**Course Outline**

**Expectations:**

* **All the required softwares are installed**
* **Basic Java, SQL, HTML, CSS and Java scripts are done**
* **Maven project is set and all are aware of the same**
* **Git is also covered**
* **Junit test cases**

1. **Introduction**

* Introduction
* SDLC
* Explain the ecosystem and people surrounding us, complete chart
* Job responsibility study
* What is Java?
* Features of Java
* JDK, JRE and JVM
* Java Variables

1. **Code formation and Encapsulation**

**Please create one Employee class example below is the link with following concepts must be covered in that example**

* Object
* Class
* Reference
* Constructor
  + Default and parameterized
  + Constructor overloading
  + And some of the constructor rules
* Encapsulation
  + Setter and getter methods
  + Reusability
  + Testing

[**https://www.geeksforgeeks.org/encapsulation-in-java/**](https://www.geeksforgeeks.org/encapsulation-in-java/)

* Method vs constructor
* this keyword
* Local, instance and static variable in the same example
* Static method, instance method
* Package
* Access modifiers
* Naming convention
* Proper package creation and java version set up all the time
* Eclipse shortcuts
* Try to install sonar lint in the eclipse
  + <https://www.youtube.com/watch?v=RytiRATMyNo&ab_channel=LoveToLearn>
* Control Statements
* Flow chart diagrams
* IF- Elseif - Else Nested with example
* Switch with example
* Loops:
  + For Loop - Traditional and Enhanced
  + Nested Loops
  + Infinite loop: compare that with Calculator example
  + While and do While loops
  + Explain array here with Enhanced for loop
  + Break
  + Continue
  + Java comments

**Assignment:**

* Java Operators with same Calculators example Again make sure that all the test cases are written properly
* Create example of Inheritance using extends and implements

1. **OOPS concepts in details**

* Oops concepts
  + Create one example in front of the student where following concepts are incorporated
  + Inheritance IS-A relationship -Vertical Relationship
  + Polymorphism
    - Compile - Method overloading
    - Runtime - Method overriding
  + Abstraction
    - Abstract class
    - Interface
* Upcasting
* Down casting
* Super Keyword
* Final keyword

Create your immutable class

* Abstraction vs Inheritance

**Assignments:**

* + Read about new features in java 1.8 interface
  + What is diamond problem creating example of the same

1. **OOPs- continue**

* Diamond Problem
  + <https://www.javainterviewpoint.com/multiple-inheritance-java-8/>
* Java 1.8 Interface
* Default Method and Static Method
* Package with respect to Inheritance
* Instance of Operator and Upcasting down casting
* Covariant return type with Animal and Trainer example
* Instance initializer block Static and Instance
* Explain Final keyword and immutable- refer assignments below
* Access Modifiers with respect to Inheritance, Visibility table
* Has - A relationship with example
  + Associations
    - One to one
    - One to many
    - Many to one
    - Many to many
  + Aggregation
  + Composition

**Assignment:**

1. Bank application use case (Use Has-A and Is-A relationship)

Main functionality: -

1) Open a Saving account

2) Open Current account

3) Register PIN

4) Deposit money

5) Withdraw money

6) View balance

Account number required to register pin/deposit/withdraw/view balance

Account number required and pin required to deposit/withdraw/view balance

1. Create your own Immutable class with example
   1. <https://www.javatpoint.com/how-to-create-immutable-class>
   2. <https://dzone.com/articles/how-to-create-an-immutable-class-in-java>
2. **Core concepts continue**

* Arrays
  + Iteration of the array
  + Characteristics of the array
  + Memory storage
  + Performance of the array
* Object Class
  + All the 11 methods 🡺 <https://www.javatpoint.com/object-class>
    - toString
    - Equality of an object
      * Equals and ==
      * <https://www.javatpoint.com/java-string-equals>
* Object Cloning
  + Deep and Shallow cloning with example
    - <https://www.javatpoint.com/shallow-copy-vs-deep-copy-in-java>
  + Marker interface
* Pass by value and Pass by reference
  + - <https://www.geeksforgeeks.org/g-fact-31-java-is-strictly-pass-by-value/>
* Pass by value and pass by reference with example
  + <https://www.geeksforgeeks.org/g-fact-31-java-is-strictly-pass-by-value/>

1. **Core concepts continue**

* Math class with simple example 🡺 <https://www.javatpoint.com/java-math>
* Wrapper Class – Autoboxing and unboxing Java 1.5 new features
* Recursion
  + Create a Fibonacci series program using java recursion
    - <https://www.javatpoint.com/fibonacci-series-in-java>
  + Sum of 1 to 100 numbers without loop
    - <https://www.geeksforgeeks.org/how-will-you-print-numbers-from-1-to-200-without-using-loop/>
* Complete all the String related methods and functionalities
* Command Line and System Argument with example
* Java String
  + String Object creations (How many objects are creating)
  + String constant Pool
  + Interns and other functions
  + String buffer and builder
  + String Tokenizer

**Assignments:**

**Data Structure Practice questions**

[**https://hackernoon.com/50-data-structure-and-algorithms-interview-questions-for-programmers-b4b1ac61f5b0**](https://hackernoon.com/50-data-structure-and-algorithms-interview-questions-for-programmers-b4b1ac61f5b0)

1. **Exception Handling**

* Start with JavaT Point Exception Video
  + <https://www.youtube.com/watch?v=ohpCMpderow&t=242s&ab_channel=javatpoint>
* Explain the end-to-end flow and importance of the catch block with respect to Log monitoring when we have the production issue occurs
* Again, start with Inheritance
* Types of Exception
* Difference between Checked and Unchecked Exception (With Example)
* Try, Catch and Multiple Try Catch
* Nested try
* Finally block
  + Final, Finally and Finalize
* Java throw keyword
* Java Exception Propagation

**Assignments:**

Create File reading and writing example with following topics from JavaTpoint

* File Inputstream
* File Outputstream
* BufferedInputStream
* BufferedOutputStream
* FileWriter
* FileReader
* BufferedWriter
* BufferedReader

1. **Exception handling cont.**

* Throws Keyword
  + Here need to start the File reading and writing example – Assignment
  + Explain the throws keyword and checked exception
* Throw vs throws (Deep explanation with Example)
* Custom exception
* Explain Autocloseable interface with example
  + <https://www.javatpoint.com/java-try-with-resources>
  + <https://www.geeksforgeeks.org/try-with-resources-feature-in-java/>
* Java 1.7 new features of Exception handling
  + Try with resource
  + Try without catch and finally
  + Catch block with more than one exception handler pipe separated
* Inner Class with example
  + [Member Inner Class](https://www.javatpoint.com/member-inner-class)
  + [Anonymous Inner Class](https://www.javatpoint.com/anonymous-inner-class) – Explain this with Lambda expression
  + [Local Inner Class](https://www.javatpoint.com/local-inner-class)
  + [Static Nested Class](https://www.javatpoint.com/static-nested-class)
  + [Nested Interface](https://www.javatpoint.com/nested-interface)
  + Lambda expression
  + Functional interface

**Assignments:**

* Example of Inner class with use case – need to update one assignment
* Create simple thread example

1. **Multithreading**  
   Explain Multi threading with real world examples where it needs to be used, theory of the same

* Thread creation with example
  + Extending the Thread class
  + Implementing the Runnable interface
* Thread life cycle
* Thread scheduler
* Thread.sleep()
* Thread naming, Priority, join, yield (create example if possible)
* Daemon thread – Create example
  + [Example1](https://dzone.com/articles/creating-daemon-thread-in-java" \l ":~:text=Creating a thread as a,an initial value of false.)
  + [Example2](https://www.javatpoint.com/daemon-thread)

Important Interview question:

* Can we start the thread twice.? Why?
* Which is a better way to create a Thread in java

Assignment:

* Create thread pool example, in next lecture we will continue with the same

<https://java2blog.com/java-executor-framework-tutorial-example/>

1. **Thread pool**

Thread pooling example 🡺 <https://drive.google.com/file/d/1BvCEIvWfPk3gL0pPc5fT89n18-K7KgS5/view?usp=sharing>

* ThreadPoolExecutor
  + ExecutorService using Callable and Future
  + NewFixedThreadPool
  + NewCachedThreadPool
  + NewSingleThreadExecutor
  + Scheduledthreadpoolexecutor
  + FutureTask
  + ExecutorCompletionService

Assignment:

* Read about design patterns
* What is Object pooling and create and example of the same
* Create example of Synchronization

1. **Thread synchronization and remaining topics**

* Synchronization
  + Synchronized Method
  + Synchronized block
  + Static synchronization
* Deadlock
  + Create example
  + Ways to solve the Deadlock
    - Change the sequence
    - Interthread process communication
      * Wait
        + Difference between wait and notify
      * Noitify
      * notifyall
* **Quiz-1** [**🡺 https://www.javatpoint.com/result.jsp**](about:blank)
* **Quiz-2 🡺** [**https://www.javatpoint.com/directload.jsp?val=91**](https://www.javatpoint.com/directload.jsp?val=91)

Important Interview question:

* Garbage collectors follow which algorithm? → **Mark-Sweep Algorithm**
* **Java concurrency interview questions** 🡺 <https://www.digitalocean.com/community/tutorials/java-multithreading-concurrency-interview-questions-answers>

**Links :**

* [**https://java2blog.com/java-threadpoolexecutor-example/**](https://java2blog.com/java-threadpoolexecutor-example/)
* [**https://java2blog.com/java-executor-framework-tutorial-example/**](https://java2blog.com/java-executor-framework-tutorial-example/)

**Assignments:**

**1) Multithreading concept –**

* **Read transaction file which contain debit & credit records**
* **Tx file – csv format**
* **Fields: id, category, description, amount, type**
  + **Mandatory: id, category, amount, type**
* **Read file & segregates the file to debit.csv, creadit.csv & error.csv**
* **Validation:**
  + **Check mandatory fields**
  + **Check amount must be double**
  + **Check id must be long**

1. **Serialization and Externalization with remaining IO topics**

* Create an example of Serialization
  + ObjectOutputStream
  + ObjectInputstream
* SerialversionUID
* Transient Keyword
* Externalization with example
* Java date time
* Java Reflection API
  + Call private method from another class

Overview of Collection framework:

* + Collection Framework
  + [Collection Diagram](https://drive.google.com/file/d/1fBYq4UhBnM9VVQwbO_zvM9q1Le_7TBaQ/view?usp=sharing)
  + List interface
  + Array List
  + Linked List
  + Iterators 🡺 <https://www.geeksforgeeks.org/iterators-in-java/?ref=lbp>

**Assignment -1**

* Create a programme of List interface iterate five different ways.

<https://www.geeksforgeeks.org/iterate-through-list-in-java/>

1. **Collection**

* What is Framework?
* What is API?
* List interface
  + ArrayList (internal working)
    - <https://javagoal.com/internal-working-of-arraylist-in-java/>
  + LinkedList (internal working)
    - <https://www.netjstech.com/2015/08/how-linked-list-class-works-internally-java.html>
  + Stack
  + Vector
* Set Interface
  + Hash Set
  + Linked HashSet
  + SortedSet Interface
  + TreeSet
* Queue Interface,
  + Deque
  + ArrayDeque
* Complexity 🡺<https://www.youtube.com/watch?v=FhNJ6aikTVI&ab_channel=BrandanJones>

**Assignments:**

1. **Collection continues:**

* Map interface 🡺 <https://drive.google.com/file/d/1jk7S7wNfMR2vCgV7UGmGgN0octCKv6Jk/view?usp=sharing>
  + HashMap (internal working)
  + Linked HashMap,
  + TreeMap
  + HashMap
  + Hashtable
* Collections class
* Sorting In collection
* Comparable interface and Comparator Interface
* Concurrency in collection
  + ConcurrentHashMap
  + Create an example fail safe and fail fast Concurrent modification example
* Take Quiz 🡪 <https://www.javatpoint.com/directload.jsp?val=92>

**Assignments:**

1. Bank application use case (Use Has-A and Is-A relationship)

Main functionality: -

1) Open a Saving account

2) Open Current account

3) Register PIN

4) Deposit money

5) Withdraw money

6) View balance

Account number required to register pin/deposit/withdraw/view balance

Account number required and pin required to deposit/withdraw/view balance

hint: to maintain an account you can use one of the collection classes here, apply the oops concept, custom exception, etc

2) Find pizza cost.

Three type of pizza margreta, peppe paneer n farmhouse each pizza has diff price. One can add toppings and each topping have diff price. One can add many topping like corn, cheese, paneer, pinaple etc for specific pizza.

Customer should get what should be the price after order pizza with toppings.

Use Decorator pattern, Singleton , factory design pattern .

1. **JDBC**

* Database queries, sql workbench
* JDBC introduction
* Type of driver
* Connectivity with MYSQL
* DriverManager
* Connection
* Statement
* ResultSet
* PreparedStatement – Hacking, sql Injections

1. **JDBC Continue**

* How to store the image or file to database
* Callable Statement
* Store procedure
* Transaction Management

**Assignments:**

**Video:**

<https://drive.google.com/file/d/1Pgr-w3GEQg3GuGfdjjekCIwuJS7ulb1o/view?usp=sharing>

* Download tomcat
* Create simple webapplication
* Watch webapplication video and create simple servlet hello world
* Make sure we create maven web project at this stage

**--------------------------------------------Revision Break --------------------------------------------------------------**

1. **Servlet and web terminology**

Webproject to import with all different possible way to create servlet

* Java Servlet Overview
* Servlet Life Cycle
* Web Terminology
* Client server architecture
* WWW
* URL
* URI
* DNS
* IP Address/localhost
* Hosting
* Basic of HTTP - GET, POST, PUT, Delete etc
* https://www.youtube.com/watch?v=iYM2zFP3Zn0&ab\_channel=TraversyMedia
* JSON and XML
* HTML - basics and forms quick recap - its expected that has done
* Get vs Post
* Servlet API
* Servlet Interface
* Generic Servlet
* HTTP Servlet
* Life Cycle of a Servlet
* War file
* Load on Start up
* Deploy the webapplication on external tomcat without Eclipse

**Assignments:**

* Start creating a webapplication with login and registration page

1. **Servlet Continue**

* Servlet Request
* Methods of Servlet request
* Request dispatcher - example
  + Include
  + forward
* send redirect
* Difference between forward and send redirect methods
* ServletConfig
* ServletContext
* Difference of ServletConfig and ServletContext interface
* Attributes in servlet
  + Context
  + Request
* ServletFilter
  + Create example of ServletFilter
  + Create login and registration page and create connection, Store data in database with the help of connection object.

**Assignments:**

Needs to be updated

1. **Servlet continues**

Explain session tracking with real-world example

**Session Tracking**

* Cookie – Explain with example
* Four techniques in Session Technique.
  + Cookies
  + Hidden from field
  + URL Rewriting
  + HTTP Session
* Session login and logout
* Event and Listener (only two listener with example Context and Session)

Assignments:

* Convert your webapplication using the JDBC using listener and complete end to end flow

1. **JSP - Continue**

Link of JSP Project example:

* Life cycle of JSP
* JSP Scripting elements
  + Scriptlet
  + declaration
  + expression
* 9 implicit objects
  + Request
  + Response
  + Config
  + Application
  + Session
  + PageContext
  + Page
  + Exception
* JSP directive elements:
  + Page directive
  + Include directive
  + Taglib directive
* JSP Action elements
  + Jsp forward
  + Jsp include
  + Java bean class
  + Jsp: useBean
  + Set & get property
  + Displaying Applet in jsp
* JSP expression Language (EL)
* JSTL

Assignment

* Convert your Servlet applications using JSP now with respect to given project
* create example with Apache DBCP pool
* hibernate project link and give some overview <https://drive.google.com/file/d/1gO5xA_LbdVpLXFP6xLYW8JZ23vzYZNT/view?usp=share_link>

1. **Hibernate**

* What is JPA and specifications
* Difference between Hibernate and JDBC
  + Let us use breakout rooms and divide and give some group activities
* Explain flow by drawing some diagrams and do some comparison with jdbc
  + Sessionfactory
  + Session
  + Transaction
* Hibernate simple registration standalone application
  + Insertion session.save
  + Retrieval session.get and session.load
  + HQL query
  + SQL query

**Assignments:**

1. **Hibernate continue**

* Explain the relationships again
  + One to one
  + One to many
  + Many to one
  + Many to many

* Showcase the example of each of above with example
  + Cascading
  + Lazy and eagar
  + Object Life cycle
  + <http://learningviacode.blogspot.com/2012/02/hibernate-object-life-cycle.html>
* Explain about IS-A relationships how hibernate does the same
  + https://www.viralpatel.net/category/j2ee/hibernate/

**Assignments:**

* What is design patterns?
* What is singleton design patterns?

<https://www.miit.tech/post/java-singleton-design-pattern-best-practices-with-examples>

* Create simple helloworld spring example – import the projects

https://drive.google.com/file/d/1A85Zvw0zVRvc0IXAuzb9hNl5wymEbrVY/view?usp=drive\_web&authuser=1

1. **Spring Framework**

**PPT Link: need to put it**

1. What is Spring?
2. IOC
3. Dependency Injection
4. EJB
5. Core Container
6. Spring modules
7. Spring application
8. **Spring Framework**
9. Bean Life Cycle
10. Creating beans
    1. Simple bean
11. Injecting inner beans
12. Auto wiring
    1. Types of wiring
    2. byname
    3. bytype
    4. constructor
    5. autodetect
13. Bean Scoping
    1. Singleton
    2. Prototype
    3. Request
    4. Session
    5. Global-session
14. Spring web Model view controller (Spring MVC)
15. Understanding flow of spring web MVC
16. Advantage of spring MVC Framework
17. Spring Web MVC Framework Example
    1. Provide project information and configuration in the pom.xml file.
    2. Create the controller class
    3. Provide the entry of controller in the web.xml file
    4. Define the bean in the xml file
    5. Display the message in the JSP page

**Links :**

**Spring and SpringwithHibernate Example:**

<https://drive.google.com/file/d/1HYLo9GpQxaMcsI4QTHNLcTQ69dMExPzX/view?usp=share_link>

<https://drive.google.com/file/d/1EtV5ltlIuxv6qfoXedrAomTuhc5Y3jPP/view?usp=share_link>

<https://drive.google.com/file/d/1Y0uH29I9Hu2XtRrHMm3K5_PQgQ-cYUI4/view?usp=share_link>

<https://drive.google.com/file/d/1Hv-6JRxoNtHOqQ92EZbDOqaY3l-IJGyr/view?usp=share_link>

1. **Spring Boot and Microservice**
2. What is Spring Boot
3. Advantage Of Spring Boot
4. Spring boot crud example
5. What IS RAD (Rapid Application Development)
6. Difference Between Spring and Spring Boot
7. Rest API
8. Understanding of different Web API ( SOAP ,  REST )
9. Core Concept of micro services
10. Exposing Microservice
11. Consuming Microservice.
12. Simple Hello World Program in Spring Boot (Also Using spring initializer)
13. Angular application
14. Rest template and webclient

<https://www.tutorialspoint.com/spring_boot/spring_boot_rest_template.htm>

[**https://spring.io/guides/gs/consuming-rest/**](https://spring.io/guides/gs/consuming-rest/)

[**https://www.section.io/engineering-education/spring-boot-rest-template/**](https://www.section.io/engineering-education/spring-boot-rest-template/)

Movie catalog from git https://github.com/lekhana3003/Movie-Rating-Application

**Links :**

* [What is REST API? | Web Service - YouTube](https://www.youtube.com/watch?v=qVTAB8Z2VmA)
* <https://zetcode.com/springboot/datajpaquery/>
* <https://examples.javacodegeeks.com/enterprise-java/jms/jms-messagelistener-example/>
* Movie catalog from git: <https://github.com/lekhana3003/Movie-Rating-Application>
* <https://drive.google.com/file/d/11OLcD9dne8cfKFc9srurr1s5giT3YNWR/view?usp=share_link>