* Java Dev – Important interview questions
* Set-1 - Mix
* Choose a design pattern and speak about it
* How do you create an immutable class within Java
* How serialization / deserialization works in Java
* Permanent generation space
* How does an object become eligible for garbage collection? Can this be manually done?
* What is Finalize() method?
* Why isn’t add using generic types and remove using object types
  + - add(E e)
    - remove(Object o)
  + and why not remove(E e)?
* What happens at compile time with generic types?
* Difference between unchecked exception and errors?
* Polymorphism: Overriding vs overloading; which one is being done at compile time and which one at run time
* Spring: 1. Define all the bean scopes; 2. which is the default one; 3. where can you use the prototype
* Dependency injection
* Lazy initialization of beans
* Autowiring of constructor rather than directly the field
* Spring AOP: 1. Pointcut vs advice; 2. where have you used AOP before?
* SpringSecurity: Used before? What types have you worked with till now?
* Authorization vs Authentication; how does Spring Security handle these?
* When will you use flatmap()?
* Why were default methods added in Java interfaces?
* Multithreading: Runnable vs Callable
* What is futures in Java?
* How do you handle deadlock situations?
* Will the below code compile? If not, what changes do you suggest?
  + int counter = 0;
  + Stream.of(1,2,3,4).forEach(e -> c += e);
* In a lambda why do you use final variable?
* What are microservices? Compare vs monolithic?
* Strategies that a load balancer can implement
* Lot of data and latency is not an issue – how would you implement a load balancing strategy?
* Different isolation levels in a database
* Have you worked with Transactions in databases?
* Do you use Maven or Gradle?
* Worked with JMS before? Topic vs Queue?
* Favorite sorting algorithm?
* Set-6 Spring
* Current project and architecture
* What you like, what you don’t like about Java - Where Java suits
* Where Java is not suitable
* Static binding and dynamic biding while calling the methods - Examples
* Inheritance
  + Change of method modifier – from public to protected in the child class? Is that possible?
  + If not, why that is the case
* Why Spring / Spring Boot?
* Spring Boot / Spring - AOP have you worked on AOP – Aspect oriented programming
* Dependency Injection - Why and how
* Fibonacci series - what is it and code
* Microservices - Securing microservices - Improve your code
* Favorite design pattern - factory design pattern
* Java Bytecode - generic types
* What is commit? Data-memory-heap
* What is Phantom reference?
* Set-3 Java Basics
* What aspect of your existing/past project are you proud of - technical not functional?
* Can you please describe a design pattern of your choosing? Any non-structural design pattern?
* When would you choose to use Builder Design Pattern?
* What is an immutable class?
* How can we make your own class immutable?
* What is the difference between final class and final field?
* Difference between unchecked exceptions and errors in Java? Examples of these?
* What happens with generic types after you compile them?
* Have you encountered concept of type Erasure?
* Why/where are generics useful? Can you write a simple example of where generics can be useful?
* Esp. example of generics where you do not need abstraction.
* With reference to generics, are you aware of super keyword?
* Difference between a linked list and array list?
* How do you remove an element in an array list?
* What is the contract between hash code and equals?
* Can two identical hash codes result in different objects?
* Why map keys have to mutable in a hash-map?
* Question on collection with failsafe iterator
* How can you make a thread safe list?
* Difference between set and list
* Question on sorting algorithms
* Functional side of Java:
* What is a functional interface?
* What about default methods? If it has default – can you call it functional interface?
* Example of a default method on a functional interface of a Java API? Are you aware of Predicate interface?
* Set-4 – SQL/PLQSL
* Concepts in chat window:
* What is self-join, what is use of self-join?
* Types of joins.
* What are indexes use of indexes?
* What is having cause used for. Can we use this with where clause.
* Difference between Union and union all.
* Difference and uses of function and procedure.
* Programming Question:
* Find third max salary of an employee
* How to do one to one mapping.
* How to do many to many mapping.
* How to join tables in many to many mapping.
* Set-5 – Microservices.
* What is fall back method?
* Set-6 – Architectural questions
* How to configure two databases in your java application or spring boot application.
* What is two phase commit? How commit and back handled in distributed systems or micro services.
* What was your last project? What have you worked on?
* What micro-services have you worked on?
* Why memory-foot print and start time was a consideration for your project?
* Why did you create your own workflow micro-services? Why not airflow or some other BPM system?
* Are you developing core system or defining the workflows? If latter, how do you exactly define the workflows?
* Given the list of integers, one element is missing. Write a function to identify which one is a missing one?
* Lot of subsequent questions on improving the code, optimizing and general efficiency
* Can you write test cases for this code?
* Can you code-review this code? Critically examine it?
* How would Java know a specific undefined variable is integer?
* What is a wrapper class?
* What does super stand for?
* Difference between super and extends?
* What is your favorite design pattern?
* Example of visitor design pattern?
* Difference between Sybase and Oracle?
* Is transaction a java-development object or database-development object?
* Any Java code that you have written that you can commit in the Oracle GUI itself? Can we do this?
* Other important areas:
* Space and time complexity of algorithms
* Data Structures in Java
* Memory management in Java
* Simple algorithmic coding – Fibonacci series, finding smallest difference from a list etc.
* Constructors
* How are immutable classes different and how can you create them
* How does serialization and deserialization work in Java
* What is an issue with the below code:
  + add(E e)
  + remove(Object o)
* Polymorphism
* Exceptions - which ones have you encountered most
* How do you catch exceptions in your code?
* Multi-threading, thread locker
* If you have a GitHUb link how would you identify if this is a SpringBoot application
* Have you configured Spring Security from scratch? Any Security providers you've used?
* What is a functional interface? Supplier functional interface?
* Have you used StreamAPI? How would you handle exceptions?
* Can you convert an Object to Int?
* Set-7 – Aptitude questions.
* A train is 1 mile long. It travels at the rate of 1 mile per minute through a tunnel which is also 1 mile long. How long will it take for the train to pass completely through the tunnel?
* You have 2 bottles with capacity 3 liters and 5 liters respectively. You are at the bank of a pond full of water. You need to draw exactly 4 liters of water from the pond. You don’t have any other container or calibration available. How will you do it?
* Set-8 – Difficult programs.
* Looking for an API which would process the list containing words and return the group of anagrams only.ELBOW , LISTEN , SMILES, SILENT, ARISE, MILES, MONEY, OPTION, BELOWSMILES, MILESLISTEN, SILENTELBOW, BELOW