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 1

Objective: After completing this lesson you will be able to : * Gain an understanding regarding the concepts of threading and multi-threading * Learn the differences between threads and processes in Java	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 80 minutes	**Practical Duration:** 40 minutes
Total Duration: 120 minutes	

Chapter 1

1.1 Concept of Threading and Multithreading

A thread in Java is a flow of control within a program. The java.lang.Thread class is used for generating and controlling threads in Java.

i) Threading

Threading deals with the management and execution of Java threads within programs. Programmers can perform threading to execute single or multi-threads, based on the requirements of a program. A single thread executes a single threading within a process. If many tasks need to be run at the same time, using multiple threads is ideal.

ii) Multiple threading

Multiple threads are used for multiple threading of processes. Multithreading is necessary if two or more tasks are to be run simultaneously within a Java process. Each thread in a multithreaded process might be assigned different tasks. They can perform their specified tasks at the same time alongside the other threads, or at different times, or at intervals. Multithreading can also be explained as multiple threads within a process executing byte-code instructions.

Basic multithreading code example

```
class Multi extends Thread{
        public void run(){
        System.out.println('multiple thread');
        }
        public static void main(String args[]){

        Multi t1=new Multi();
```

```
t1.start();
    }
   }
```

Output: multiple thread

1.2 Thread vs Process

The differences between thread and process in Java

* Process is a program while thread is a part of a program.
* Process cannot communicate effectively while thread provides efficient communication.
* Processes run in an isolated manner while threads share memory resources.
* Processes take more time to create and destroy, while threads take less time to both create and destroy.
* Processes are known as heavyweight tasks, while threads are known as lightweight tasks.

Practical (40 minutes)

See the example programme for Java multithreading below. Write the same programme to show multithreading for the string "multi threads". Show the resulting output. Repeat the same programme for the string "thread multiple"

```
class Multi extends Thread{
public void run(){
  System.out.println('multiple thread');
  }
  public static void main(String args[]){
  Multi t1=new Multi();
  t1.start();
```

```
}

}
```

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

MCQ (10 minutes)

1. A Java thread is a flow of _____ within a program.

a) control

b) commands

c) controllers

d) None of the mentioned

2. What class is used for generating a thread in Java?

a) Java.lang.Thread

b) Java.lang

c) Java.util

d) None of the mentioned

3. If a programmer must create a program where two tasks run simultaneously, he will choose _____.

a) multi threading

b) single threading

c) both a and b

d) None of the mentioned

4. Multiple threads handle _____ tasks.

a) different

b) only identical

c) rejected

d) None of the mentioned

5. Can two threads work at the same time?

a) No

b) Yes

c) Very rarely

d) They cannot co-exist

6. Multiple threads execute _____ instructions.

a) bit-code

b) byte-code

c) both a and b

d) None of the mentioned

7. Is a Java process a part of a thread?

a) No

b) sometimes

c) Thread is a process sub-part

d) None of the mentioned

8. Which communicates better?

a) process

b) thread

c) sub-thread

d) None of the mentioned

9. Can threads share memory resources?

a) yes

b) no

c) only sometimes

d) Memory shares thread resources

10. Threads are also known as _____ tasks.

a) base weight

b) heavyweight

c) lightweight

d) None of the mentioned