**1.哪个选项和show函数重载**

class Demo{

void show(int a,int b,float c){}

}

A.void show(int a,float c,int b){}//yes

B,void show(int a,int b,float c){}//一模一样。不可以出现在同一个类中。

C.int show(int a,float c,int b){return a;}//yes。

D.int show(int a,float c){return a;}//yes

**2.方法重载(overload)必须满足\_\_\_\_\_\_\_\_。**

A. 在不同class中定义的方法 B.在同一类型中定义的方法

C. 方法名必须相同 D.返回类型必须相同

E. 参数一定不同 F.参数可以相同

**3.构造器Constructor是否可被override**

答：构造器Constructor不能被继承，因此不能重写Override，但可以被重载Overload

**4.写出结果。(关于参数传递)**

public class Test{

public static void leftshift(int i, int j){

i+=j;

}

public static void main(String args[]){

int i = 4, j = 2;

leftshift(i, j);

System.out.println(i);

}

}

//4 和leftShift函数没关系。

**5.写出结果。(关于参数传递)**

public class Demo{

public static void main(String[] args){

int[] a=new int[1];

modify(a);

System.out.println(a[0]); //

}

public static void modify(int[] a){

a[0]++;

}

}

**6.写出结果**

public class TestA {

int i ;

void change(int i){

i++;

System.out.println(i);

}

void change1(TestA t){

t.i++;

System.out.println(t.i);

}

public static void main(String[] args) {

TestA ta = new TestA();

System.out.println(ta.i); //

ta.change(ta.i);//

System.out.println(ta.i); //

ta.change1(ta); //

System.out.println(ta.i);//

}

}

**7.写出结果。（关于参数传递）**

class Value{

int i = 15;

}

class Test{

public static void main(String argv[]) {

Test t = new Test();

t.first();

}

public void first() {

int i = 5;

Value v = new Value();

v.i = 25;

second(v, i);

System.out.println(v.i);

}

public void second(Value v, int i) {

i = 0;

v.i = 20;

Value val = new Value();

v = val;

System.out.print(v.i + " " + i);

}

}

A.15 0 20

B.15 0 15

C.20 0 20

D.0 15 20

A is correct.

**8.写出结果。（关于参数传递）**

public class TestArgsValue {

public static void main(String[] args) {

int i = 10;

TestArgsValue tv = new TestArgsValue();

tv.method1(i);

System.out.println("i=" + i);

System.out.println();

Demo d = new Demo();

System.out.println(d);

tv.method2(d);

System.out.println("d.i = " + d.i);

}

public void method1(int i){

System.out.println("i=" + i++);

}

public void method2(Demo d){

System.out.println(d);

System.out.println("d.i : " + d.i++);

}

}

class Demo{

int i = 5;

}

**9.当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，**

**那么这里到底是值传递还是引用传递?**

答：是值传递。Java 编程语言只有值传递参数。当一个对象实例作为一个参数被传递到方法中时，

参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的

**10.What is the result when you compile and run the following code?**

public class Test{

public void method(){

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

System.out.print(i);

}

System.out.print(i);

}

}

A. 0122

B. 0123

C. compile error

D. none of these

**11.写出输出结果**。

class Demo{

public static void main(String[] args){

show(0);

show(1);

}

public static void show(int i){

switch(i){

default:

i+=2;

case 1:

i+=1;

case 4:

i+=8;

case 2:

i+=4;

}

System.out.println("i="+i);

}

}

**12.写出输出结果**

class Demo{

public static void main(String[] args){

int x = 1;

for(show('a'); show('b') && x<3; show('c')){

show('d');

x++;

}

}

public static boolean show(char ch){

System.out.print(ch);

return true;

}

}

//答案：abdcbdcb

**13.设计一个立方体类Box，定义三个属性，分别是长，宽，高。定义二个方法，分别计算并输出立方体的体积和表面积**

**14.请定义一个交通工具(Vehicle)的类，其中有:属性：速度(speed)，体积(size)等等方法：移动(move(int s))，设置速度(setSpeed(int speed))，加速speedUp(),减速speedDown()等等.最后在测试类Vehicle中的main()中实例化一个交通工具对象，并通过方法给它初始化speed,size的值，并且通过打印出来。另外，调用加速，减速的方法对速度进行改变。调用 move方法输出移动距离**

**15.定义一个Hero类属性有 power,name，分别代表体力值和英雄的名子，体力值默认为100；方法有 1.void go(); //行走的方法，如果体力值为0，则输出不能行走，此英雄已死亡的信息 2.void eat(int n); //吃的方法，参数是补充的血量，将 n的值加到属性power中，power的值最大为100，3.void hurt();//每受到一次伤害，体力值－10，体力值最小不能小于0**

**16.定义一个计算器；它的功能有加,减,乘,除,累加,阶乘求平方,求次方,判断一个数是否为素数；boolean（int）,并写测试类来测试这个方法**

**17.编写Java程序，用于显示人的姓名和年龄**。

定义一个人类（Person），该类中应该有两个私有属性，姓名（name）和年龄（age）。定义构造方法，用来初始化数据成员。再定义显示（display）方法，将姓名和年龄打印出来。

在main方法中创建人类的实例，然后将信息显示

**18.定义一个类，该类有一个私有成员变量，通过构造方法将其进行赋初值，并提供该成员的getXXX()和setXXX()方法**

提示：假设有private String name;则有

public void setName(String name){

this.name = name;

}

public String getName(){

return this.name;

}

**19.构造方法与重载**

为“无名的粉”写一个类：class WuMingFen 要求：

1.有三个属性：面码:String theMa 粉的份量(两)：x

是否带汤：boolean likeSoup

2.写一个构造方法，以便于简化初始化过程，如：

WuMingFen f1 = new WuMingFen("牛肉",3,true);

3.重载构造方法，使得初始化过程可以多样化：

WuMingFen f2 = new WuMingFen("牛肉",2);

4.如何使得下列语句构造出来的粉对象是酸辣面码、2两、带汤的？

WuMingFen f3 = new WuMingFen();

5.写一个普通方法：check()，用于查看粉是否符合要求。即：将对象的三个属性打印在控制台上。

**20.构造方法的重载**：

在程序中，经常要对时间进行操作，但是并没有时间类型的数据。那么，我们可以自己实现一个时间类，来满足程序中的需要。

定义名为MyTime的类，其中应有三个整型成员：时（hour），分（minute），秒（second）， 为MyTime类定义构造方法，以方便创建对象时初始化成员变量。

再定义diaplay方法，用于将时间信息打印出来。

为MyTime类添加以下方法：

addSecond(int sec)

addMinute(int min)

addHour(int hou)

subSecond(int sec)

subMinute(int min)

subHour(int hou)

分别对时、分、秒进行加减运算。

**21.构造方法与重载**

定义一个牛肉面的类(Noodle),它的属性有，

牛肉面宽度 width

尺寸：size (大碗还是小碗) 大碗8元，小碗6元

是否加肉：beef :加肉+4元

加蛋的数量：eggs :每个1元；

定义构造方法来根据不同的条件创建不同的牛肉面

Noodle(){

//不加肉，大碗，不加蛋，中宽；

}

Noodle(String width,int size)

Noodle(String width,int size,boolean beef);

Noodle(String width,int size,boolean beef,int eggs);

再定义一个方法，用来显示当前牛肉面的信息，并显示金额；

void showNoodleInfo();

**22.构造方法与重载、包**

编写Addition类，该类中应包含一组实现两数相加运算的重载方法。

实现加法运算的方法，应接受两个参数（即加数和被加数），方法将两个参数进行加法运算后，返回相加结果。考虑可能针对不同的数据类型进行计算，重载一组方法，包括整型、长整型、浮点型、双精度浮点型、还有字符串。 在main方法中创建Addition类的实例，分别调用重载方法测试其效果。 应将Addition类打入到包中，以自己名字的拼音为包命名。

**23.构造方法与重载**

定义一个网络用户类，要处理的信息有用户ID、用户密码、email地址。在建立类的实例时，把以上三个信息都作为构造函数的参数输入，其中用户ID和用户密码时必须的，缺省的email地址是用户ID加上字符串"@gameschool.com"

**24.构造方法与重载**

建立一个汽车类，包括轮胎个数，汽车颜色，车身重量等属性。并通过不同的构造方法创建事例。

至少要求：汽车能够加速，减速，停车。

要求：命名规范，代码体现层次，有友好的操作提示。

**25.构造方法与重载**

创建一个类，为该类定义三个构造函数，分别执行下列操作：

1、传递两个整数值并输出其中较大的一个值

2、传递三个double值并计算出其乘积，将结果输出

3、传递两个字符串值并将它们连接到一起后输出

4、在main方法中测试构造函数的调用

**26.类的综合练习**

1、写一个名为Account的类模拟账户。该类的属性和方法如下图所示。该类包括的属性：账号id，余额balance，年利率annualInterestRate；包含的方法：访问器方法（getter和setter方法），取款方法withdraw()，存款方法deposit()。

|  |
| --- |
| Account |
| private int id  private double balance  private double annualInterestRate |
| public Account (int id, double balance, double annualInterestRate ) |
| public int getId()  public double getBalance()  public double getAnnualInterestRate()  public void setId( int id)  public void setBalance(double balance)  public void setAnnualInterestRate(double annualInterestRate)  public void withdraw (double amount)  public void deposit (double amount) |

**提示：**在提款方法withdraw中，需要判断用户余额是否能够满足提款数额的要求，如果不能，应给出提示。

2. 创建Customer类。

|  |
| --- |
| Customer |
| private String firstName  private String lastName  private Account account |
| public Customer(String f,String l) |
| public String getFirstName()  public String getLastName()  public Account getAccount()  public void setAccount(Account account) |

a. 声明三个私有对象属性：firstName、lastName和account。

b. 声明一个公有构造器，这个构造器带有两个代表对象属性的参数（f和l）

c. 声明两个公有存取器来访问该对象属性，方法getFirstName和getLastName返回相应的属性。

d. 声明setAccount 方法来对account属性赋值。

e. 声明getAccount 方法以获取account属性。

3.写一个测试程序。

（1） 创建一个Customer ，名字叫 Jane Smith, 他有一个账号为1000，余额为2000元，年利率为 1.23％ 的账户。

（2） 对Jane Smith操作。

存入 100 元，再取出960元。再取出2000元。

打印出Jane Smith 的基本信息

成功存入 ：100.0

成功取出：960.0

余额不足，取款失败

Customer [Smith, Jane] has a account: id is 1000, annualInterestRate is 1.23％, balance is 1140.0

**27.题目要求：**

（1）定义一个Circle类，包含一个double型的radius属性代表圆的半径，一个findArea()方法返回圆的面积。

（2）定义一个类PassObject，在类中定义一个方法printAreas()，该方法的定义如下：

**public void printAreas(Cirlce c, int time)**

在printAreas方法中打印输出1到time之间的每个整数半径值，以及对应的面积。例如，times为5，则输出半径1，2，3，4，5，以及对应的圆面积。

在main方法中调用printAreas()方法，调用完毕后输出当前半径值。