Java Training2

Days	Topics to be covered
1	Java-Why, JDK- package and tools, Java - environments
2 , 3	Linux- basic commands, VI editor, Shell - basic programming
4	Java - basic programming
5 , 6	Java- OOPS , classes and class designs
7, 8 ,9	Exceptions, I/O Streams and Multithreading
9 - 12	Java - Data structures
13 ,14	Network communications
15 -17	Socket communications and programming
17- 20	Java - NIO

Java - why?

- Learn about basic characteristics of Java.
- What are its advantages comparable to other languages such as C, C++
- What are the components of JRE?

JDK - package and tools

- Explore the component of a JDK package.
- Try installing and uninstalling different JDK versions.
- Know the tools like javac, java, jar, jstack, jmap, jhat and keytool.

Java - environments

- What are environmental variables how it is useful? How can you use them in your application?
- What is the difference between PATH and CLASSPATH?
- What are system properties and how it is different from environmental variables?
- List out all of the system properties in java using a program.

Linux - basic commands

- Learn about basic linux/unix commands . Try exploring them in your editor

VI editor

- Learn about VIM editor and some basic commands.
- Know how to open a file, edit, save, code mode, cut , copy , paste , move horizontal, move vertical, select etc ..
- Below tutorial can be a help

https://www.cs.colostate.edu/helpdocs/vi.html

http://openvim.com/

Shell - Basic programming

- Learn some basics about shell **sh** (Bourne shells).
- Learn basic programming in shell.

    1. How to use linux commands within shell?
    2. Try getting output of any command inside a shell and print it.
    3. Try if , if - else, nested loops.
    4. Try for loop, while loop, do while loop
    5. Try sleep inside a shell program.

- Now that you are familiar with basics of shell programming, write a shell program to continuously monitor your local network to see if there are any new connections. Print the newly established connection count and also the IP. (You can have 30 sec interval between each monitoring).

Java - basic programming

- Write a simple "hello zoho" program.

    1. Why main class must be "public"?
    2. Why main method should be "public static"?

- What happens when a class is compiled ?

    1. What information does ".class" file contain?
    2. What happens when you run a class?
    3. What is the role of class loader? Different types of class loader ? Write a sample program.

- What is *packaging* in java means , why it is used?

    1. Add "*package "com.zoho.test";* " at the beginning of the java file.
    2. Compile and run it, if you are not able to run check why ?
    3. What can be done using javac "-d" and "-cp" options?
    4. Try "export CLASSPATH=/home" in your environment , compile and run again. See any problem ? Check why.

- Write a program to find the size of different data types (Integers,Long, char etc).
- Is it possible to set/get a system properties/ environmental variable from the application? Write a program to prove the same.
- Write a simple method and invoke that method using class an object.

    1. Can we invoke a method without using class objects ? how ? Prove with an example.
    2. If the above is achievable , can we invoke a method which is in another class ? (without creating an object). Prove with an example.

- Understand the difference between local variable, global variable and static variables.

Java - oops, classes and class designs

- Learn about basic oops concepts like abstraction, encapsulation, objects, classes, inheritance, polymorphism, association, composition and aggression. How are these concepts used in java? Explore all with an example of your own.
- What are access specifiers? Which of the above oops concepts does it illustrate? List out its different types and usage using a program.
- What is use of constructors in a class.
- Write any sample to explain the use of constructor in a class.
- Assume there is a base class B and another class A (derived from B) A has 12 arguments in its constructor, all these 12 arguments are in class B. How can I pass/initialize all these 12 arguments to base class in a single statement ?
- Design a class such that it can only be instantiated only once .
- Design a class such that it can only be instantiated only once, if attempted to instantiate more than once.. return the already created object or same instance.
- Can we have a class within a class. Prove with an example.
- Can we access inner class from outer class or classes from another package? Is no , why ?
- What is the difference between overloading and overriding? Can you override toString() function in a class? Write a sample program to show the difference.
- What is the use of abstract class and how different it is from a interface? Can a abstract class be instantiated? Write a sample program for all cases.
- What is enumeration? How different or similar is it to a class? Can you use abstract methods in enum? Show us with an example.

Java - Exceptions / IO Streams

- Exceptions

    1. What is the need for exceptions in java? Overview its subtypes.
    2. How can we handle exception in java? Try out an example.
    3. What is the need for finally block? Show us with an example
    4. How can we create and throw exception?
    5. When you mention a method throws Exception? What does it mean?

- I/O Streams

    1. How can we write to / read from disk files, devices, etc... in a java?
    2. Overview different types of I/O streams mentioned below
        1. Byte streams
        2. Characters streams
        3. File streams
        4. Data Streams
        5. Object streams
    3. How is one type of streams advantageous over other? Explain with examples.
    4. What happens when you use System.err/ System.out ?
    5. What happens when we use a buffered stream instead of a normal stream? Is there any advantages?
    6. How can we randomly access a file using java such that one can position the read index anywhere one chooses?
    7. Can we read/write a stream simultaneously? Try and find out.

Multithreading

- What is the need for threading concept?

1. What is serial processing and parallel processing?
2. What is the need for multi core CPU's?

- Threading in java.

    1. How to create a thread?
    2. How to start and run the thread?
    3. How to create more than 1 thread ? Can we identify each thread separately ? Can I name these threads? Show us with an example.
    4. Difference between Thread and Runnable?
    5. Create two threads T1 and T2, (you can have some looping functionality inside each thread run) . Start T1 and T2 immediatly. Hope both threads started parallelly . Now, is it possible to start T2 thread after T1 thread completes ?
    6. What is the use of interrupt, join, wait in Thread? Write an example to show their usage.
    7. Write a program such that 1 thread will continously produce messages in an interval , another thread should consume the same messages and print.

- Why we need multithreading ?
- Write a simple use case to show why we need multithreading and how it can be useful.
- Lets say, we have a huge directory with lot of files in each. How can we use multithreading here to achieve better performance ? Write a program for the same.
- What is immutability and why it is good ?
- How to make a class immutable ?
- String , StringBuilder , StringBuffer what is the difference between these ? How immutability is correlated with String,StringBuilder and StringBuffer. Try out with an example for each.

Java - data structures

- Java List API's.

    1. Overview ArrayList class in Java. Try its features with few sample programs.
    2. Overview LinkedList class in Java. Try its features with few sample programs.
    3. When you will use LinkedList over ArrayList? Show us with an example.
    4. What happens if multiple threads access LinkedList/ArrayList? Try it with an example.
    5. Overview Vector, Stack.
    6. When you will use Vector over ArrayList? Show us an example.

- Java Map API's.

    1. Overview about Hashtable and HashMap in Java. What is difference between these two.
    2. Following examples can be tried
    3. Iterating by keys.
    4. Iterating by values.
    5. Can multiple threads access Hashtable ? Try an example and find out the solution.
    6. Overview ConcurrentHashMap.
    7. Why ConcurrentHashMap should be used over HashMap? Show us with an example
    8. Difference between Properties and Hashtable ? When to use which? Show us with an example.

- Java Queue.

    1. Overview about LinkedBlockingQueue, ConcurrentLinkedQueue
    2. Try to understand when to use these queue's with an example.

- What is the difference between pass by value and pass by reference? How does it apply in java? Write a program testing the same for basic data types, objects, arrays, strings and custom objects.

Network communications

- How do computers communicate?
- What happens when you enter a website in your browser and fetch it?
- Overview the basic difference between transmission modes and communication modes.
- Learn about basic network concepts such as IP address, MAC address, Subnetting, Routing, etc..
- Overview OSI and TCP/IP reference models.
- Understand the diffference between TCP and UDP.
- What is the need for a secure connection?
- Overview SSL handshake. Understand the different steps involved in it.

Java - Socket programming.

- What is a Socket? Overview Closeable.
- What is a Datagram? How is it differs from Socket?
- What is the difference between keystore and truststore? How can we set them in a socket program?
- How can we create a self-signed certificate using java?
- Try to write a simple socket program that accepts a client and transmits data between them?
- How can we write a server that accept that accepts multiple clients? Show us it in a chat scenario.

Java - NIO

- What is a Buffer? Overview its state variable, member functions and understand their usage?
- Overview Channel. How different is it from a Stream?
- What is the need for Selector and SelectionKey?
- Overview FileChannel.
- Overview SocketChannel.
- How is SSL communication established in a SocketChannel? Write a client/server program and try out.
- Recreate the chat scenario in the socket programming section using NIO Socket channel and prove how one is advantageous over other.