# Chapter 2: Java Core Concepts Recap

## 1. Abstract Class vs Interface

Abstract Class:  
- Can have method implementations.  
- Used when you want to provide common functionality and force subclasses to override specific parts.  
- Only one abstract class can be inherited.  
  
Interface:  
- Only method declarations (Java 8+ allows default/static methods).  
- Used for defining a contract or capability.  
- Multiple interfaces can be implemented.  
  
Use abstract classes for shared logic and interfaces for capability declaration.

## 2. import vs export in Java

import:  
- Allows the use of classes from other packages without needing to specify full package names.  
- Does NOT add file paths or manipulate runtime behavior.  
  
export (in Java modules):  
- Not a keyword in Java language, but used in module-info.java in Java 9+.  
- It exposes a package to other modules.

## 3. Object Comparison in Java

- == checks if two references point to the same object (memory address).  
- .equals() checks if two objects are logically equal (content comparison).  
- hashCode() must be consistent with equals() if overridden.  
  
In C++, struct comparison is usually field-by-field. In Java, equals() needs to be overridden for that behavior.

## 4. clone() and Cloneable Interface

- clone() is a method from java.lang.Object to make a field-wise copy.  
- The class must implement Cloneable interface or else clone() throws CloneNotSupportedException.  
- It performs shallow copy unless deep copy is manually implemented.  
- Cloneable is a marker interface (no methods).

## 5. StringBuilder to String Conversion

Use .toString() to convert a StringBuilder into a String.  
Example:  
StringBuilder sb = new StringBuilder("hello");  
String str = sb.toString();

## 6. char to String Conversion

- Cannot assign a char directly to a String variable.  
- Use:  
 - String.valueOf(char);  
 - Character.toString(char);  
 - "" + char;

## 7. Java Strings are Immutable

- Strings cannot be modified after creation.  
- You cannot do h[0] = 'x';  
- Use charAt() to read, and create new strings for modification.

## 8. Useful String Methods

- repeat(int): Repeat string N times. (Java 11+)  
- split(regex): Split string using delimiter into String[].  
- toCharArray(): Convert string into a char[].

## 9. C++ std::string vs Java String

- C++ std::string: dynamic array of char with flexible memory.  
- Java String: immutable reference type with fixed content.  
- C++ strings can be modified with [], Java requires new string construction.