**1**、一个应用程序中有多条并发执行的线索，每条线索都被称作一个\_\_\_\_\_\_，它们会交替执行，彼此间可以进行\_ \_\_\_\_\_。

2、在实现多线程的程序时有两种方式，一是通过继承\_\_Thread\_\_\_\_类，二是通过实现\_\_Runnable\_\_\_\_接口。

**3**、yield()方法只能让**相同优先级或者更高优先级**、处于\_\_\_\_\_\_状态的线程获得运行的机会。

**4**、在Java语言中，同步方法需要用到关键字\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，对于同步方法而言无需指定同步锁，它的同步锁是方法所在的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，也就是\_\_\_\_\_\_(关键字)。

5、在多任务系统中，每个独立执行的程序称之为\_\_\_\_\_\_\_\_\_，也就是“正在运行的程序”。

6、线程的整个生命周期分为五个阶段，分别是\_\_\_\_\_\_、\_\_\_\_\_\_、\_\_\_\_\_\_、\_\_\_\_\_\_、和\_\_\_\_\_。

7、线程的优先级用1~10之间的整数来表示，其中\_\_10\_\_\_代表优先级最高，\_\_1\_\_\_代表优先级最低，默认级别是\_\_\_\_\_\_\_\_\_。

8、在Thread类中，提供了一个start()方法，该方法用于\_\_\_\_\_\_\_\_\_\_，当新线程启动后，系统会自动调用\_run()\_\_\_\_\_方法。

**9**、要想解决线程间的通信问题，可以使用\_\_\_\_\_\_、\_\_\_\_\_\_、\_\_\_\_\_\_方法。

二、判断题

1、当我们创建一个线程对象时，该对象表示的线程就立即开始运行。

2、如果前台线程全部死亡，后台线程也会自动死亡。

3、同步代码块中的锁对象可以是任意类型的对象。

4、静态方法不能使用synchronized关键字来修饰。

5、线程结束等待或者阻塞状态后，会进入运行状态。

三、选择题

**1**、Thread类位于下列哪个包中？b

A、java.io B、java.lang C、java.util D、java.awt

2、关于线程的创建过程，下面四种说法正确的有哪些？(多选)

A、定义Thread类的子类，重写Thread类的run()方法，创建该子类的实例对象，调用对象的start()方法

B、定义Thread类的子类，重写Thread类的run()方法，创建该子类的实例对象，调用对象的run()方法

C、定义一个实现Runnable 接口的类并实现run()方法，创建该类实例对象，将其作为参数传递给Thread类的构造方法来创建Thread对象，调用Thread对象的start()方法

D、定义一个实现Runnable 接口的类并实现run()方法，创建该类对象，然后调用run()方法

**3**、对于通过实现Runnable接口创建线程，下面说法正确的有哪些？(多选)

A、适合多个相同程序代码的线程去处理同一个资源的情况

B、把线程同程序代码、数据有效的分离，很好的体现了面向对象的设计思想

C、可以避免由于Java的单继承带来的局限性

D、编写简单，可以不通过Thread类直接创建线程

**4**、对于线程的生命周期，下面四种说法正确的有哪些？(多选)   
　 A、调用了线程的start()方法，该线程就进入运行状态

B、线程的run()方法运行结束或被未catch的InterruptedException等异常终结，那么该线程进入死亡状态

C、线程进入死亡状态，但是该线程对象仍然是一个Thread对象，在没有被垃圾回收器回收之前仍可以像引用其它对象一样引用它

D、线程进入死亡状态后，调用它的start()方法仍然可以重新启动

5、对于死锁的描述，下面四种说法正确有哪些？(多选)

A、当两个线程互相等待对方释放同步锁时会发生死锁

B、Java虚拟机没有检测和处理死锁的措施

C、一旦出现死锁，程序会发生异常

D、处于死锁状态的线程处于阻塞状态，无法继续运行

**6**、下面四个选项中，哪些是线程进入阻塞状态的原因？(多选)

A、线程试图获取某个对象的同步锁，而该锁被其它线程持有

B、线程调用了另一个线程的join()方法

C、当线程调用了一个阻塞式的IO方法

D、线程调用了setDaemon(boolean b)方法

7、线程调用sleep()方法后，该线程将进入以下哪种状态？

A、就绪状态 B、运行状态

C、阻塞状态 D、死亡状态

8、在以下哪种情况下，线程进入就绪状态？

A、线程调用了sleep()方法时 B、线程调用了join()方法(终止)

C、线程调用了yield()方法时 D、线程调用了notify()方法

9、下面四个选项中，对线程同步的目的描述正确的有哪些？(多选)

A、锁定资源，使同一时刻只有一个线程去访问它，防止多个线程操作同一个资源引发错误

B、提高线程的执行效率

C、让线程独占一个资源

D、让多个线程同时使用一个资源

10、对于wait()方法，下面说法正确的是？(多选)

A、wait()方法的调用者是同步锁对象

B、wait()方法使线程进入等待状态

C、调用同一锁对象的notify()或notifyAll()方法可以唤醒调用wait()方法等待的线程

D、调用wait()方法的线程会释放同步锁对象

四、程序分析题

阅读下面的程序，分析代码是否能编译通过，如果能编译通过，请列出运行的结果。如果不能编译通过，请说明原因。

class RunHandler {

public void run(){

System.out.println("run");

}

}

public class Test {

public static void main(String [] args) {

Thread t = new Thread(new RunHandler());

t.start();

}

}

**2.代码二**：

public class A extends Thread{

**protected** void run() {

System.out.println("this is run()");

}

public static void main(String[] args) {

A a = new A();

a.start();

}

}

3.代码三：Exception**(程序不能编译通过，因为同步方法中调用wait()方法的对象必须为同步锁对象)**

public class Test{

private Object obj=new Object();

public synchronized void a(){

try {

obj.wait()；改**this.wait()**

System.out.println("waiting");

} catch (InterruptedException e) {

System.out.println("Exception");

}

}

public static void main(String[] args) {

new Test().a();

}

}

4.为了使下面的程序能够输出“Thread is running”，请在在空格处填上相应的代码。

public class B implements Runnable {

public static void main(String[] args) {

Thread t = new Thread(new B());

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //此处填空

}

public void run() {

System.out.println("Thread is running");

}

public void go() {

start(1);

}

public void start(int i) {

}

}

五、简答题

1. java中有几种方法可以实现一个线程？请举例说明。
2. sleep() 和 wait() 有什么区别?

调用sleep()方法，正在执行的线程主动让出CPU去执行其他线程，在sleep()方法指定的时间过后，CPU才会回到这个线程上继续往下执行，如果当前线程进入了同步锁，sleep()方法并不会释放锁，即使当前线程使用sleep()方法让出了CPU，但其它被同步锁挡住了的线程也无法得到执行。wait()在一个已经进入了同步锁的线程内进行调用，让当前线程暂时让出同步锁，以便其它正在等待此锁的线程可以得到同步锁并运行。当其它线程调用了notify()方法后，调用wait()方法的线程就会解除wait状态，当再次获得同步锁后，程序可以继续向下执行。

六、编程题

请按照题目的要求编写程序并给出运行结果。

1、通过继承Thread类的方式创建两个线程，在Thread构造方法中指定线程的名字，并将这两个线程的名字打印出来。

2、通过实现Runnable接口的方式创建一个新线程，要求main线程打印100此次“main”，新线程打印50次“new”。

3、模拟三个老师同时分发80份学习笔记，每个老师相当于一个线程。

4、编写10个线程，第一个线程从1加到10，第二个线程从11加到20...第十个线程从91加到100，最后再把十个线程结果相加。(考虑把和对应的信息做成临界资源)