**Core Java Fundamentals**

* **Basic Syntax:**
  + What are the different data types in Java?
* There are two types of data types in java
* Primitive
  + - byte : 1 byte : -2^7 to 2^7-1
    - short : 2 bytes : -2^15 to 2^15-1
    - int : 4 bytes
    - long : 8 bytes
    - float : 4 bytes
    - double : 8 bytes
    - char : 2 bytes
* Non-Primitives
  + - String
    - array
    - classes
    - interfaces
  + Explain the difference between == and .equals() methods.
* If we have 2 strings assuming str1 = “ABC” and str2 = “ABC” so, the .equals() method compares the data of the two strings and the == operator compares the references of the variable.
  + What is the difference between int and Integer?
* int is a primitive datatype and has a size of 4 bytes, but Integer is a Wrapper class. It is an immutable class.
  + How do you convert a string to an integer in Java?
* Integer.parseInt(<StringVariable>) method is used to convert String into Integer. And if the string do contain alphanumeric or special character it gives “NumberFormatException”
* **Object-Oriented Programming (OOP):**
  + What are the four pillars of OOP? Explain each with examples.
* Java follows OOPS paradigm. There are 4 pillers of oops
* Encapsulation : binding the data into single unit.
  + - Example: In a class, there are data members and member functions.
* Abstraction : hiding the inner-implementation from the end user is called Abstraction.

Abstraction is achieved by using Abstract classes.

abstract keyword is used.

This class contains both abstract methods as well as concrete methods

Abstract class cannot be instantiated. (Object of abstract class cannot be created.)

Subclass must extend abstract class to give proper implementation

Interfaces are used to provide 100% abstraction.

All methods in Interfaces are public and abstract.

A class can implement multiple interfaces.

* + - Example :
* Polymorphism : doing a same task in different ways.

There are two types of poly.

* + Run Time :
    - also known as overriding.
    - it is done in between two classes.
    - Using Inheritance is important.
    - Same method with same argument is used but it different classes.
    - Example :
  + Compile Time :
    - also known as overloading.
    - It is done in within a single class.
    - Using Inheritance is not important.
    - Same method signature but with different argument is used.
    - Example :
* Inheritance : It is a “Is a relationship” between two or more classes. Where Parent class share it properties to the child class. And Parent not allowed to see what child class have implemented in its class.
  + There are different types of inheritance.
    - Single :
      * Child class extends Parent class
      * Single level of inheritance only
    - Multiple :
      * implemented using interfaces only due to diamond problem issue
    - Multi-level :
      * Grand parent to parent to child reletionship
    - Hierarchical :
      * One parent with 2 child.
    - Hybrid :
      * Combination of different inheritance.
  + What is the difference between inheritance and polymorphism?
* Runtime poly. Requires inheritance in it’s implantation.
  + Explain the concept of method overriding and overloading.
  + What is the purpose of the final keyword?
  + 🡺 Final is a keyword used in java. It is used with class, methods and variables.
    - Final class : cannot be extended.
    - Final method : cannot be overridden
    - Final variable : cannot be re-initialized.
* **Exception Handling:**
  + What is the difference between checked and unchecked exceptions?
* Exceptions are of two types Checked exception and Unchecked exception.
  + Checked exception :
    - checked at compile time.
    - In IDE if we get red dot on left side of the screen, it denotes compile time exception.
    - It needs to be handled there itself.
    - Example : ClassNotFoundException (if class is not found), IOException (Occurs during input output operations)
  + Unchecked Exception :
    - Checked at runtime.
    - For this exception IDE also don’t tell
    - It not mandatory to be handle, but can be done using try catch block.
    - Example : NullPointerException, ArrayIndexOutOfBoundsException
  + How do you handle multiple exceptions in a single try-catch block?
* Only the code that seems to be risky is kept under try block. We can also use one try and multiple catch block to handle exception. In exception we can also use || operator or write the exception but all those exception should be from similar level.
  + What is the purpose of the finally block?
* In java if we open a file or some other resource, it needs to closed. If there arise any exception it does not closes out resource. So to do all theses clean up activity we use finally block. It always run, if there is an exception occurred or not.
* **String Handling:**
  + What are the immutable nature of strings in Java?
* Java String are immutable in nature. i.e. the object that are created using new keyword are created inside heap. But the String literals which are created are created inside string constant pool. And to push any object into string pool “intern” method is used.
* String should also be not manipulated.
  + Explain the difference between String, StringBuilder, and StringBuffer.
* Difference :
  + String :
    - String are immutable. I.e. cannot be changed/ manipulated.
    - Literals are stored inside String Constant Pool.
    - String are thread safe.
  + StringBuilder :
    - StringBuilder are mutable in nature.
    - Literals are not created.
    - It is not synchronized only for single threaded but not for multi thread safe.
  + StringBuffer :
    - StringBuffer are mutable in nature.
    - Literals are not created.
    - It is synchronized, available for multi threaded environment.
  + How do you reverse a string in Java?
* Convert into toCharArray(). And swap from starting to end
* **Collections Framework:**
  + What is the difference between ArrayList and LinkedList?
  + Explain the concept of generics in Java.
  + How do you sort a list in Java?
  + What is the difference between HashMap and HashSet?
* **Input/Output:**
  + How do you read and write data to a file in Java?
  + What is serialization and deserialization?
  + How do you handle file exceptions in Java?

**Advanced Java**

* **Multithreading:**
  + What is the difference between thread and process?
  + Explain the concept of thread synchronization.
  + What is the difference between wait() and sleep() methods?
  + How do you create a thread in Java?
* **Java Generics:**
  + What are the benefits of using generics in Java?
  + Explain wildcards in Java generics.
  + How do you create a generic class and method?
* **Java Collections:**
  + What is the difference between List and Set interfaces?
  + Explain the concept of iterators in Java.
  + How do you implement a custom comparator in Java?
* **Java IO and NIO:**
  + What is the difference between byte streams and character streams?
  + Explain the concept of buffers in NIO.
  + How do you read and write data to a file using NIO?
* **Serialization and Deserialization:**
  + What is the purpose of the transient keyword?
  + How do you serialize and deserialize objects in Java?