CS 61A   Structure and Interpretation of Computer Programs
Summer 2016

INSTRUCTIONS

- You have 25 minutes to complete this quiz.

- The exam is closed book, closed notes, closed computer, closed calculator.

- Mark your answers **on the quiz itself**. We will *not* grade answers written on scratch paper.
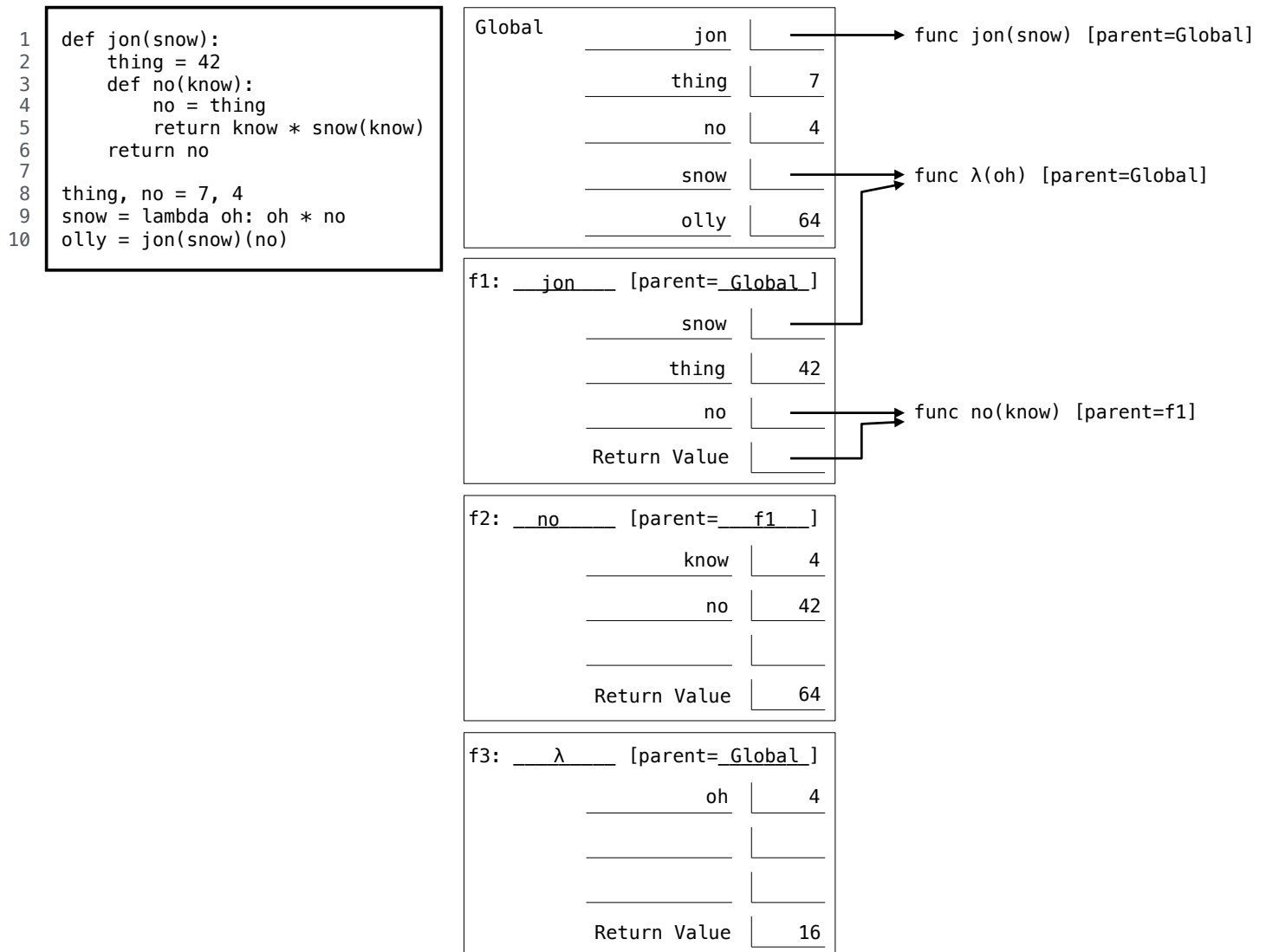
Last name	
First name	
Student ID number	
Instructional account (`cs61a-_`)	
BearFacts email (`_@berkeley.edu`)	
TA	
Name of the person to your left	
Name of the person to your right	
All the work on this exam is my own. **(please sign)**	

1. (5 points) The Great Frame is Terrifying

(a) **(4 pt)** Fill in the environment diagram that results from executing the code below until the entire program is finished, an error occurs, or all frames are filled. *You may not need to use all of the spaces or frames.*

A complete answer will:

- Add all missing names and parent annotations to all frames.
- Add all missing values created or referenced during execution.
- Show the return value for each local frame.

```
1   def jon(snow):
2       thing = 42
3       def no(know):
4           no = thing
5           return know * snow(know)
6       return no
7
8   thing, no = 7, 4
9   snow = lambda oh: oh * no
10  olly = jon(snow)(no)
```

Global
- jon → func jon(snow) [parent=Global]
- thing | 7
- no | 4
- snow → func λ(oh) [parent=Global]
- olly | 64

f1: __jon__ [parent=_Global_]
- snow |
- thing | 42
- no → func no(know) [parent=f1]
- Return Value |

f2: __no__ [parent=__f1__]
- know | 4
- no | 42
- |
- Return Value | 64

f3: ___λ___ [parent=_Global_]
- oh | 4
- |
- |
- Return Value | 16

(b) **(1 pt)** If we change line 8 to `thing, no = 12, 10` , what value does `olly` get bound to in line 10?

1000. It doesn't matter what `thing` is bound to, `jon(snow)(no)` will return the cube of `no`.