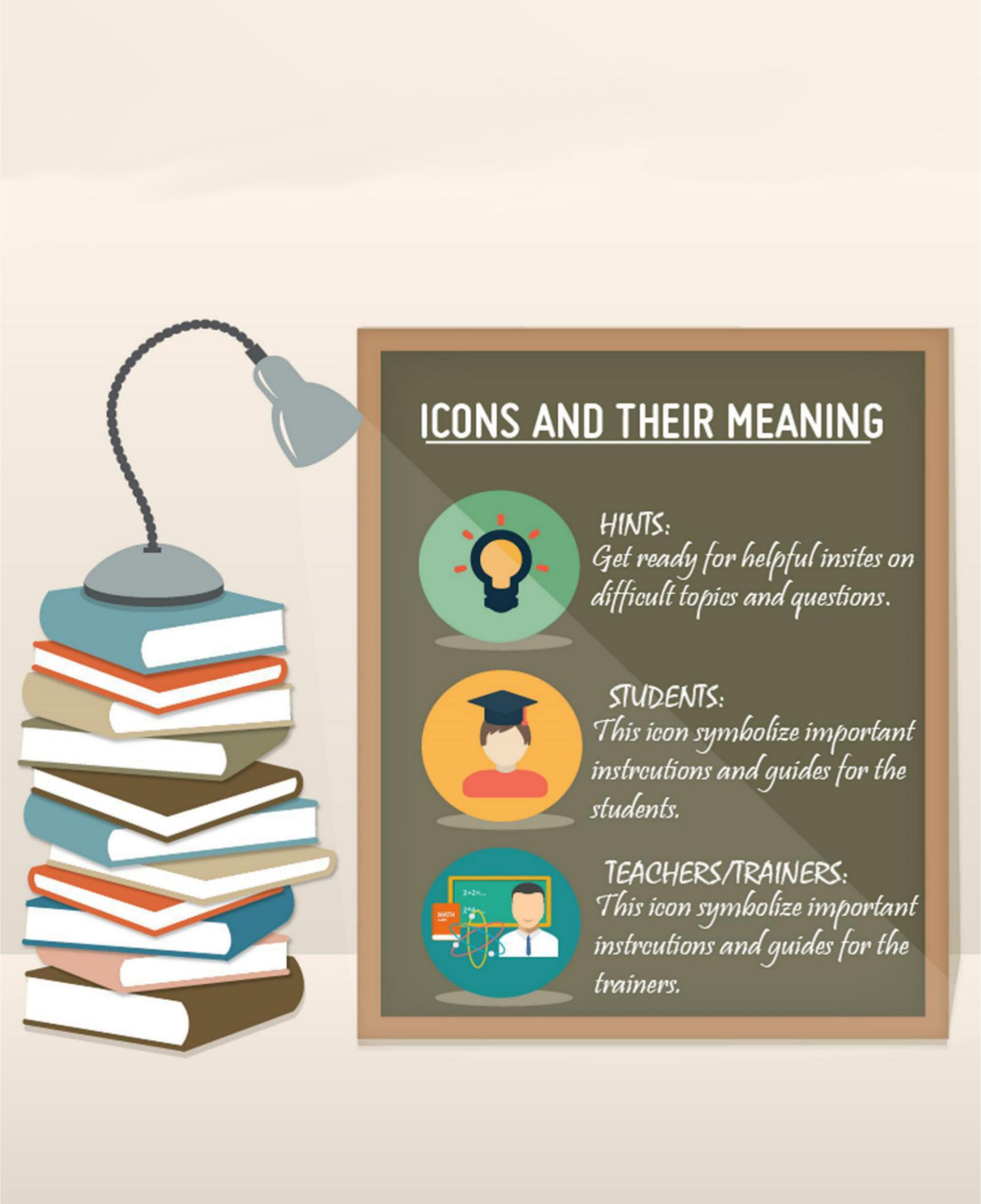
CORE JAVA

MANUAL V8.3

MODULE CODE:

ANUDIP FOUNDATION



ICONS AND THEIR MEANING

HINTS:
Get ready for helpful insites on difficult topics and questions.

STUDENTS:
This icon symbolize important instrcutions and guides for the students.

TEACHERS/TRAINERS:
This icon symbolize important instrcutions and guides for the trainers.

Module 6: Thread and Exception Handling

Chapter 2

Objective: After completing this lesson you will be able to : * Gain an understanding of thread life-cycle and its states * Learn about thread declarations and thread functions	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 120 minutes	**Practical Duration:** 0 minute
Total Duration: 120 minutes	

Chapter 2

2.1 Thread Life-cycle and states

i) Thread Life-cycle

A Java thread can be in five different states during its life-cycle. These are different stages when a thread is capable (or incapable) of performing certain functions. The five states of a thread life-cycle are -

1. New
2. Runnable
3. Running
4. Non-Runnable
5. Terminated

Let us take a look at each of the states of a thread' s life cycle -

1. New - A thread is in a new state if a programmer creates a thread class instance prior to the initiation of a start() method.
2. Runnable - A thread is in a runnable state if the start() method has been initiated, but the scheduler has not selected the thread as a running thread, yet.
3. Running - A thread is in a running state if it has been selected by the thread scheduler.
4. Non-Runnable - A thread is in a non-runnable or blocked state if it is alive, but not selected to run currently.
5. Terminated - A thread is a terminated state once it dies after the exiting of its run() method.

2.2 Thread Declaration- Extend Threads and Implements Runnable

A Java thread can be declared in two ways -

i) **by extending Java thread –** This way of declaring a thread involves creating a new class extending Thread. The steps that follow are overriding the run() method and creating a class instance. After calling start(), the run() method is executed.

Example code –

```
public class MyClass extends Thread {
          public void run(){
          System.out.println('Class is running');
        }
      }
```

ii) **by implementing runnable interface –** This method involves the creation of a class for implementing the Java Runnable interface. To achieve this, a class implements a single run() method. The new thread' s code is included within the run() method.

Example code –

```
public class MyClass implements Runnable {
public void run(){
System.out.println('Class is running');
  }
}
```

2.3 Thread Functions- run(), start(),

Java threads have many functions, among which these are the most fundamental ones –

public void run(): It executes actions for a thread

public void start(): It begins the execution of a thread

Implementing runnable interface is considered as more suitable for creating a thread, and is preferred over extend threads. The reasons behind this are –

* Extend thread class does not allow programmers to extend another class even if the need arises. Conversely, implement runnable lets programmers keep space for the possible extension of other classes.

* Using extend thread class entails every thread creating its unique objection and respective associations. Conversely, implement runnable enables programmers to share one object across multiple threads.

Instructions: The progress of students will be assessed with the exercises mentioned below.

MCQ (10 minutes)

1. Are thread states part of their life-cycles?

a) Yes

b) No

c) Life-cycles are parts of states

d) only sometimes

2. A Java thread can be in one of _______ states.

a) four

b) seven

c) five

d) None of the mentioned

3. Which of these is not a Java thread state?

a) runnable

b) running

c) runner

d) terminated

4. A thread is in a new state if a class instance is created ________ start() method initiation.

a) before

b) after

c) during

d) None of the mentioned

5. A thread in a runnable state has not been selected by a __________ yet.

a) compiler

b) scheduler

c) counter

d) None of the mentioned

6. A ____________ thread is one already selected by a thread scheduler.

a) runner

b) running

c) run

d) None of the mentioned

7. A non-runnable thread state is also known as a ________ state.

a) unblocked

b) blocked

c) open

d) None of the mentioned

8. What state is a thread in when a run() method exits?

a) running

b) terminated

c) initiated

d) None of the mentioned

9. A Java thread can be declared by ___________ a Thread class.

a) extending

b) encapsulating

c) enumerating

d) None of the mentioned

10. The ____________ interface can be implemented to declare a class.

a) Running

b) Runnable

c) Run()

d) None of the mentioned