Eg : String\_Eg113

// go through the code

Do we have append() in java.lang.String

Ans : False ( we have concat() )

Eg: String\_Eg114

// go through the code

Eg: String\_Eg115

// go trough the code

Eg: String\_Eg116

// go through the code

Eg: String\_Eg117

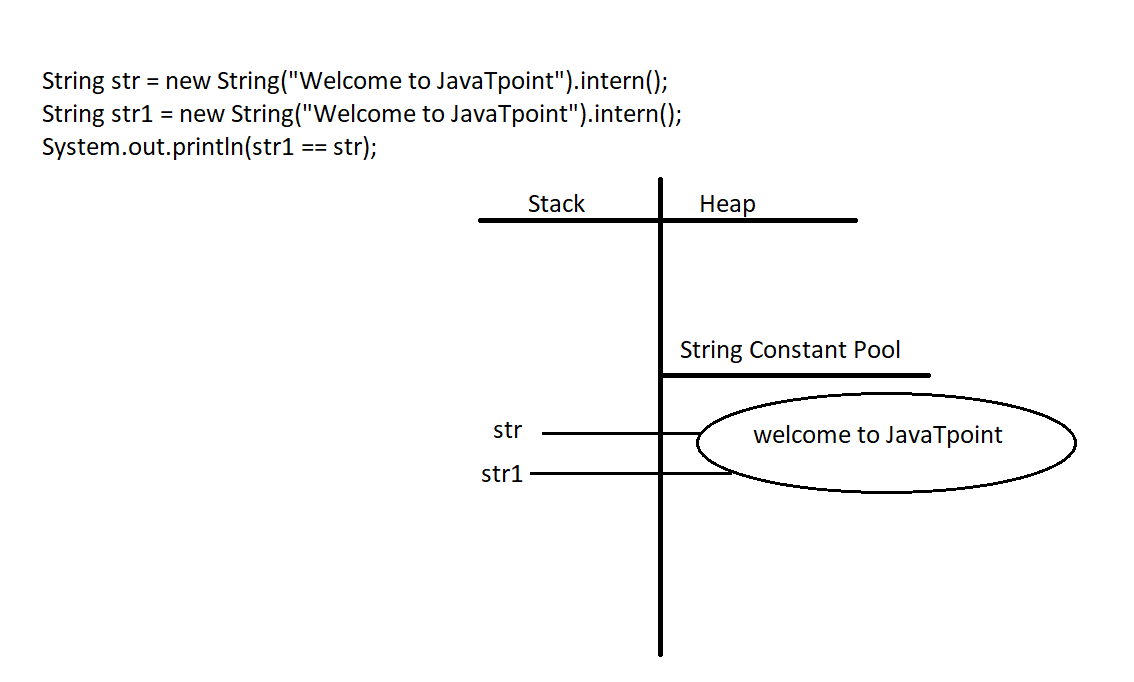
//go through the code

Intern()

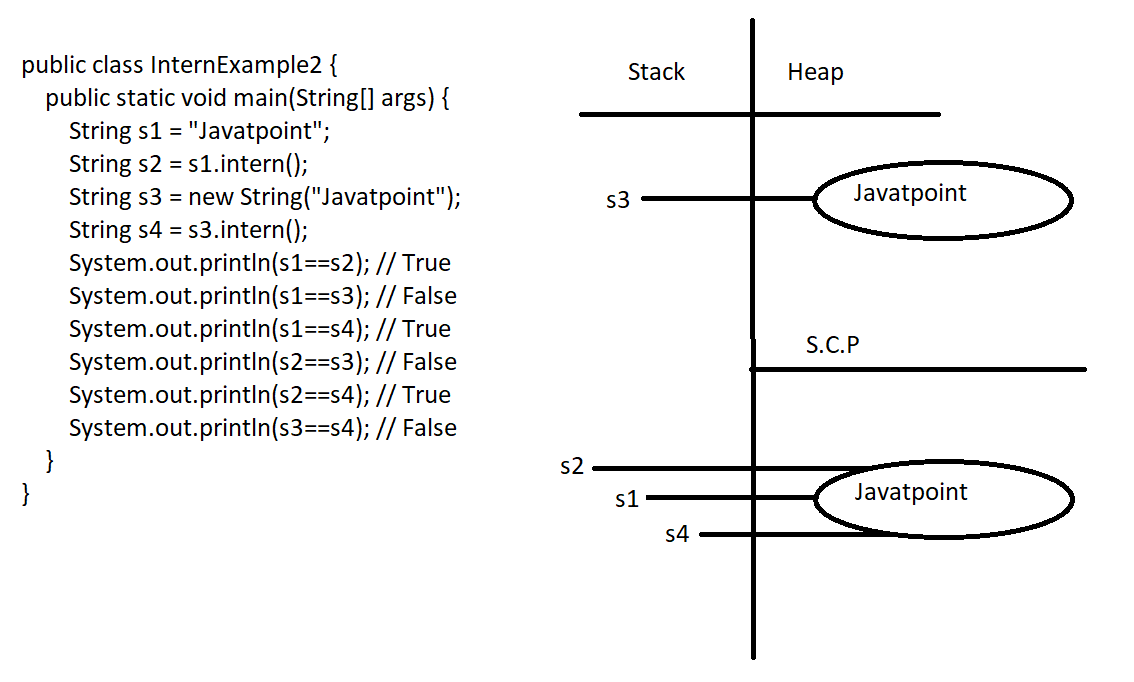
 In order to put the strings in the string pool, one needs to call the **intern()** method. Before creating an object in the string pool, the JVM checks whether the string is already present in the pool or not. If the string is present, its reference is returned.

Eg: String\_Eg118

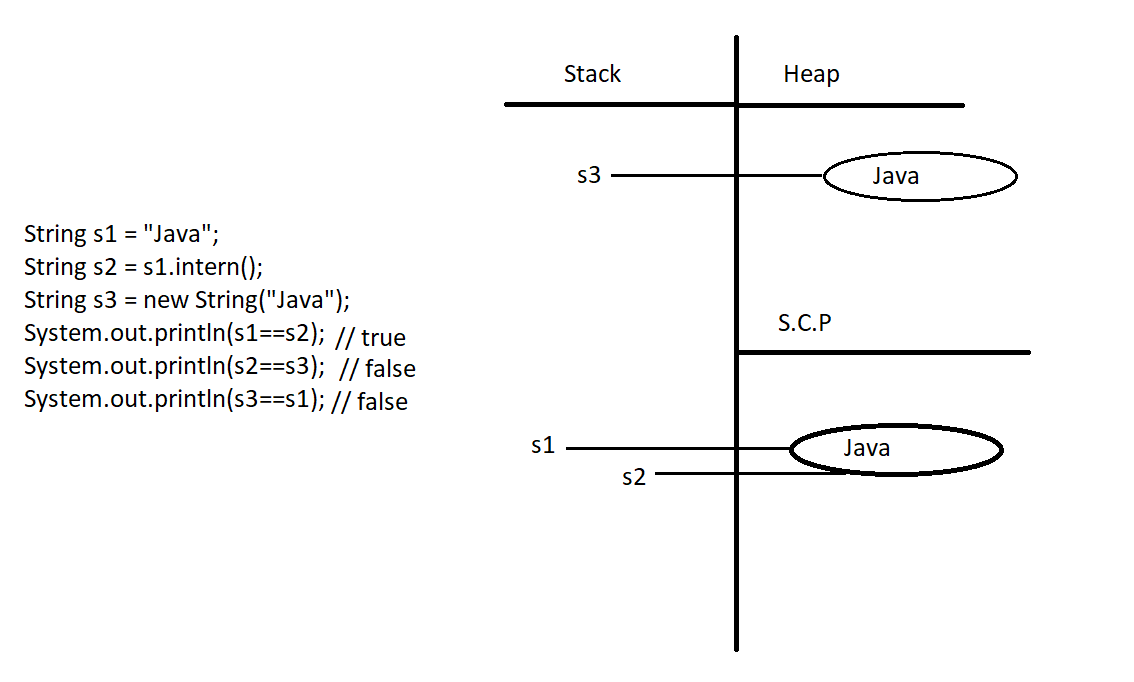
// go through the code



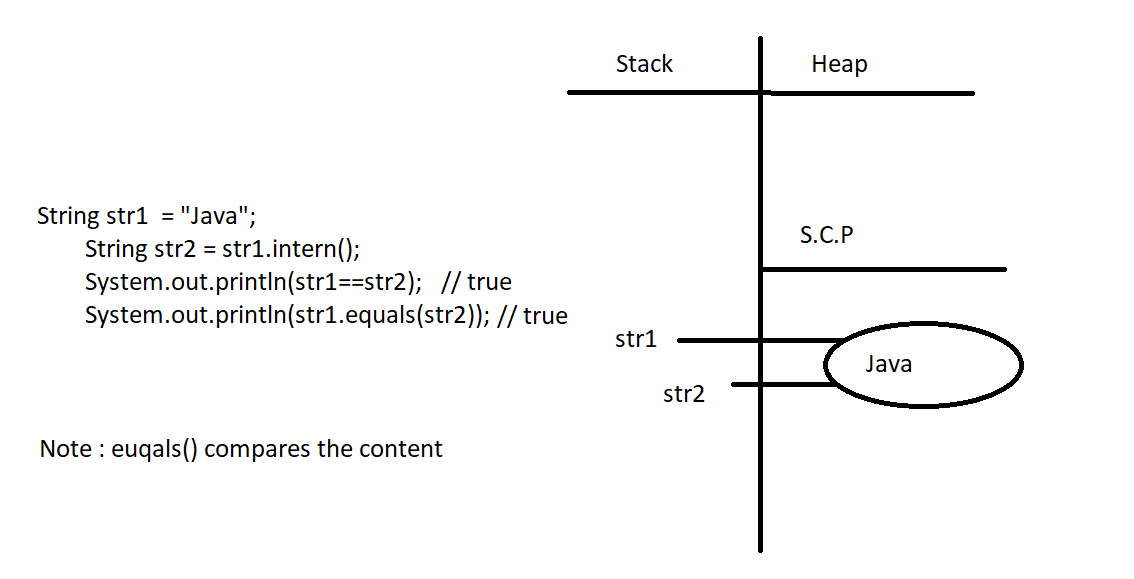
Eg: String\_Eg119



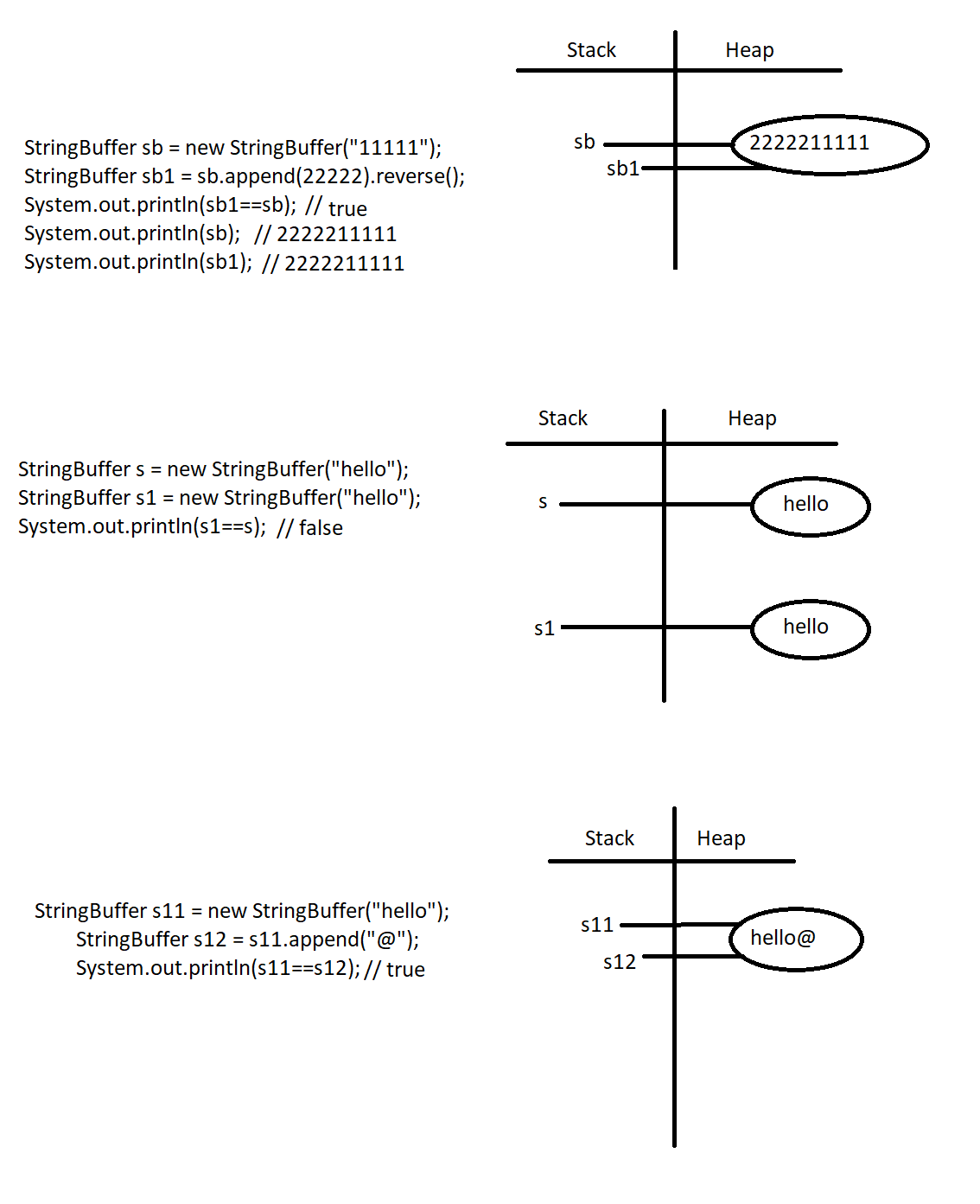
Eg: String\_Eg120



Eg: String\_Eg121



Eg: String\_Eg122



In StringBuffer if we try to append() data to another StringBuffer Object then both the references are pointed to the same object.

Eg: String\_Eg123

// go through the code

Does StringBuilder and StringBuffer have intern() method.

No // only String class have intern() method

Eg: String\_Eg124

// go through the code

Eg: String\_Eg125

// go through the code

Eg: String\_Eg126

// go through the code

Eg: String\_Eg127

// go through the code

Eg: String\_Eg128

// go through the code

Eg: String\_Eg130

// go through the code

Eg: String\_Eg131

//go through the code

Eg: String\_Eg132

// go through the code

Eg: String\_Eg133

// go through the code

Eg: String\_Eg134

// go through the code

Eg: String\_Eg135

// go through the code

Eg: String\_Eg136

//go through the code.

Eg: String\_Eg137

// go through the code

Eg: String\_Eg138

// go through the code

Eg: String\_Eg139

// go through the code

Eg: String\_Eg140

// go through the code

Eg: String\_Eg141

// go through the code

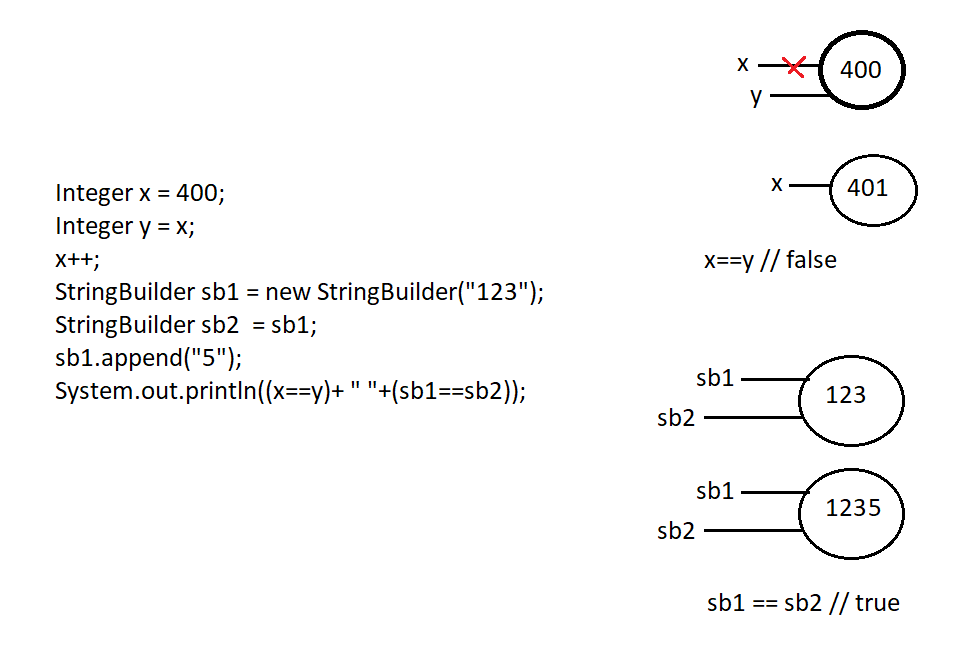
Eg: String\_Eg142

// go through the code

Eg: String\_Eg143

Note: Wrapper classes and String class is immutable (changes will not be made in the same object instead new object will be created with the new change)

StringBuilder, StringBuffer are mutable ( changes will be made in the same object) .



Eg: String\_Eg144

Arrays : to find the length of the array we use length property

Eg: int[] data = { 10,20,30}

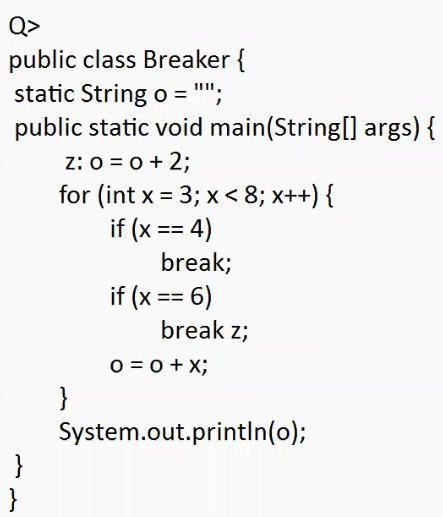
System.out.println(data) // 3

Strings : to find no of characters present in String we use length property

Eg: String[] names = { “ Pavan” , “Charan”,”Anand”};

System.out.println(names[0].length); // gives the total no. of characters present in 0th index

Eg: String\_Eg145



Here z is a label, we can’t use label for a statement. Label is used only with loop, to come out of the loop. So it results in compile time error.