How to send back data when popping and pushing a viewcontroller from a navigationcontroller stack

**Pushing and popping VCs**

Push and pop controllers as follows:

self.navigationController?.pushViewController(nextVC, animated: true)

if let navController = self.navigationController {

navController.popViewControllerAnimated(true)

}

In case of pop above we have checked that navigationcontroller exists before unwrapping (proper way).

Navigationcontroller should exist in all views added to the controller (start off in appdelegate).

There are also methods of navigationcontroller that allow to move to particular VCs in stack. Pushing alone adds to top of stack, popping removes top and shows next one down in stack.

**Sending data on push and pop to destination VCs**

If pushing, can send data to next VC by placing code in prepareforSegue() function. This is called before the segue that moves to next VC is initialized. To get the name of the destination VC you can use something like navigationcontroller.destinationviewcontroller or something like that.

If popping, then you can send data back by delegation. i.e. make sure that the delegation VC adheres to a protocol and then in the one you are to pop, define a delegate variable that is of type of the protocol (but optional), and back when the VC you are in that you want to pop was first instantiated and pushed onto the stack, set the delegate property of it to be the destination VC that you will be popping to. Don’t forget that any parameters or methods that you want to access from that destination VC need to have been declared in the protocol that it conforms to.

Altering stack before segues

This code suggests a way in which it is possible to manipulate the stack before transitioning (i.e. how to pop off top, and move to a new one that has just been pushed)

If I understand you correctly, you have a stack of view controllers:

A (root) - B - C - D - E

And you want it to become:

A (root) - F

Right? In that case:

NSArray \*viewControllers = self.navigationController.viewControllers;

NSMutableArray \*newViewControllers = [NSMutableArray array];

// preserve the root view controller

[newViewControllers addObject:[viewControllers objectAtIndex:0]];

// add the new view controller

[newViewControllers addObject:viewThreadController];

// animatedly change the navigation stack

[self.navigationController setViewControllers:newViewControllers animated:YES];

(Essentially creates an array of ordered Viewcontrollers – with all the ones want to keep from old stack in it, followed by new one, and then uses a navigationController command that animates from previous stack to this new one gotten from the array)