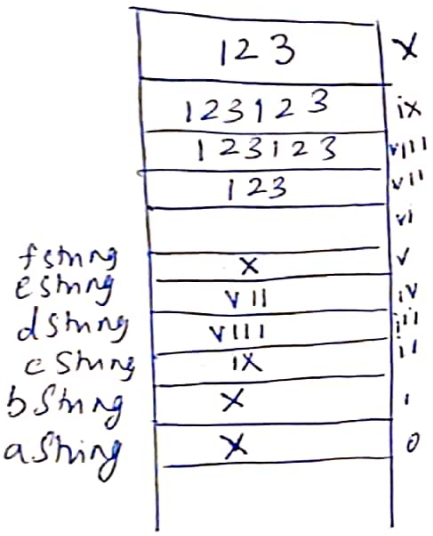
BLOCK 1



→ aString == bString (same memory location)
True

→ ~~Not~~ False (Diff memory location, diff content

→ False
dString creates a new String object @ runtime
as one is a variable. even though content is same
for dString we have a new String object

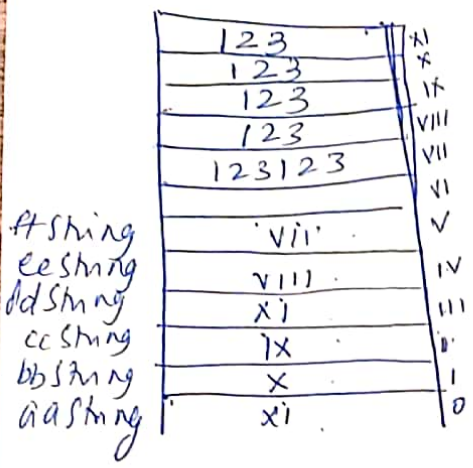
→ False
eString creates a new String object @ runtime
∴ diff from fString.

→ False
eString creates a new String object @ runtime
∴ even though both LHS & RHS content wise are same they diff address

→ False
same reason as above creates a new String object @ runtime
∴ even though both content same address differs

→ False
same reason as above - eString
creates a new String object @ runtime
∴ even though both content same address differs.

BLOCK 2



- False
  aaString = "123"
  bbString = creates a new String object
  ∴ though content same address differs

- False
  new String object created for ccString
  ∴ though content same address differs

- True
  both are strings 123

- False
  aaString is a string
  eeString a string object created
  ∴ though content same address differs

- False
  ffString is a string object
  but 123123 is a string
  ∴ address differs even though content same

- ~~No parantheses~~ False
  No parantheses is
  so Concats 1st which is not ie "13.123123" ===" 123123"
  Neither content / address will ∴ be same

- 14. 123123123123