LET'S SAY WE HAVE 2 STRINGS

TWEEDLEDEE

AND

TWEEDLEDUM

LET'S SAY WE WANTED TO TEST THESE
2 STRINGS AND SEE IF THEY ARE EQUAL -
HOW MUCH WORK WOULD BE INVOLVED?

WELL, IT DEPENDS

LET'S SAY WE WERE DOING THIS IN C
OR C++, WE WOULD NEED TO COMPARE
THE 2 STRINGS ONE CHARACTER AT A TIME

THIS MAKES STRING COMPARISON
IN C OR C++ O(N) WHERE N IS THE
LENGTH OF THE STRING

BUT LET'S SAY WE ARE DOING THIS IN JAVA,
THIS WOULD BE AN O(1) OPERATION

HOW SO?

BECAUSE JAVA - LIKE MANY OTHER
MODERN LANGUAGES - MAKES USE OF
SOMETHING CALLED **STRING INTERNING**

STRINGS IN JAVA ARE "IMMUTABLE", WHICH MEANS THAT JAVA ONLY STORES ONE COPY OF EACH DISTINCT STRING VALUE

TO TEST IF TWO STRINGS ARE THE SAME, SIMPLY TEST IF THEY ARE THE SAME OBJECT ("OBJECT IDENTITY TEST")

SO - STRING VARIABLES ARE ACTUALLY REFERENCES TO THESE IMMUTABLE STRINGS

IF YOU MODIFY A STRING (VARIABLE), WHAT IS REALLY HAPPENING IS - A NEW UNDERLYING STRING VALUE IS CREATED AND THE STRING VARIABLE IS SET TO POINT TO THAT NEW VALUE

THE IMMUTABILITY OF STRINGS IS NOW A PRETTY STANDARD PART OF MANY LANGUAGES - JAVA, .NET, PYTHON, RUBY,...

THIS IS AN EXAMPLE OF

THE FLYWEIGHT PATTERN