

For First Block

0	counter
3	thenumberthre
123	i
123	ii
123	iii
123	iv
123123	v
i	a
ii	b
iii	c
i	d
iv	e
v	f

1. False

B is declared as a new string, therefore, not equal.

2. False

C is declared as a new aString therefore is not the same.

3. True

The method1() returns "123" and so does the aString.

4. False

As method2() returns a new string. aString and eString are not the same.

5. False

fString uses method2() which creates a new string and that is why 123123 of fString is stored at a different location.

6. False

The brackets are missing here and therefore the whole statement before = is compared to 123123.

7. 123123123123

Here we print fString + "123123"; ie. 123123123123.

For Second Block

123	i
123123	ii
123123	iii
123	iv
i	a
i	b
ii	c
iii	d
iv	e
i	f

8. True

aString and bString are literals and are stored at the same location.

9. False

aString and cString are totally different strings and cannot be stored in the same memory.

10. False

cString 123123 is different from dString as d was created with aStrings address.

11. False

eString has the int 3 and will create a new storage in the memory.

12. False

eString was created with thenumberthre and hence it has different and new memory allocation from the literal "123"

13. False

Strings computed by concatenation at run time are newly created and therefore distinct.

14. False

Strings computed by concatenation at run time are newly created and therefore distinct.