- Wrapper class, autoboxing, unboxing: https://www.javatpoint.com/wrapper-class-in-java
- Constructor & its rules, Types of Constructor, Constructor vs Method: https://www.javatpoint.com/java-constructor
- String: https://www.javatpoint.com/java-string
- Immutable string: https://www.javatpoint.com/immutable-string
- toString(): https://www.javatpoint.com/understanding-toString()-method
- String FAQS: https://www.javatpoint.com/java-string-faqs
- This vs super keyword && this() vs super() method: https://www.techiedelight.com/difference-between-this-super-keyword-java/
- Referencing subclass objects with subclass vs superclass reference: https://www.geeksforgeeks.org/referencing-subclass-objects-subclass-vs-superclass-reference/
- Upcasting vs Downcasting: https://www.javatpoint.com/upcasting-and-downcasting-in-java
- Static vs Dynamic binding: https://www.javatpoint.com/static-binding-and-dynamic-binding https://www.tutorialspoint.com/static-vs-dynamic-binding-in-java
- Difference Between Cohesion and Coupling: https://www.baeldung.com/cs/cohesion-vs-coupling
- Class relationship in java(uses a, is a, has a): https://www.scientecheasy.com/2021/02/class-relationships-in-java.html/
- Packages: https://www.scientecheasy.com/2020/06/packages-in-java.html/
- Singleton class: https://www.geeksforgeeks.org/singleton-class-java/
- UML diagram: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/ https://java-programming.mooc.fi/part-11/1-class-diagrams
- Reasons to serialize an object: https://stackoverflow.com/questions/2232759/what-is-the-purpose-of-serialization-in-java
- Why java doesn't allow pass-by reference?: Some people will say incorrectly that objects are passed "by reference." In programming language design, the term pass by reference properly means that when an argument is passed to a function, the invoked function gets a reference to the original value, not a copy of its value. If the function modifies its parameter, the value in the calling code will be changed because the argument and parameter use the same slot in memory.... The Java programming language does not pass objects by reference; it passes object references by value. Because two copies of the same reference refer to the same actual object, changes made through one reference variable are visible through the other. There is exactly one parameter passing mode -- pass by value -- and that helps keep things simple. -- James Gosling, et al., The Java Programming Language, 4th Edition
- Java short questions & interview questions (must read!): https://www.edureka.co/blog/interview-questions/java-interview-questions/#basic