# JVM Performance Engineering & Troubleshooting Training

## Introduction

| | |
|---|---|
| 1 | Common Performance Problems |
| 2 | JVM Overview |
| 3 | Troubleshooting Principle: CAR |

## Garbage Collection

| | |
|---|---|
| 4 | What is Garbage Collection? |
| 5 | What are the Garbage Collection KPI? |
| 6 | How to enable GC Log? |
| 7 | Understanding GC Log |
| 8 | Garbage Collection log analysis tool |
| 9 | How to reduce GC pause time? |
| 10 | Rotating GC log files |
| 11 | Which GC Algorithm to use? |
| 12 | Lab: Enable GC logs |
| 13 | Lab: Change memory settings and study its impact |
| 14 | Lab: Troubleshooting Garbage Collection problem |
| 15 | Lab: Analyze Sample GC logs |
| 16 | Lab: Analyze your applications GC logs |

## Threads

| | |
|---|---|
| 17 | What are threads? What are different thread states? |
| 18 | 8 options to capture Thread Dumps |
| 19 | Understanding Thread Dumps |
| 20 | Thread dump analysis patterns |
| | a. Athlete Pattern<br>b. Deadlock Pattern<br>c. Thread Mill Pattern<br>d. Atherosclerosis pattern<br>e. Traffic Jam Pattern<br>f. RSI Pattern<br>g. Additives Pattern<br>h. Leprechaun Pattern |
| 21 | Thread dump analysis tool |
| 22 | Profiling & optimizing CPU |
| 23 | Lab: Capture thread dumps |
| 24 | Lab: Troubleshooting CPU spikes |
| 25 | Lab: Analyze sample Thread dumps |
| 26 | Lab: Analyze your applications thread dumps |

## Memory

| | |
|---|---|
| 27 | Why Memory Leaks? |
| | a. lang.OutOfMemoryError: Java heap space<br>b. lang.OutOfMemoryError: PermGen space<br>c. lang.OutOfMemoryError: Kill process or sacrifice child<br>d. lang.OutOfMemoryError: GC overhead limit exceeded<br>e. lang.OutOfMemoryError: Metaspace<br>f. lang.OutOfMemoryError: request bytes. Out of swap space?<br>g. lang.OutOfMemoryError: unable to create new native thread<br>h. lang.OutOfMemoryError: Requested array size exceeds VM limit |

| | |
|---|---|
| 28 | 7 options to capture Heap Dump |
| 29 | 3 options to capture Heap dump from Android |
| 30 | Understanding Heap dump |
| 31 | Profiling & optimizing memory usage |
| 32 | Memory wastage patterns<br>a. Duplicate Strings<br>b. Duplicate Objects<br>c. Duplicate Arrays<br>d. Inefficient Collections<br>e. Inefficient Finalization<br>f. Inefficient Arrays<br>g. Boxed Numbers<br>h. Wrong memory settings |
| 33 | Heap dump analysis tools |
| 34 | Lab: Capture heap dumps |
| 35 | Lab: Troubleshooting Memory leaks |
| 36 | Lab: Analyze sample Heap dumps |
| 37 | Lab: Analyze your applications heap dumps |

## Monitoring

| | |
|---|---|
| 38 | Primary JVM arguments & their impacts |
| 39 | Application Performance Monitoring |
| 40 | Micrometrics to forecast Performance Tsunamis |