Future vs CompletableFuture:

- Completion
  - future provides an isDone() to check if the results are done, and use get() to retrieve results. But when you need to complete it manually, Future does not provide any API to do so.
  - In CompletableFuture, complete() method helps hs to manually complete a future.
- chain executions
  - CompletableFuture has the ability to chain executions with thenApply() or thenAccept that can take a method to process the result after it's available
- asynchronous computation
  - In CompletableFuture we can run some tasks asynchronously in the background

Summary

- Maven
  - a project management tool
  - can import dependencies into the project by importing repository in pom.xml
    - local: sits inside local machine, default folder location is ~/.m2
    - central: repository from the official maven community
    - remote: other artifacts deployed on the other domains
- Git
  - version control tool
- Basic Data Types
  - primitive types: int, long, char, double, float, long, byte, boolean
- String/StringBuilder/StringBuffer
  - a string object is immutable, while objects of StringBuilder and StringBuffer is mutable
  - StringBuilder vs StringBuffer: StringBuffer is thread-safe while StringBuilder is not, due to thread safety, StringBuffer has performance overhead.
- equals() and hashcode()
  - to ensure equals and hashcode contract, in a class we should override both methods when we want to define custom equals method
  - if two objects have the same content (equals return true) then their hashcode must also return the same value. However, if two objects have the same hashcode, their content might not be the same
- Java Collection
  - LinkedList vs ArrayList:
    - Add operation: Both O(1)
    - Remove Operation: ArrayList is O(n) while LinkedList is O(1)
    - Get Operation: ArrayList is O(1) while LinkedList is O(n)
  - list vs set
    - list is ordered while set is not
    - set does not allow duplicate element
    - remove complexity: list is O(n) set is O(n)/O(logn)

    - HashMap and HashSet:
        - HashSet is implemented using HashMap, each element is a key with null value

- Comparator vs Comparable
    - comparator can be used to create many sorting sequences while comparable is used to create a single sorting sequence (compare() vs compareTo())
- JVM
    - a specification that provides runtime environment in which java bytecode can be executed
    - consists of classloader, memory data area and execution engine
- Java ClassLoader
    - responsible for loading classes to JVM dynamically during runtime
        - bootstrap class loader: loading JDK internal classes, parent of all other classloader instances
        - extension class loader: child of bootstrap class loader, takes care of loading extensions of the standard core java classes
        - system class loader: child of extension class loader, loads files found in the classpath environment variable, -cp command line option
- Garbage Collector
    - regularly cleans up unreferenced objects in heap, default GC is parallel GC
    - Heap structure is divided into generations: young (eden, survivor 0, survivor 1), old, permanent. Minor GC happens in young generation, major GC happens in old generations
- Keywords
- OOP
    - JAVA is an OOP language. It centers around 4 concepts
        - Abstraction: only essential characteristics of an object are presented to users
        - Encapsulation: wrapping the implementation details in a unit
        - Polymorphism: achieved by overloading and overriding
            - method overload: methods with same name but different parameters and return types
            - method override: happens when child classes want to give a new implementation of the the same method from the parent class
        - Inheritance: A class can act as an child class by extending another (abstract) class, in doing so the child class has access to parent class methods and properties
- Exception
    - customize exception: create a class extending Exception class
    - checked vs unchecked: checked happens at compile time, unchecked happens at runtime
    - throw vs throws: throw is a keyword used in a statement, while throws used at the method definition body

- Generics
    - ensure type corrections at the compile time (type erasure)
    - use <> to specify parameter types
    - bounded generics and wildcards:
        - <T extends E> T class is a subtype of E
        - <? extends E> allows class E or sub classes of E
        - <? super T> allows class T and upper class of class T
- IO Stream
    - java.io package. stream is a continuous flow of data, usually connected to a data source, like a file or database connection. There are two kinds of streams, input and output.
- Serialization and Deserialization
    - done by implements Serializable
    - converts objects into a byte stream, deserialization is the opposite.
    - transient keyword: prevents a field to be serialized
- Java 8 features
    - Lambda expression − Adds functional processing capability to Java.
    - Method references − Referencing functions by their names instead of invoking them directly. Using functions as parameter.
    - Default method − Interface to have default method implementation.
    - Stream API − New stream API to facilitate pipeline processing.
    - Optional − Emphasis on best practices to handle null values properly.
- Java Multi-threading
    - lifecycle of a thread:
        - New − A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a born thread.
        - Runnable − After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing its task.
        - Waiting − Sometimes, a thread transitions to the waiting state while the thread waits for another thread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.
        - Timed Waiting − A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transitions back to the runnable state when that time interval expires or when the event it is waiting for occurs.
        - Terminated (Dead) − A runnable thread enters the terminated state when it completes its task or otherwise terminates.
    - Two ways to create a thread:
        - extends Thread class

        - implements Runnable
    - Runnable vs Callable
        - Runnable: overrides run()
        - Callable: overrides call()
    - ThreadPoolExecutor
        - customized thread pool
        - an extensible thread pool implementation with lots of parameters and hooks for fine-tuning.