Threads Test

1. Show different ways to create a thread. Give one example of each. (extends and implements way)
2. What is output of following code.

```
public class JavaDaemonThread1
 {
        public static void main(String[] args) throws InterruptedException
         {
                Thread dt = new Thread(new SampleThread(), "dt");
                dt.setDaemon(false);
                dt.start();
                Thread.sleep(3000);
                System.out.println("Finishing program");
        }
}

class SampleThread implements Runnable {
        @Override
        public void run() {
                while (true) {
                System.out.println("Processing thread");
                }
        }
}
```

3. State true or false with one liner explanation : Thread can be created by extending runnable interface.
4. State true or false with one liner explanation : Thread class contains wait() method.
5. State true or false with one liner explanation : Object can be shared when thread is created by extending thread class. If false also show a way where using extends Thread we can share object. [3M]
6. State true or false with one liner explanation : Sleeping thread can be woke up by using notify() method.
7. State true or false with one liner explanation : Instance variables are thread safe.
8. wait method has to be called in run method or in main method
9. join method has to be called in run method or in main method
10. notify method has to be called in run method or in main method
11. 1 liner difference between sleep and wait
12. How to set thread priority with one example
13. I want to print following output using 2 threads run method:

```
10
20
30
10
20
30
```

Write the code for it [2M]

14. If i sleep a thread and call notify what will happen?
15. What are 5 states of thread life cycle. only draw diagram not detailed theory.
16. What is output of following program.

```
public class TestThread_Extends {
        public static void main(String args[]) throws Exception {
        Thread t = Thread.currentThread();
        ExtendsThread1 tc1 = new ExtendsThread1();
        ExtendsThread1 tc2 = new ExtendsThread1();
        tc1.setName("tc1");
        tc2.setName("tc2");
        tc2.start();
        tc2.join();
        tc1.start();
        System.out.println("Ending Main !!");
        }
}
class ExtendsThread1 extends Thread {
        private int counter = 0;
        public void run() {
                counter++;
                Thread t = Thread.currentThread();
                System.out.println("Thread in context : "+t.getName());
                System.out.println("ExtendsThread : Counter : " + counter);
                System.out.println("end");
                }
}
```