<Question 1> Explain the differences among the following getTop() functions.

T Stack<T>::getTop() const
T& Stack<T>::getTop()
const T& Stack<T>::getTop() const

T Stack<T>::getTop() const

This header expects a value returned back since it's a T type(template), also the function is constant so we can't modify the value of the TOP value of the object in our linked list stack.

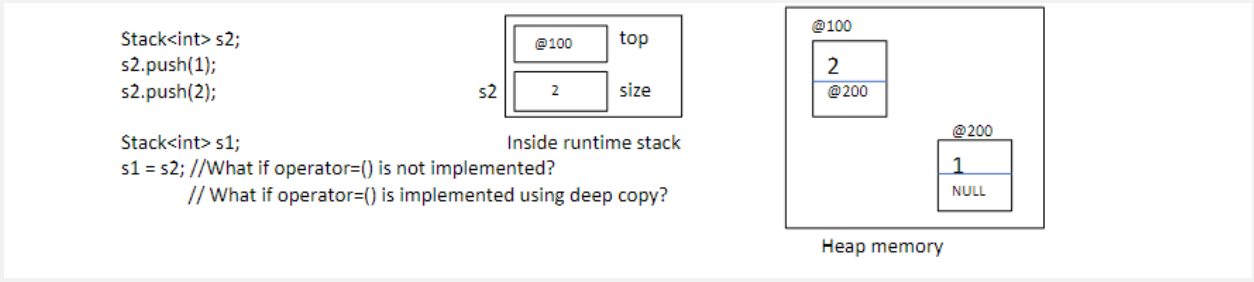
T& Stack<T>::getTop()

This header has T(template) as pass by reference so we can change the value of the TOP value to the Invoked object by using the address.

const T& Stack<T>::getTop() const

This header receives a **const T&** that is a constant <template> value that is passed by reference so that a copy is not made in the run time stack. The function is also const so we can't modify the value of the TOP object in our linked list stack.

<Question 2> Explain the difference between the shallow copy and deep copy by using the following scenario



Making a shallow copy will copy the stack from the object in the RHS, but since they share the same memory address any change you make in one object will also occur to the other object. Making a deep copy lets the new object have its own stack, so any change won't affect the object being copied from. We c while simultaneously traversing the new stack and using another pointer for copying each node.

.