summarize missing points:

class MyClass implement interface1, 2, 3 extends AbstractClass

template -> access elements from template

JVM -> Java Virtual Machine -> type of virtual machine (Process Virtual machine)

stack, heap, PC(program counter), method area, native method stack

immutable class:

1. final -> class
2. private final fields
3. getter only, no setter
4. in getter, given referenced data type field,
5. always return deep copy dummies

double vs Double -> primitive vs referenced data type -> 8 primitive data type byte < short < int < long，float < double，boolean，char （1 byte < 2 byte < 4 byte < 8 byte，4 byte < 8 byte，1 bit{1 byte}，2 byte）

auto boxing

auto unboxing

Throwable: Exception vs Error

==============================

Hash Collision???

Employee e1 -> hash-value1

Employee e2 -> hash-value1

vs

Normal update???

Employee e1 -> hash-value1

Employee e1 -> hash-value1

hashcode + equals -> Employee extends Object -> e1.equals(e2) -> euqlas check each and every fields

employee -> id, age, name, contactInfo, email, phoneNumber

class template (Object) equals -> ==

key2 hashvalue1

HashMap key1-value1 : key1 -> hashcode function -> hashvalue1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Bucket(array)

|

V

value1

|

V

value2

|

V

vluae3

|

V

.... \* 1m records O(1) -> O(n)

java 8 -> 8/16 -> linkedlist -> black and red tree data structure

OOP ->

Inheritance -> interface + abstract class + final

Encapsulation, -> access modifier

Polymorphism, -> override vs overload

Abstraction -> interface + abstract class

S -> single responsibility

O -> Open/Closed

L -> Loskov principle

I -> Interface segeration

D -> Dependency injection