向内容不能修改

pointer=&a; //赋值类型不匹配

pointer1=pointer4;

pointer2=pointer4; //指针常量不能修改

pointer3=pointer4;

}

3、#include<iostream>

using namespace std;

class A {

int x=0;

A(int a) { x = a; } //构造函数不能私有

public:

setA(int y) { x = y; } //没有返回类型

};

class B : private A {

public:

B(int b):A(b) { cout << "B" << endl; }

};

void main() {

A a1(2),a2; //没有无参的构造函数

A a3 = a1;？

B b(1);

b.setA(3); //私有

}

4、#include<iostream>

using namespace std;

template<class T>

int i = 10; //template与函数之间不可存在语句

T max(T a, T b) { return a>b ? a : b; }

void main() {

int x;

cin >>"x=">> x; //cin不可输入字符串，只能接变量

int j = max(3, x);

int i = max(3, 4.2); //T类型模糊，无法确定

char ch = max('a', 1); //T类型模糊，无法确定

char ch = max('a', 'A');

float sh = max(3.0, 2); //T类型模糊，无法确定

}

四、每个结果5分

1、1000 2000 1000 1000

2、constructor:1

constructor:2

constructor:3

constructor:4

destructor:4

destructor:3

destructor:2

destructor:1

3、Constructing B

Constructing A

Constructing C1

Constructing A

Constructing C2？？？？？？？？？？？？

Constructing A

Constructing D

4、constructor1

constructor0

operator1

destructor1

destructor1

五、每道题10分，每个类成员函数1分，main函数两分

1、#include<iostream>

#include<string>

using namespace std;

class Employee{

string name;

string address;

string number;

public:

Employee(){name=”zhang”;address=”Beijing”;number=”188000000000”}

Employee(string a, string b, string c){name=a;address=b;number=c;}

Employee(const Employee &e){name=e.name;addree=e.address;number=e.number;}

void change\_name(string a){name=a;}

void display(){cout<<name<<” ”<<address<<” ”<<number<<endl;}

};

void main() {

Employee e1;

e1.display();

Employee e2("zhang", "Beijign", "13811111111");

e2.display();

Employee e3 = e2;

e3.display();

e2.display();

e1.display();

cout << "the number of employee:" << Employee::number << endl;

}

2、

#include <iostream>

using namespace std;

class Figure{

protected:

double x,y;

public:

void set(double i,double j){ x=i; y=j; }

virtual void area()=0;

virtual void print\_name()=0;

};

class Circle:public Figure{

int r;

public:

Circle(int i){r=i}

void area(){cout<<3.14\*r\*r<<endl;}

void print\_name(){cout<<” Circle”<<endl}

};

class Rectangle:public Figure{

public:

void area(){cout<<x\*y<<endl;}

void print\_name(){cout<<” Rectangle”<<endl}

};

int main(){

Figure \*pF;

Rectangle r;

Circle t(1);

pF=&t;

pF->area();

pF->print\_name();

Figure &rF=r;

rF.set(20,20);

rF.area();

rf.print\_name();

}