1. Classes, objects, data types
    a. For loop types (3) `
    b. Constructors `
    c. What is an iterable Interface `
    d. Encapsulation `
    e. Constructors `
    f. Static , non static methods
2. Extending Classes, inheritance, abstract classes and interfaces
    a. Super Fathers objects and fathers constructor `
    b. Overloading and overriding `
    c. Function and interface may have default methods `
    d. Anonymous classes 3 types how to make `
    e. Local class is inner class declared inside a block `
    f. Outer objects `
    g. Interface may have colliding default methods `
    h. Access modifiers `
    i. Abstract classes `

3. Exception handling
    a. Exception vs error`
    b. How to point where an exception occurred, print a message with when exception raised `
4. Threads
    a. Life cycle of a thread `
    b. How to start and stop a thread`
    c. How to create a thread with overriding and runnable `
    d. Default method in runnable thread `
    e. Does thread implement runnable ? Yes, we only need a run method to implement runnable `
    f. How to stop a thread?   Intereput
    g. How to throw an interruptedexception`
    h. Notify `
    i. Notify all`
    j. Illegal monitor exception `
    k. How does a thread notify another thread? T. notify, and a random thread is chosen `
    l. Deadlocks `
    m. Thread starvation `
    n. Volatile
5. Lambda expression, generics and reflection
    a. Functional Interface An interface with one abstract method `
    b. How to write a lambda function `
    c. Implement interface on fly using lambda expressions`
    d. Stack using reflections `

e. Classes in reflection package `
f. Object of generic type without instantiating the generic variable
g. Generic type `