**C++面向对象程序设计(甲)（B）卷参考答案**

**座位号：**

**注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交**

**一、单项选择题（每题2分，总计20分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **题号** | **1** | **2** | **3** | **4** | **5** |
| **答案** | **D** | **A** | **B** | **B** | **A** |
| **题号** | **6** | **7** | **8** | **9** | **10** |
| **答案** | **B** | **C** | **C** | **B** | **D** |

**二、程序填空题（每空2分，总计20分）**

**（1）word.disp()**

**（2）cout << word[i]**

**（3）T a[], int i, T b[], int j**

**（4）sum(a, i) + sum(b, j)**

**（5）a +=x**

**（6）A::show()**

**（7）return complex(x+p.x, y+p.y)**

**（8）p3=p1+p2**

**（9）bool operator()(int x);**

**（10）Error: the side length should fit the triangle inequality**

**三、阅读程序题（每题6分，总计30分）**

**1. 阅读该程序，给出程序的输出结果。**

**a=7, b=8**

**2. 阅读该程序，给出程序的输出结果。**

**9 7 2 1 5**

**3. 阅读该程序，给出程序的输出结果。**

**s1=11**

**s2=8.2**

**4. 阅读该程序，给出程序的输出结果。**

**base class**

**base class**

**base class**

**derive1 class**

**derive2 class**

**5. 阅读该程序，给出程序的输出结果。**

**Error : the side length should fit the triangle inequality**

**四、编程题（每题15分，总计30分）**

**1. 按照要求，编写程序。**

**2. 按照要求，编写程序。**

**座位号：**

**C++面向对象程序设计(甲)（B）卷**

**注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交**

**一、单项选择题（每题2分，总计20分）**

|  |  |
| --- | --- |
| **1.** | **下面关于友元的描述中，错误的是\_\_\_D\_\_。**  **A. 友元函数可以访问该类的私有数据成员**  **B. 一个类的友元类中的成员函数都是这个类的友元函数**  **C. 友元可以提高程序的运行效率**  **D. 类与类之间的友元关系可以继承** |
| **2.** | **下列对重载函数的描述中，\_\_\_A\_\_是错误的。**  **A. 重载函数中不允许使用默认参数**  **B. 重载函数中编译是根据参数表进行选择的**  **C. 不要使用重载函数来描述毫无相干的函数**  **D. 构造函数重载将会给初始化带来多种方式** |
| **3.** | **有如下程序：**  **#include<iostream.h>**  **class Test**  **{**  **public:**  **Test (){}**  **~Test() { cout << ‘#’; }**  **};**  **void main()**  **{ Test temp[2],\*pTemp[2]; }**  **执行这个程序输出“#”号的个数为\_\_2\_\_\_。**  **A. 1 B. 2 C. 3D. 4** |
| **4.** | **const int \*p说明不能修改\_\_\_B\_\_\_\_**  **A. p指针 B. p指针指向的量**  **C. p指针指向的数据类型 D. 上述A、B、C三者** |
| **5.** | **友元运算符obj1>obj2被C++编译器解释为\_A\_\_\_。**  **A. operator>(obj1,obj2) B. >(obj1,obj2)**  **C. obj2.operator>(obj1) D. obj1.operator>(obj2)** |
| **6.** | **现有语句int iarray[] = { 0, 1, 2, 3, 4, 5, 6, 6, 6, 7, 8 };**  **vector<int> ivector(iarray, iarray + sizeof(iarray) / sizeof(int));**  **请问要找出ivector之中大于2的第一个元素所在位置的元素，采用下面哪个算法\_B\_\_\_\_。**  **A. count\_if B. find\_if C. sort D. find** |
| **7.** | **下列关于虚基类的描述中，错误的是\_\_C\_\_\_。**  **A. 使用虚基类可以消除由多继承产生的二义性**  **B. 构造派生类对象时，虚基类的构造函数只被调用一次**  **C. 声明class B:virtual public A说明类B为虚基类**  **D. 建立派生类对象时，首先调用虚基类的构造函数** |
| **8.** | **在下面的程序中，A、B、C、D四个语句编译时出现错误的是\_\_C\_\_\_。**  **class A //A**  **{**  **public: //B**  **A() { func() ; } //C**  **virtual void func()=0; //D**  **};** |
| **9.** | **若定义cin>>str;,当输入Object Windows Programming! 所得的结果是str=\_\_B\_\_\_。**  **A. Object Windows Programming! B. Object**  **C. Object Windows D. Object Windows Programming** |
| **10** | **类模板的模板参数\_\_D\_\_\_。**  **A. 只可作为数据成员的类型 B. 只可作为成员函数的返回类型**  **C. 只可作为成员函数的参数类型 D. 以上三者皆可** |

**二、程序填空题（每空2分，总计20分）**

|  |  |
| --- | --- |
| **1.** | **请在下面程序的横线处填上适当内容，以使程序完整,并使运行结果为：**  **Hello**  **hELLO**  **#include <iostream.h>**  **#include <string.h>**  **class Words**  **{ char \*str;**  **public:**  **Words(char \*s)**  **{ str=new char[strlen(s)+1];**  **strcpy(str,s);**  **}**  **void disp() {cout<<str<<endl;}**  **char operator[](int i)**  **{ if(str[i]>='A'&&str[i]<='Z')**  **return char(str[i]+32);**  **else if(str[i]>='a'&&str[i]<='z')**  **return char(str[i]-32);**  **else**  **return str[i];**  **}**  **};**  **int main()**  **{ int i;**  **char \*s="Hello";**  **Words word(s);**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_(1)\_\_\_\_\_\_\_\_\_\_\_\_; // word.disp()**  **for (i=0;i<strlen(s);i++)**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_(2)\_\_\_\_\_\_\_\_\_\_\_\_; // cout << word[i];**  **cout<<endl;**  **}** |
| **2.** | **利用函数模板，设计求一个数组元素之和的函数sum和两个数组元素之和的函数sum，请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为:**  **数组a之和:15**  **数组b之和:55**  **两数组之和:70**  **#include <iostream.h>**  **template <class T>**  **T sum (T a[],int n)**  **{ int i;**  **T s=0;**  **for(i=0;i<n;i++)**  **s+=a[i];**  **return s;**  **}**  **template <class T>**  **T sum(\_\_\_\_\_\_\_\_\_\_\_\_\_\_(3)\_\_\_\_\_\_\_\_\_\_\_\_) // (T a[], int i, T b[], int j)**  **{**  **return \_\_\_\_\_\_\_\_\_\_\_\_\_\_(4)\_\_\_\_\_\_\_\_\_\_\_\_; // sum(a, i) + sum(b, j)**  **}**  **int main()**  **{ int a[5]={1,2,3,4,5};**  **int b[10]={1,2,3,4,5,6,7,8,9,10};**  **int s1=sum(a,5);**  **int s2=sum(b,10);**  **int s3=sum(a,5,b,10);**  **cout<<"数组a之和:"<<s1<<endl;**  **cout<<"数组b之和:"<<s2<<endl;**  **cout<<"两数组之和:"<<s3<<endl;**  **}** |
| **3.** | **请在下面程序的横线处填上适当字句，以使程序完整,并使程序的输出为**  **(1,2)**  **5,6**  **(6,9)**  **#include <iostream.h>**  **class A**  **{**  **public:**  **A(int i,int j){a=i;b=j;}**  **void move(int x,int y){ \_\_\_\_(5)\_\_\_\_\_\_;b+=y;} // a+=x**  **void show(){cout<<"("<<a<<","<<b<<")"<<endl;}**  **private:**  **int a,b;**  **};**  **class B:private A**  **{**  **public:**  **B(int i,int j,int k,int l):A(i,j){x=k;y=l;}**  **void show(){cout<<x<<","<<y<<endl;}**  **void fun(){move(3,5);}**  **void fl(){\_\_\_\_(6)\_\_\_\_\_\_;} // A::show()**  **private:**  **int x,y;**  **};**  **int main()**  **{ A e(1,2);**  **e.show();**  **B d(3,4,5,6);**  **d.fun();**  **d.show();**  **d.fl();**  **}** |
| **4.** | **请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为:**  **10+10i**  **#include <iostream.h>**  **class complex{**  **int x,y;**  **public:**  **complex(){}**  **complex(int i,int j)**  **{x=i; y=j;}**  **complex operator+(complex &p)**  **{\_\_\_\_\_\_\_\_(7)\_\_\_\_\_\_\_;} // return complex(x+p.x, y+p.y)**  **void show()**  **{cout<<x<<"+"<<y<<"i";}**  **};**  **int main()**  **{ complex p1(5,8),p2(5,2),p3;**  **\_\_\_\_\_\_\_\_(8)\_\_\_\_\_\_\_; // p3=p1+p2**  **p3.show();}** |
| **5.** | **下面程序用STL的条件计数算法和自定义的函数对象对一个存放在整数向量类对象中的学生成绩进行统计及格人数并显示结果。请在下面程序的横线处填上适当字句，以使程序完整。**  **#include<iostream>**  **#include<vector>**  **#include<algorithm>**  **#include<functional>**  **using namespace std;**  **class Pass**  **{**  **public:**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(9)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //** bool operator()(int x);  **{**  **return x>=60;**  **}**  **};**  **int main()**  **{**  **vector<int> a;**  **int count;**  **a.push\_back(78);**  **a.push\_back(92);**  **a.push\_back(52);**  **count =\_\_\_\_\_\_(10)\_\_\_\_\_\_\_\_; //** count\_if (a.begin (), a.end (), Pass())  **cout<<"count="<<count<<endl;**  **}** |

**三、程序阅读题（每题6分，总计30分）**

|  |  |
| --- | --- |
| **1.** | **#include <iostream.h>**  **class B**  **{**  **public:**  **B(){}**  **B(int i,int j){a=i;b=j;}**  **void printb()**  **{ cout<<"a="<<a<<",b="<<b<<endl;}**  **private:**  **int a,b;**  **};**  **class A**  **{ B c;**  **public:**  **A(){}**  **A(int i,int j):c(i,j){}**  **void printa(){c.printb();}**  **};**  **int main()**  **{ A a(7,8);**  **a.printa(); // a=7,b=8**  **}** |
| **2.** | **#include <iostream.h>**  **class CArray**  **{ int \*m\_pArray;**  **int m\_iSize;**  **public:**  **CArray(int iArray[],int iSize)**  **{ m\_pArray=iArray;**  **m\_iSize=iSize;**  **}**  **int GetSize()**  **{ return m\_iSize;}**  **int &operator[](int i)**  **{ return m\_pArray[i];}**  **};**  **int main()**  **{ int s[]={3,7,2,1,5};**  **CArray oArray(s,5);**  **oArray[0]=9;**  **for(int i=0;i<5;i++)**  **cout<<oArray[i]<<" "; // 9 7 2 1 5**  **cout<<endl;**  **}** |
| **3.** | **#include <iostream.h>**  **#include <math.h>**  **template <class T>**  **class TAdd**  **{ T x,y;**  **public:**  **TAdd(T a,T b){x=a,y=b;}**  **T add(){return x+y;}**  **};**  **int main()**  **{ TAdd<int> A(5,6);**  **TAdd<double> B(2.4,5.8);**  **cout<<"s1="<<A.add()<<endl; // s1=11**  **cout<<"s2="<<B.add()<<endl; // s2=8.2**  **}** |
| **4.** | **#include <iostream.h>**  **class base**  **{**  **public:**  **void who(){cout<<"base class"<<endl;}**  **};**  **class derive1:public base**  **{**  **public:**  **void who(){cout<<"derive1 class"<<endl;}**  **};**  **class derive2:public base**  **{**  **public:**  **void who(){cout<<"derive2 class"<<endl;}**  **};**  **int main()**  **{ base obj1,\*p;**  **derive1 obj2;**  **derive2 obj3;**  **p=&obj1;**  **p->who(); // base class**  **p=&obj2;**  **p->who(); // base class**  **p=&obj3;**  **p->who(); // base class**  **obj2.who(); //derive1 class**  **obj3.who(); //derive2 class**  **}** |
| **5.** | **#include <iostream>**  **#include <cmath>**  **#include <stdexcept>**  **using namespace std;**  **double area(double a, double b, double c) {**  **if (a <= 0 || b <= 0 || c <= 0)**  **throw invalid\_argument("the side length should be positive");**  **if (a + b <= c || b + c <= a || c + a <= b)**  **throw invalid\_argument("the side length should fit the triangle inequality");**  **double s = (a + b + c) / 2;**  **return sqrt(s \* (s - a) \* (s - b) \* (s - c));**  **}**  **int main() {**  **double a, b, c;**  **cin >> a >> b >> c; //输入1 2 4**  **try {**  **double s = area(a, b, c);**  **cout << "Area: " << s << endl;**  **} catch (exception &e) {**  **cout << "Error: " << e.what() << endl;**  **}**  **}**  **// Error: the side length should fit the triangle inequality** |

**四、编程题（每题15分，总计30分）**

|  |
| --- |
| **1. 按下列要求编程，既实现相关类，并在主函数中使用这些类。**  **某高校规定：教授（professor）的基本工资4000元，每堂课35元；副教授（viceprofessor）的基本工资为3500元，每堂课30元；讲师（lecturer）的基本工资为3000元，每堂课25元。教授、副教授、讲师每个月上多少堂课由各个类的成员函数完成设置。请编程实现某人的月工资收入是多少。（要求用抽象类和类继承）**  **2、编程声明一个点类Point，采用运算符重载方式实现点的偏移、==、！=、+=、-=、+、-和输出等运算，并在主函数中用若干数据进行测试。** |