

```

Q>
class Counter implements Runnable {
    private static int i = 3;

    @Override
    public void run() {
        System.out.print(i--);
    }
}

public class Test {
    public static void main(String[] args) {
        Thread t1 = new Thread(new Counter());
        Thread t2 = new Thread(new Counter());
        Thread t3 = new Thread(new Counter());
        Thread[] threads = {t1, t2, t3};
        for(Thread thread : threads) {
            thread.start();
        }
    }
}

```

What will be the result of compiling and executing Test class?

- A. Always print 321
- B. prints 321 but order can be different
- C. It always print 210
- D. It always print 210 but order can be different
- E. None of the other options

Answer: B

2.  
Which method must be defined by a class implementing the java.lang.Runnable interface?

- 1. void run()
- 2. public void run()
- 3. public void start()
- 4. None of these.

Answer: 2

3.  
Which of the following are the valid constructors of Thread class?

- i. Thread(Runnable r, String name)
- ii. Thread()
- iii. Thread(int priority)
- iv. Thread(Runnable r, ThreadGroup g)
- v. Thread(Runnable r, int priority)

Option

- 1. ii and iv
- 2. ii and v
- 3. i and iii
- 4. i and ii
- 5. None

Answer: 4

4.

What will be the output of the following Java code snippet?

```
class ThreadDemo extends Thread {
    final StringBuffer sb1 = new StringBuffer();
    final StringBuffer sb2 = new StringBuffer();
    public static void main(String args[]) {
        final ThreadDemo h = new ThreadDemo();
        new Thread() {
            public void run() {
                synchronized(this) {
                    h.sb1.append("Java");
                    h.sb2.append("Thread");
                    System.out.println(h.sb1);
                    System.out.println(h.sb2);
                }
            }
        }.start();

        new Thread() {
            public void run(){
                synchronized(this) {
                    h.sb1.append("Multithreading");
                    h.sb2.append("Example");
                    System.out.println(h.sb2);
                    System.out.println(h.sb1);
                }
            }
        }.start();
    }
}
```

Choose the option

1. Multithreading Example java Thread.
2. None of these.
3. Java Thread Multithreading Example.
4. The order in which text will be printed cannot be determined.
5. Java Thread ThreadExample JavaMultithreading.

Answer : 5

5.

What will be the output of the below Java program?

```
class MyThread extends Thread {
    MyThread() {}
    MyThread(Runnable r) {super(r); }
    public void run() {
        System.out.print("Inside Thread ");
    }
}

class RunnableDemo implements Runnable {
    @Override
    public void run() {
        System.out.print(" Inside Runnable");
    }
}

class ThreadDemo {
    public static void main(String[] args) {
        new MyThread().start();
        new MyThread(new RunnableDemo()).start();
    }
}
```

- ```
    }  
}
```
1. None of these.
  2. Prints "Inside Thread Inside Runnable".
  3. Does not compile.
  4. Prints "Inside Thread Inside Thread".
  5. Throws exception at runtime.

Answer: 4