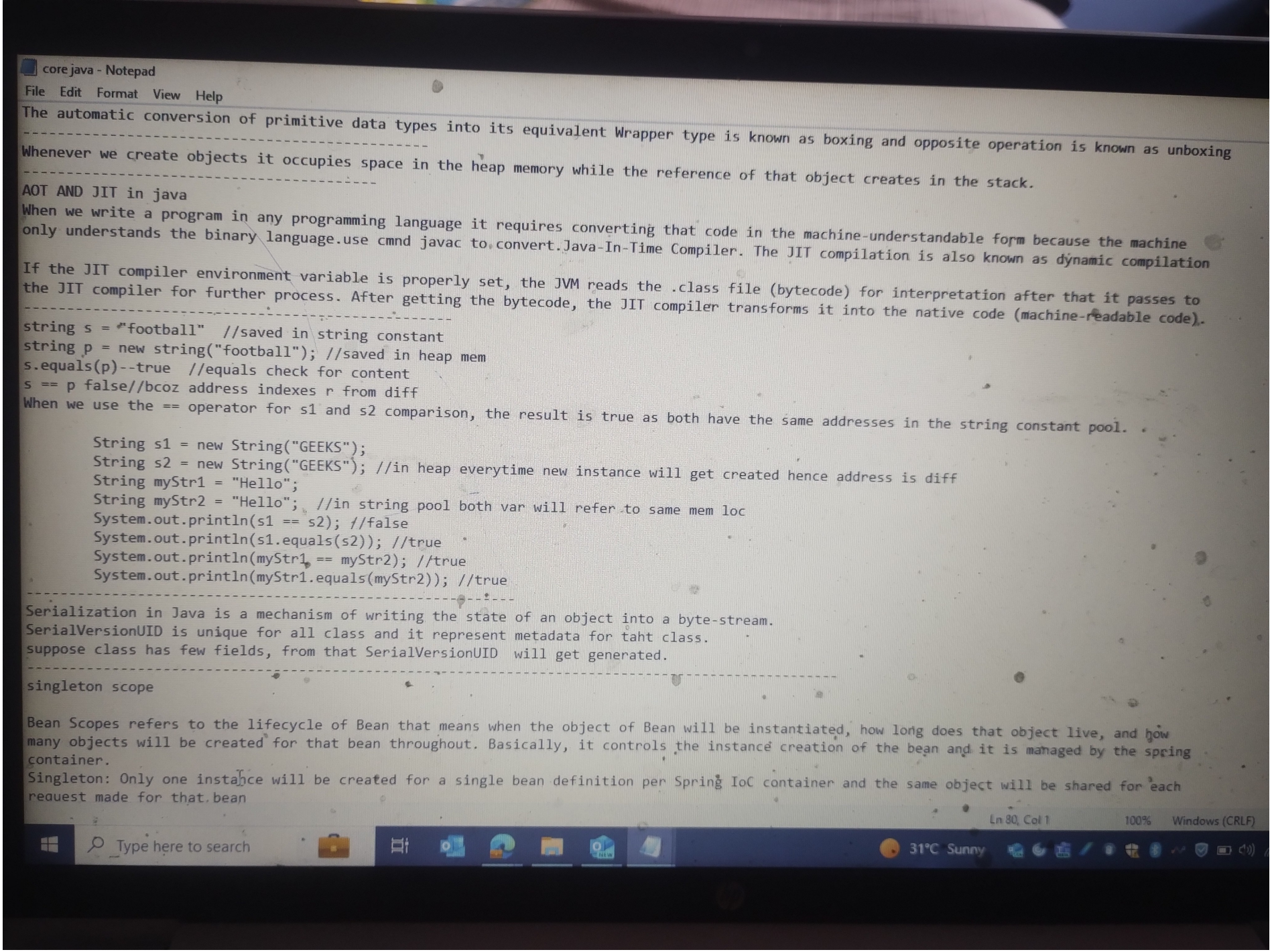


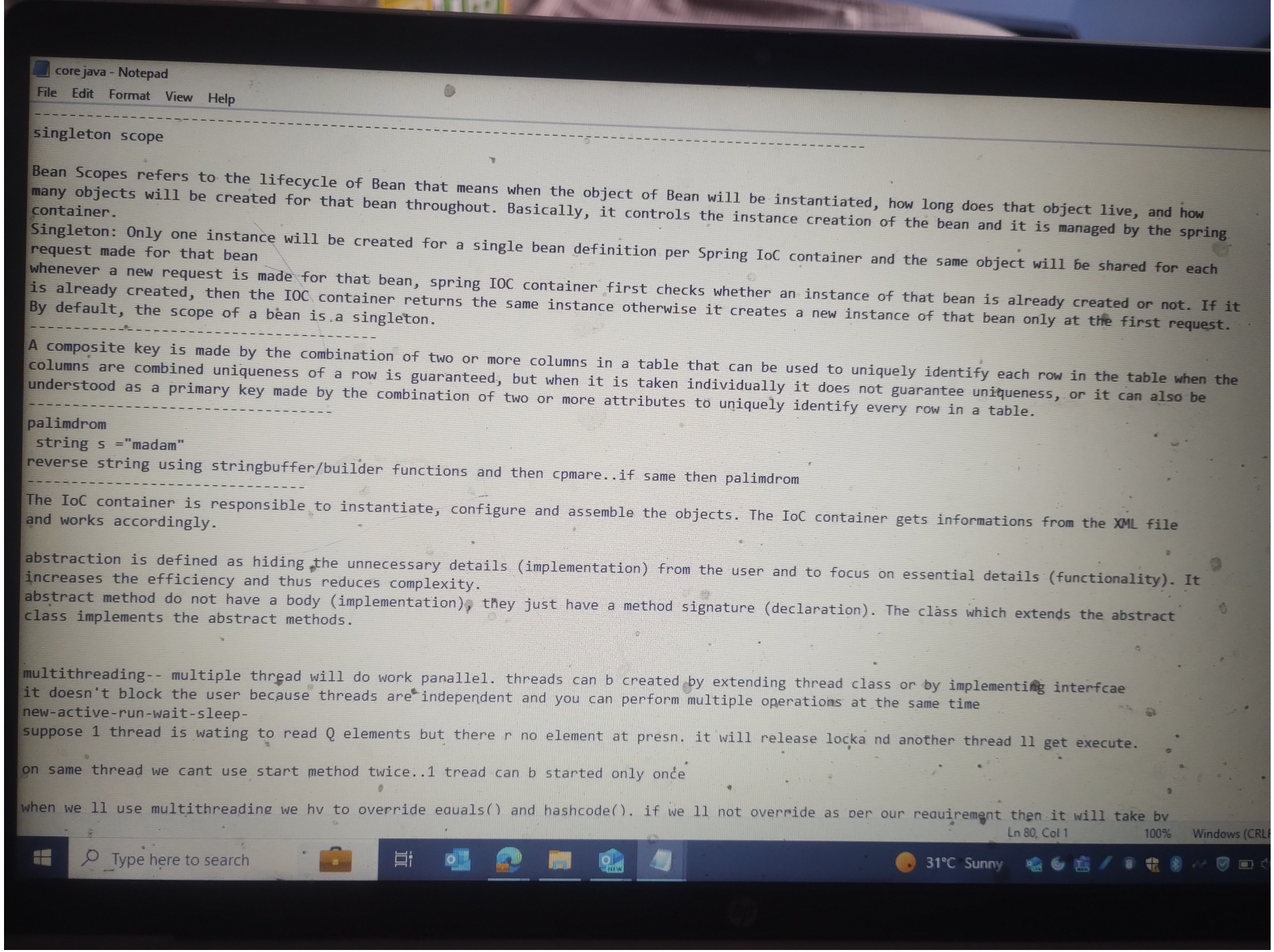
core java - Notepad

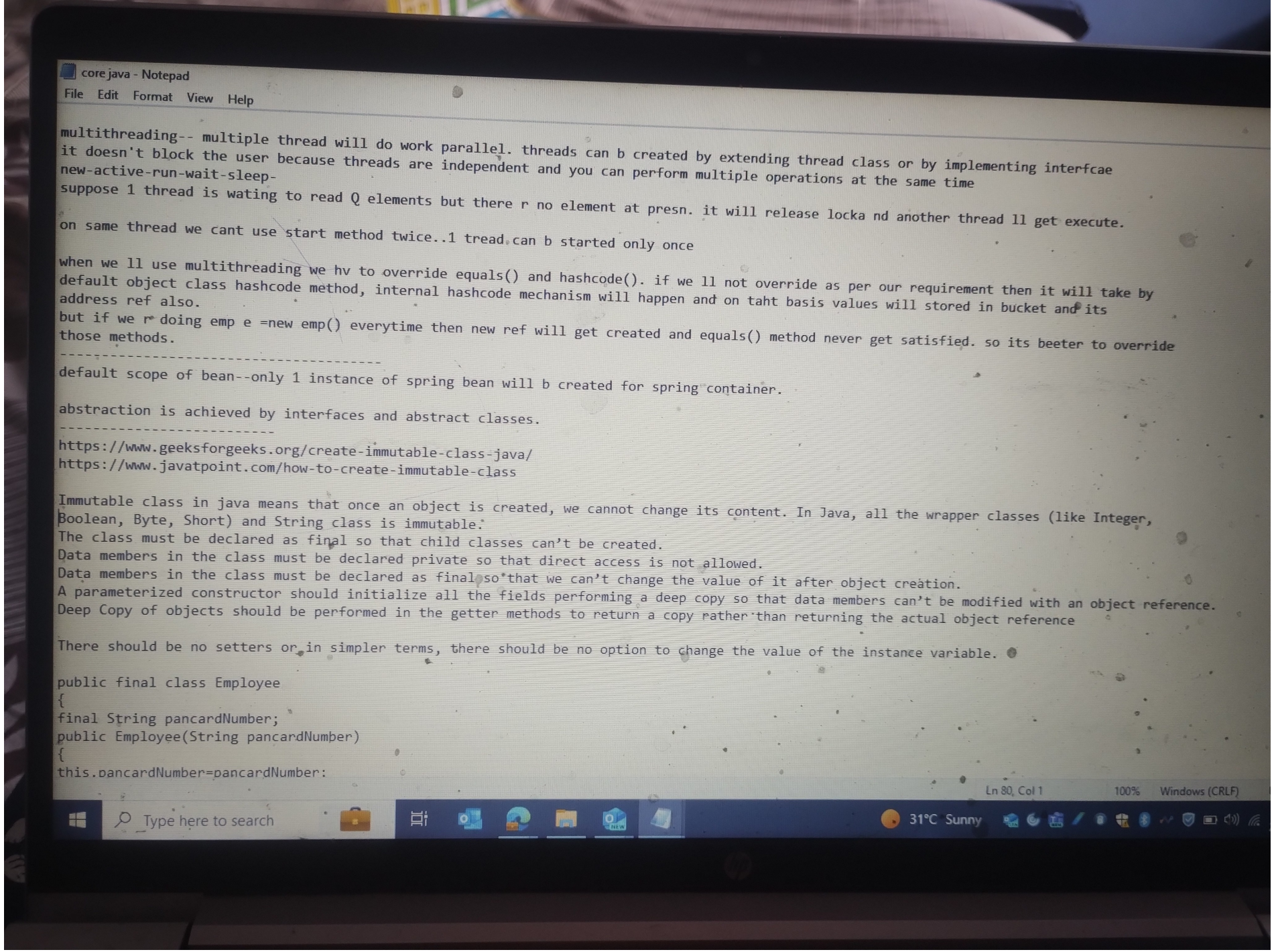
File Edit Format View Help

```
The automatic conversion of primitive data types into its equivalent Wrapper type is known as boxing and opposite operation is known as unboxing
--------------------------------------------
Whenever we create objects it occupies space in the heap memory while the reference of that object creates in the stack.
--------------------------------------
AOT AND JIT in java
When we write a program in any programming language it requires converting that code in the machine-understandable form because the machine
only understands the binary language.use cmnd javac to convert.Java-In-Time Compiler. The JIT compilation is also known as dynamic compilation

If the JIT compiler environment variable is properly set, the JVM reads the .class file (bytecode) for interpretation after that it passes to
the JIT compiler for further process. After getting the bytecode, the JIT compiler transforms it into the native code (machine-readable code).
-------------------------------------------------
string s = "football"  //saved in string constant
string p = new string("football"); //saved in heap mem
s.equals(p)--true  //equals check for content
s == p false//bcoz address indexes r from diff
When we use the == operator for s1 and s2 comparison, the result is true as both have the same addresses in the string constant pool.

        String s1 = new String("GEEKS");
        String s2 = new String("GEEKS"); //in heap everytime new instance will get created hence address is diff
        String myStr1 = "Hello";
        String myStr2 = "Hello";  //in string pool both var will refer to same mem loc
        System.out.println(s1 == s2); //false
        System.out.println(s1.equals(s2)); //true
        System.out.println(myStr1 == myStr2); //true
        System.out.println(myStr1.equals(myStr2)); //true
-----------------------------------------------------
Serialization in Java is a mechanism of writing the state of an object into a byte-stream.
SerialVersionUID is unique for all class and it represent metadata for taht class.
suppose class has few fields, from that SerialVersionUID  will get generated.
-----------------------------------------------------------------------------------
singleton scope

Bean Scopes refers to the lifecycle of Bean that means when the object of Bean will be instantiated, how long does that object live, and how
many objects will be created for that bean throughout. Basically, it controls the instance creation of the bean and it is managed by the spring
container.
Singleton: Only one instance will be created for a single bean definition per Spring IoC container and the same object will be shared for each
request made for that bean
```

Ln 80, Col 1 100% Windows (CRLF)

Type here to search 31°C Sunny





Immutable class in java means that once an object is created, we cannot change its content. In Java, all the wrapper classes (like Integ
Boolean, Byte, Short) and String class is immutable.
The class must be declared as final so that child classes can't be created.
Data members in the class must be declared private so that direct access is not allowed.
Data members in the class must be declared as final so that we can't change the value of it after object creation.
A parameterized constructor should initialize all the fields performing a deep copy so that data members can't be modified with an object
Deep Copy of objects should be performed in the getter methods to return a copy rather than returning the actual object reference

There should be no setters or in simpler terms, there should be no option to change the value of the instance variable.

```
public final class Employee
{
final String pancardNumber;
public Employee(String pancardNumber)
{
this.pancardNumber=pancardNumber;
}
public String getPancardNumber(){
return pancardNumber;
}
}
public class ImmutableDemo
{
public static void main(String ar[])
{
Employee e = new Employee("ABC123");
String s1 = e.getPancardNumber();
System.out.println("Pancard Number: " + s1);
}
}
```

The instance variable of the class is final i.e. we cannot change the value of it after creating an object.
The class is final so we cannot create the subclass.
There is no setter methods i.e. we have no option to change the value of the instance variable.