1、运行过程中的多态需要满足的条件：类之间满足赋值兼容原则；声明虚函数；由成员函数来调用或者是通过指针、引用来访问虚函数。

2、请简述构造函数的作用是：构造函数的作用就是在对象被创建时利用特定的值构造对象，将对象初始化为一个特定的状态。构造函数在对象被创建的时候自动调用。

3、函数重载是两个以上函数，具有相同的函数名，但是形参的个数或者类型不同，编译器根据实参和形参的类型及个数的最佳匹配，自动确定调用哪一个函数，这就是函数重载。

4、三个继承的特性是public：基类公有和保护成员的访问属性在派生类中不变，而基类私有成员不可直接访问；protected：基类公有和保护成员都以保护成员的身份出现在派生类中，而基类私有成员不可直接访问；private： 基类公有和保护成员都以私有成员的身份出现在派生类中，而基类私有成员不可直接访问。

5、C++与C语言的不同之处是C++支持面向对象编程，而C不支持。C++引入了类、对象、继承、多态等概念，使得程序设计更加灵活、抽象和模块化。C++支持泛型编程，而C不支持。C++提供了模板机制，可以实现数据结构和算法的参数化，提高了代码的复用性和效率。C++使用名称空间，而C不使用。C++使用名称空间（namespace），将相同逻辑的代码组织到一起中，以防止名称冲突。C++使用对象来实现输入和输出，而C使用函数。C使用函数如printf和scanf来实现输入和输出，而C++使用对象如cout和cin来实现输入和输出。

6、C++语言的特点主要表现在：首先，它是C语言的继承，尽量兼容C语言，既保持了C语言的简洁和高效，可以像C语言那样进行结构化程序设计，同时也增强了C语言对类型的处理。其次加入了面向对象的特征，可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。

7、运算符重载的规则：除个别运算符（如：类属关系运算符“.” 成员指针运算符“.\*” 作用域运算符“：：” 三目运算符“？：”）不能被重载，其余都可以重载，只能重载C++中已经存在的运算符；重载后运算符优先级和结合性都不变；一般来说，重载的功能与原有功能类似，不能改变运算符的操作对象个数，同时至少要有一个操作对象是自定义类型。

8、纯虚函数与函数体为空的虚函数之间的异同：不同之处：（1）纯虚函数没有函数体，而空的虚函数的函数体为空。（2）纯虚函数所在的类是抽象类，不能直接进行实例化；而空的虚函数所在的类是可以实例化的。

相同点：纯虚函数与函数体为空的虚函数都可以派生出新的类，然后在新类中给出虚函数的实现，而且这种新的实现具有多态特征。

9、析构函数的功能：析构函数在对象的生存期即将结束的时刻被自动调用的，它用来完成对象被删除前的一些清理工作。（析构函数不接受任何参数）

10、纯虚函数是C++中一种特殊的虚函数，它没有函数体，只有函数声明，在函数名后面加上=0来表示。

11、多态：多态是指相同的操作可以作用于不同的对象，而具体执行的操作则取决于对象的类型和特性。

12、类的派生：派生是创建一个新类作为另一个已存在类的子类的过程。派生类继承了基类的属性，并可以添加新的属性和方法。

13、友元：友元是C++中的一个特殊机制，它可以实现在某些情况下，一个类的私有成员可以被其他类或者函数访问。友元可以是一个函数，也可以是一个类。

14、浅拷贝：浅拷贝是C++中一种简单的拷贝操作，它只是对对象的成员变量进行按位复制，如果成员变量是指针，那么拷贝后的对象和原对象会共享同一块内存空间。

15、标准流：标准流是C++中用于实现数据输入和输出的一种机制，它把数据传输的过程看作是数据从一个地方流动到另一个地方。C++中有三种标准流，分别是标准输入流、标准输出流、标准错误流。

16、多重继承：C++允许从多个类派生一个类，即一个派生类可以同时有多个基类。这称为多重继承。

17、类的继承 ：继承是创建一个新类，该新类可以继承一个或多个已存在的类的属性（成员变量和方法）。新类被称为子类或派生类。

18、命名空间：命名空间也称为名字空间，这是C++的一种机制，使用关键字namespace把大量有逻辑联系的程序实体组合在一个标识符下。

19、深拷贝：C++深拷贝是指在拷贝对象时，不仅复制对象的基本类型成员，还复制对象的指针指向的内容。这样，拷贝后的对象和原对象就不会共享同一块内存空间，修改其中一个不会影响另一个。

20、抽象类：抽象类是C++中一种特殊的类，它包含了至少一个纯虚函数，它不能创建对象，只能作为基类被继承。抽象类的作用是定义一些通用的接口，让派生类根据自己的需要来实现。

21、虚函数：虚函数是指一个类中希望重载的成员函数，当用一个基类指针或引用指向一个继承类对象的时候，调用一个虚函数时, 实际调用的是继承类的函数。

22、单继承：单继承是指一个派生类只有一个基类，从而继承了基类的属性和方法。

23、运算符重载:运算符重载就是给已有的运算符赋予多重含义，使同一个运算符作用于不同类型的数据时产生不同的行为。

24、函数重载：函数重载是指在程序的同一范围内声明几个功能类似的同名函数。

25、读取一行文字，将此行文字颠倒顺序输出。

#include <iostream>

#include<iomanip>

using namespace std;

int main() {

char ch[20];

cin.getline(ch,20);

for (int i = strlen(ch)-1; i >= 0; i--) {

cout << ch[i];

}

return 0;

}

26、设置标准输入重定向，文件data.txt中每一行保存一个整数，全部整数的个数和平均值：

#include <iostream>

#include<iomanip>

using namespace std;

int main() {

int x, count = 0;

double sum = 0;

FILE\* stream1;

freopen\_s(&stream1, ""data.txt"", ""r"", stdin);

while (cin >> x) {

sum += x;

count++;

}

cout << ""个数："" << count << "",平均数："" << sum / count << endl;

return 0;

}

27、输出一行，包含一个整数，表示N的2进制表示中1的个数。

#include<bits/stdc++.h>

using namespace std;

int q(int n){

int a[2]={0};

while(n!=0){

++a[n%2];

n=n/2;}

return a[1];

}

int main(){

int n;

cin>>n;

cout<<q(n);

return 0;

}

28、给出一个含有n（0 < n <= 1000）个整数的数组，请找出其中出现次数超过一半的数。

数组中的数大于-50且小于50。

输入格式：第一行包含一个整数n，表示数组大小；第二行包含n个整数，分别是数组中的每个元素，相邻两个元素之间用单个空格隔开。

输出格式：如果存在这样的数，输出这个数；否则输出no。

样例输入

3

1 2 2

样例输出

2

#include<bits/stdc++.h>

using namespace std;

int main(){

int n,t,b[1005]={0};

cin>>n;

int a[1005];

for(int i=0;i<n;i++)

{cin>>t;

a[i]=t+50;//加50是为了避免出现负数。

++b[a[i]];}//a[i]的值就会对应b中相应下标所对应的值次数+1。

for(int i=0;i<1005;i++){//最后找出大于一半的值，记得最后再减50。

if(b[i]>n/2)

{cout<<i-50;

return 0;}

}

cout<<""no"";

return 0;}

29、输入若干个整数，分别以16进制，8进制输出，然后以10个字符宽度输出，宽度不足左边补0。

#include <iostream>

#include<iomanip>

using namespace std;

int main() {

int x[5] = { 0 };

int i = 0;

cout << ""输入整数:\n"";

while (cin >> x[i]) {

cout << setbase(16) << x[i] << endl;

cout << setbase(8) << x[i] << endl;

cout << setw(10) << right << setfill('0') << x[i] << endl;

cout << ""\n"";

i++;

if (i >= sizeof(x) / sizeof(int)) {

break;

}

}

return 0;

}

30、编写一个程序，将从键盘输入的n个字符串保存在一个一维数组A中。在输入字符串之前，先输入n的值。要求，数组A需要动态申请空间，程序运行结束前再释放掉。

#include <iostream>

using namespace std;

int main(){

cout<<"请输入字符串个数:";

int n; //输入n

cin>>n;

char \*\*A; //定义一个指针数组

A=new char\*[n]; //动态申请空间

for(int i=0;i<n;i++){

A[i]=new char[20]; //每个指针指向一个字符串

cout<<"请输入第"<<i+1<<"个字符串:";

cin>>A[i]; //输入字符串

}

cout<<"输入的字符串为:"<<endl;

for(int i=0;i<n;i++){

cout<<A[i]<<endl; //输出字符串

}

for(int i=0;i<n;i++){

delete []A[i]; //释放空间

}

delete []A;

return 0;

}

31、实现斐波拉契数列。

//波那契数列其实就是这样一组数列：1、1、2、3、5、8、13、21、34、……

#include<iostream>

using namespace std;

int main()

{

long s1 = 1, s2 = 1;

//最大循环23，24就会有溢出

（1）【for (int i = 1; i <= 23; i++)】

{

//i变化一次，输出两个数字

//i取模于2，四个数字就换一次行！

cout<<"" "" << s1 << "" "" << s2;

（2）【if (i % 2 == 0)】

{

cout << endl;

}

s1 = s1 + s2;

s2 = s1 + s2;

}

return 0;

}

32、下面程序的功能是将一个 4×4 的数组进行逆时针旋转 90 度后输出，要求原始数组的数据随机输入，新数组以 4 行 4 列的方式输出。

#include<iostream>

using namespace std;

int main()

{

int a[4][4], b[4][4];

cout<<""请输入4x4的数组（16数字）："" << endl;

for (int i = 0; i < 4;i++)

{

for (int j = 0; j < 4; j++)

{

cin >> a[i][j];

（1）【b[3 - j][i] = a[i][j]】//i,j的转换关系

}

}

cout << ""逆时针旋转90度的结果如下："" << endl;;

（2）【for (int i = 0; i < 4; i++)】

{

for (int j = 0; j < 4; j++)

{

cout << b[i][j]<<"" "";

}

cout << ""\n"";

}

}

33、已知数组a中的元素已按由小到大顺序排列，以下程序的功能是将输入的一个数插入数组a中，插入后，数组a中的元素仍然由小到大顺序排列。

#include<iostream>

using namespace std;

int main()

{

int s[8] = {0,12,17,20,25,28,30};//s[0]为工作空间，从s[1]开始存放数据

//如果不对数组元素进行设定，由于s[i + 1] = s[i]，会报错C6386写入“s”时缓冲区溢出

//修复警告只需最少设置为8个元素

int x=0,i=0,j = 6;//j为数据个数

cout << "请输入您要排序的数字（0~30）：" << endl;

cin >> x;

s[0] = x;

i = j;

（1）【while (s[i] > x)】//从最后一个单元开始

{

s[i + 1] = s[i];

i--;

}//将比 x 大的数往后移动一个位置

s[i++] = x;

j++;//插入 x 后元素总个数增加

cout << "-------------\n";

（2）【for (i = 1; i <= j; i++)】

{

cout << s[i] <<"\t";

}

return 0;

}

34、编程打印直角杨辉三角形。

#include<iostream>

using namespace std;

int main()

{

int a[6][6];

（1）【for (int i = 0; i <= 5; i++)】 {

a[i][i] = 1;

a[i][0] = 1;

}

（2）【for (int i = 2; i <= 5; i++)】

for (int j = 1; j <= i-1; j++)

a[i][j] = a[i-1][j] + a[i-1][j-1];

for (int i = 0; i <=5; i++)

{

（3）【for (int j = 0; j <= i; j++)】

{

cout << a[i][j]<<"" "";

}

cout << ""\n"";

}

return 0;

}

35、通过键盘输入 3 名学生 4 门课程的成绩，分别求每个学生的平均成绩和每门课程的平均成绩。要求所有成绩均放入一个 4 行 5 列的数组中，输入时同一人数据间用空格, 不同人用回车其中最后一列和最后一行分别放每个学生的平均成绩、每门课程的平均成绩及全班平均分。

#include<iostream>

using namespace std;

int main()

{

（1）【float a[4][5]】

cout <<""请输入学生成绩（3x4格式）！""<< endl;

//输入成绩

for(int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

{

cin >> a[i][j];

}

}

//计算每个学生的平均成绩

for(int i=0;i<3;i++)

{

float sum01 = 0;

for (int j = 0; j < 4; j++)

{

（2）【sum01 += a[i][j]】

a[i][4] = sum01 / 4;

}

}

//计算每门课程的平均成绩

for (int j = 0; j < 5; j++)

{

float sum02 = 0;

for (int i = 0; i < 3; i++)

{

sum02 += a[i][j];

（3）【a[3][j] = sum02 / 3】

}

}

//输出结果，右下角的数据为全班的平均成绩

cout << ""结果如下："" << endl;

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 5;j++)

{

cout << a[i][j]<<"" "";

}

cout << ""\n"";

}

}

36、输入一个字符串，判断其是否为回文。回文字符串是指从左到右读和从右到左读完全相同的字符串。

#include<iostream>

#include<string>

using namespace std;

int main()

{

char s[100];

int i, j, n;

cout << ""输入字符串（最长100个字符）：\n"";

cin >> s;

n = strlen(s);

（1）【for (i = 0, j = n - 1; i < j; i++, j--)】

{

if (s[i] != s[j])

break;

}

（2）【if (i >= j)】

cout<<""恭喜你是回文串！\n"";

else

cout<<""抱歉不是回文串！\n"";

return 0;

}

37、打印九九乘法表。

#include<iostream>

using namespace std;

int main()

{

for (int i = 0; i < 9; i++)

{

（1）【for (int j = 0; j < i; j++)】

{

（2）【int result = i \* j】

cout<<i<<""\*"" <<j<<""=""<<result<<"" "";

}

cout << ""\n"";

}

return 0;

}

38、冒泡排序，从小到大，排序后结果输出到屏幕。

#include<iostream>

using namespace std;

int main()

{

int a[100], i=0, j=0, t=0, n=0;

printf(""请输入你要排序的总个数："");

cin>>n;//输入一个数n，表示接下来有n个数

cout<<""请输入具体的整数：\n"";//循环读入n个数到数组a中

for (i = 1; i <= n; i++)

{

cin>>a[i];

}

//冒泡排序的核心部分

（1）【for (i = 1; i <= n - i; i++)】//n个数排序，只用进行n-1趟

{

（2） 【for (j = 1; j <= n - i; j++)】 //从第1位开始比较直到最后一个尚未归位的数

{

if (a[j] > a[j + 1])//将大的数往后放

{

t = a[j];

a[j] = a[j + 1];

a[j + 1] = t;

}

}

}

cout << ""冒泡排序的结果为：\n"";

for (i = 1; i <= n; i++)//i++输出法

{

cout << a[i]<<"" "";

}

return 0;

}