

List Of Scenarios where **var** keyword cannot be used for variable declaration.

- 1) You can't use local variable type inference with method arguments:

```
public long countNumberOfFiles(var fileList); // Compilation
//error because compiler cannot infer type of local
//variable fileList; cannot use 'var' on variable without
//initializer
```

var cannot be used for fields, method parameters, and method return types. The reason is that types in these **locations appear explicitly in class files and in Javadoc specifications**. With type inference, it's quite easy for a change to an initializer to cause the variable's inferred type to change.

- 2) You cannot initialize a **var** variable to **null**.

```
var count=null; // Compilation error because
//compiler cannot infer type for local variable
//count since any Java object reference can be null
```

By assigning null, it is not clear what the type should be, since in Java, any object reference can be null. In the following example, because there is no predefined data type for a null value, the compiler is not able to interpret a type for **count**, which would cause a compilation error.

- 3) You can't use local variable type inference with lambda expressions.

The reason is because those require an explicit target type. For example, the following causes a compilation error:

```
var z = () -> {} // Compilation error because
//compiler cannot infer type for local variable z;
//lambda expression needs an explicit target type
```