

## Day 1

1. 如果 `int x=20, y=5`, 则语句 `System.out.println(x+y + ""+(x+y)+y);` 的输出结果是 ( )。

A. 2530

B. 55

C. 2052055

D. 25255

2. 在 Java 中, 在同一包内, 类 Cat 里面有个公共方法 `sleep()`, 该方法前有 `static` 修饰, 则可以直接用 `Cat.sleep()`。( )

A. 正确

B. 错误

3. 给定以下方法声明, 调用执行 `mystery(1234)` 的输出结果是 ( )。

```
//precondition: x >= 0
public void mystery (int x) {
    System.out.print(x % 10);
    if ((x / 10) != 0){
        mystery(x / 10);
    }
    System.out.print(x % 10);
}
```

A. 1441

B. 43211234

C. 3443

D. 12344321

4. 实现函数 `ToLowerCase()`, 该函数接收一个字符串参数 `str`, 并将该字符串中的大写字母转换成小写字母, 之后返回新的字符串。OJ 链接-转换成小写字母

## 转换成小写字母

```
1 class Solution {
2     public String toLowerCase(String str) {
3
4     }
5 }
```

5. 给定一个数组, 将数组中的元素向右移动 `k` 个位置, 其中 `k` 是非负数。OJ 链接-旋转数组

```
1 class Solution {
2     public void rotate(int[] nums, int k) {
3
4     }
5 }
```

## Day 2

1. 如果类的方法没有返回值, 该方法的返回类型是 ( )。

- A. `void` B. `null` C. `abstract` D. `default`

2. `Java Application`(Java应用程序) 源程序文件编译后的字节码文件的扩展名是 ( )。

- A. `.java` B. `.class` C. `.exe` D. `.jar`

3. 关于继承的描述正确的是 ( )。

- A. 函数的出口应该尽可能少, 最好只有一个出口  
B. 为了防止程序中的内存泄漏, 应该不使用动态内存分配  
C. 在函数实现中应该少使用全局变量  
D. 函数的功能应该单一

4. 给定一个数组 `nums` 和一个值 `val`, 你需要原地移除所有数值等于 `val` 的元素, 返回移除后数组的新长度。不要使用额外的数组空间, 你必须在原地修改输入数组并在使用  $O(1)$  额外空间的条件下完成。[OJ 链接-移除元素](#)

```
1 class Solution {  
2     public int removeElement(int[] nums, int val) {  
3  
4     }  
5 }
```

5. 给定一个排序数组和一个目标值, 在数组中找到目标值, 并返回其索引。如果目标值不存在于数组中, 返回它将会被按顺序插入的位置, 你可以假设数组中无重复元素。[OJ 链接-搜索插入位置](#)

```
1 class Solution {  
2     public int searchInsert(int[] nums, int target) {  
3  
4     }  
5 }
```

## Day 3

1. 以下代码的循环次数是 ( )。

```
public class Test {  
    public static void main(String[] args) {  
        int i = 7;  
        do {  
            System.out.println(--i);  
            --i;  
        } while (i != 0);  
        System.out.println(i);  
    }  
}
```

A. 0                                      B. 1                                      C. 7                                      D. 无限次

2. 下列选项中属于面向对象设计方法的主要特征是 ( )。

A. 继承                                      B. 自顶向下                                      C. 模块化                                      D. 逐步求精

3. 下面的方法，当输入为 2 的时候返回值是多少 ( )。

```
public static int getValue(int i) {  
    int result = 0;  
    switch (i) {  
        case 1:  
            result = result + i;  
        case 2:  
            result = result + i * 2;  
        case 3:  
            result = result + i * 3;  
    }  
    return result;  
}
```

A. 0                                      B. 2                                      C. 4                                      D. 10

4. 赎金信。 [OJ 链接-赎金信](#)

## 赎金信

```
1 class Solution {  
2     public boolean canConstruct(String ransomnote, String magazine){  
3  
4     }  
5 }
```

5. 判断一个整数是否是回文数。回文数是指正序（从左到右）和倒叙（从右向左）读都是一样的整数。 [OJ 链接-回文数](#)

## 回文数

```
1 class Solution {  
2     public boolean isPalindrome(int x){  
3  
4     }  
5 }
```

## Day 4

1. 下面有关 `java final` 的基本规则, 描述错误的是 ( )。
- A. `final` 修饰的类不能被继承
  - B. `final` 修饰的成员变量只允许赋值一次, 且只能在类方法赋值
  - C. `final` 修饰的局部变量即为常量, 只能赋值一次
  - D. `final` 修饰的方法不允许被子类覆盖
2. 选项中哪一行代码可以替换 `//add code here` 而不产生编译错误 ( )。
- A. `public abstract void method(int a);`
  - B. `consInt = constInt + 5;`
  - C. `public int method;`
  - D. `public abstract void anotherMethod(){}`
3. 第三行将输出什么 ( )。

```
1 public class SwitchTest{
2     public static void main(String[] args) {
3         System.out.println("value=" + switchit(4));
4     }
5     public static int switchit(int x) {
6         int j = 1;
7         switch (x) {
8             case 1:j++;
9             case 2:j++;
10            case 3:j++;
11            case 4:j++;
12            case 5:j++;
13            default:j++;
14        }
15        return j+x;
16    }
17 }
```

- A. `value=6;`                      B. `value=8;`                      C. `value=3;`                      D. `value=5;`

4. 给定一个仅包含大小写字母和空格 ' ' 的字符串, 返回其最后一个单词的长度。如果不存在最后一个单词, 请返回 0。 [OJ 链接-最后一个单词的长度](#)

## 最后一个单词的长度

```
1 class Solution {
2     public int lengthOfLastWord(String s) {
3
4     }
5 }
```

5.给你两个有序整数数组 `nums1` 和 `nums2`，请你将 `nums2` 合并到 `nums1` 中，使 `nums1` 成为一个有序数组。  
[OJ 链接-合并两个有序数组](#)

## 合并两个有序数组

```
1 class Solution {  
2     public void merge(int[] nums1, int m, int[] nums2, int n) {  
3  
4     }  
5 }
```

## Day 5

1. 下列 Java 程序输出的结果为 ( )。

```
public class Example{
    String string = new String("hello");
    char[] chars = {'a','b'};
    public static void main(String args[]){
        Example ex = new Example();
        ex.change(ex.string, ex.ch);
        System.out.print(ex.str + "and");
        System.out.print(ex.ch);
    }
    public void change(String string, char ch[]){
        string = "test ok";
        ch[0] = 'c';
    }
}
```

- A. hello and ab      B. hello and cb      C. hello and a      D. test ok and ab

2. *transient* 变量和下面哪一项有关 ( )。

- A. Cloneable      B. Serializable      C. Runnable      D. Comparable

3. 已知有下列 *Test* 类的说明, 在该类的 *main* 方法内, 则下列那个语句是正确的 ( )。

```
public class Test {
    private float f = 1.0f;
    int m = 12;
    static int n = 1;
    public static void main (String args[]){
        Test t = new Test();
    }
}
```

- A. t.f;      B. this.n;      C. Test.m;      D. Test.f;

4. 给定一个整数数组, 判断是否存在重复元素。如果存在一值在数组中出现至少两次, 函数返回 **true** 。如果数组中每个元素都不相同, 则返回 **false** 。 [OJ 链接-存在重复元素](#)

```
1 class Solution {
2     public boolean containsDuplicate(int[] nums) {
3
4     }
5 }
```

5. 你的朋友正在使用键盘输入他的名字 *name*。偶尔, 在键入字符 *c* 时, 按键可能会被长按, 而字符可能被输入 1 次或多次。你将会检查键盘输入的字符 **typed**。如果它对应的可能是你的朋友的名字 (其中一些

字符可能被长按), 那么就返回 `True`。 [OJ 链接-长按键入](#)

```
1 class Solution {  
2     public boolean containsDuplicate(int[] nums) {  
3  
4     }  
5 }
```



## Day 6

1. 下面关于 Java 的垃圾回收机制正确的是 ( )。

- A. 当调用 "System.gc()" 来强制回收时, 系统会立即回收垃圾
- B. 垃圾回收不能确定具体的回收时间
- C. 程序可明确地标识某个局部变量的引用不再被使用
- D. 程序可以显式地立即释放对象占有的内存

2. 以下会产生信息丢失的类型转换是 ( )。

- A. `float a = 10;`
- B. `int a = (int)8846.0 ;`
- C. `byte a = 10; int b = a;`
- D. `double = 100`

3. 面向对象方法的多态性指的是 ( )。

- A. 一个类可以派生出多个特殊类
- B. 一个对象在不同的运行环境中可以有不同的变体
- C. 针对一消息, 不同的对象可以以适合自身的方式加以响应
- D. 一个对象可以由多个其他对象组合而成的

4. 给定一个按非递减顺序排序的整数数组 `A`, 返回每个数字的平方组成的新数组, 要求新数组也按非递减顺序排序。(注意: 非递减顺序即递增, 要注意原数组里的负数) [OJ 链接-有序数组的平方](#)

## 有序数组的平方

```
1 class Solution {  
2     public int[] sortedSquares(int[] nums) {  
3  
4     }  
5 }
```

5. 给定一个字符串 `S`, 返回“反转后的”字符串, 其中不是字母的字符都保留在原地, 而所有字母的位置发生反转。[OJ 链接-仅仅反转字母](#)

## 仅仅反转字母

```
1 class Solution {  
2     public String reverseOnlyLetters(String S) {  
3  
4     }  
5 }
```