Rohan Nyati 500075940 148

Batch 5

Date ______ Page ___

Assignment - 2

Q.1) Inner Classes aka Non-Stack Classes have access to other members of the top class, even if they are declared ~~for~~ private while static nested classes do not have access to other members of the top class.

Q.2) Anonymous Classes enable ~~you~~ us to make ~~y~~our code more concise. They enable ~~you~~ us to declare and instantiate a class at the same time. They are like local classes except that they do not have a ~~one~~ name. we use them if we need to use a local class only once.

Syntax:

```
Test t = new Test()
{
      // data members and methods
    public void test-method()
        {
           -- --
        }
   };
```

Scanned with CamScanner

Q.3) 1) New → The thread is in new state if you create an instance of Thread class but before the invocation of Start () method.

2) Runnable → The Thread is in runnable state after invocation of start() method, but the thread scheduler has not selected it to be the running thread.

3) Running → If the thread scheduler has selected it.

4) Non-Runnable (Blocked) → This is the state when the thread is still alive but is currently not eligible to run.

5) Terminated → A thread is in terminated or dead state when its run() method exits.

Q.4) If your code is executing in a multi threaded environment, we need Sync. for objects, which are shared among multiple threads, to avoid any corruption of state or any kind of unexpected behavior. Synchronization in java will only be needed if a shared object is mutable.

So there is a need to synchronize the action of multiple threads and make sure that only one thread can access the resources at a given point of time. This is implemented using a concept called monitors.

Q.5) Thread group creates a group of threads. It offers a convinient way to manage groups of threads as a unit. This is particularly valuable in situation in which we want to suspend ~~&~~ and resume a number of related threads. The thread group from a tree in which every thread group ~~except~~ except the initial thread group has a parent.

Constructors:

1) public ThreadGroup (String name)

2) public ThreadGroup (ThreadGroup parent, String name)

Scanned with CamScanner