University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Fall 2011 Instructor: Dan Garcia 2011-09-19

CS10 Quest

Last Name	
First Name	
Student ID Number	
cs10- Login First Letter	**a b c d e f g h i j k l m**
cs10- Login Last Letter	**a b c d e f g h i j k l m** **n o p q r s t u v w x y z**
The name of your LAB TA (please circle)	**Aijia Glenn Luke Navin Rabbit Samir**
Name of the person to your Left	
Name of the person to your Right	
All my work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS10 who have not taken it yet. ***(please sign)***	

Instructions

- Don't panic!
- This booklet contains 4 numbered pages including this cover page. Put all answers on these pages; don't hand in any stray pieces of paper.
- Please turn off all pagers, cell phones and beepers. Remove all hats and headphones.
- Fill in the front of this page and put your login on every sheet of paper.
- You have 50 minutes to complete this exam. The Quest is closed book, no computers, no PDAs, no cell phones, no calculators, but you are allowed one double-sided set of notes. There may be partial credit for incomplete answers; write as much of the solution as you can. When we provide a blank, please fit your answer within the space provided.

Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**TOTAL**
Points	**2**	**2**	**2**	**2**	**2**	**2**	**3**	**5**	**20**
Score									

Question 1a : Why are *abstraction violations* bad? (I.e., what is the danger of a system peeking below an abstraction line, seeing how something is implemented, and making use of that, say for optimizing performance or cost)

Question 2a : Of the four stages of the 3D graphics pipeline listed below, circle the one that is the *most different* between 3D video games and 3D feature films:

Modeling Animation Lighting-and-Shading Rendering

Question 3a : Circle all of the following that are *not* functions:

Question 4a : Draw lines from the programming paradigms to the facts that match them best

a) Functional	1) "What" not "How"
b) Imperative	2) Built through *composition* of blocks
c) OOP	3) Works via *message passing*
d) Declarative	4) Like a *recipe*. Do this, then that, then that, etc.

Question 5a : In his "Program or Be Programmed" talk, Douglas Rushkoff argues that we should "direct technology or run the risk of lettering ourselves be directed by it and those who have mastered it". He goes on to say we should understand the *motivations* behind technology, which is often very different from what you think it is. His compelling example was that a naive person may think that Facebook is there to help you make friends. "Not so!" he says -- what does he believe is Facebook's motivation?

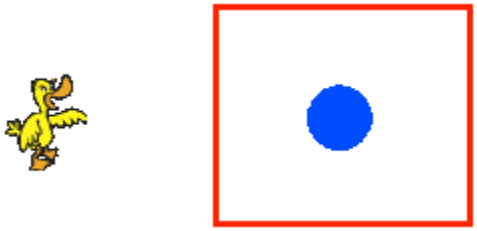
Question 6a : The function `foo` is used in the following way (and you have no idea what `a`, `b`, or `c` are). What can you say about the *domain* and *range* of `foo`?

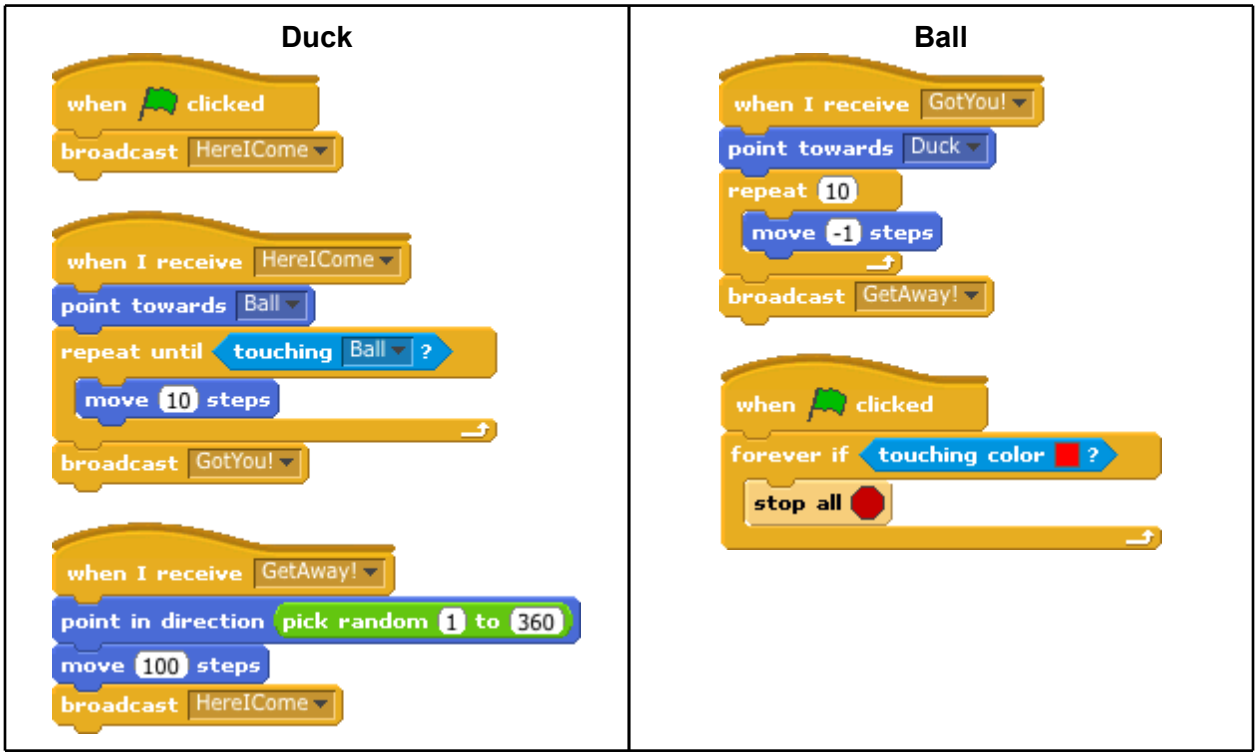
Domain of `foo` (first argument):

Domain of `foo` (second argument):

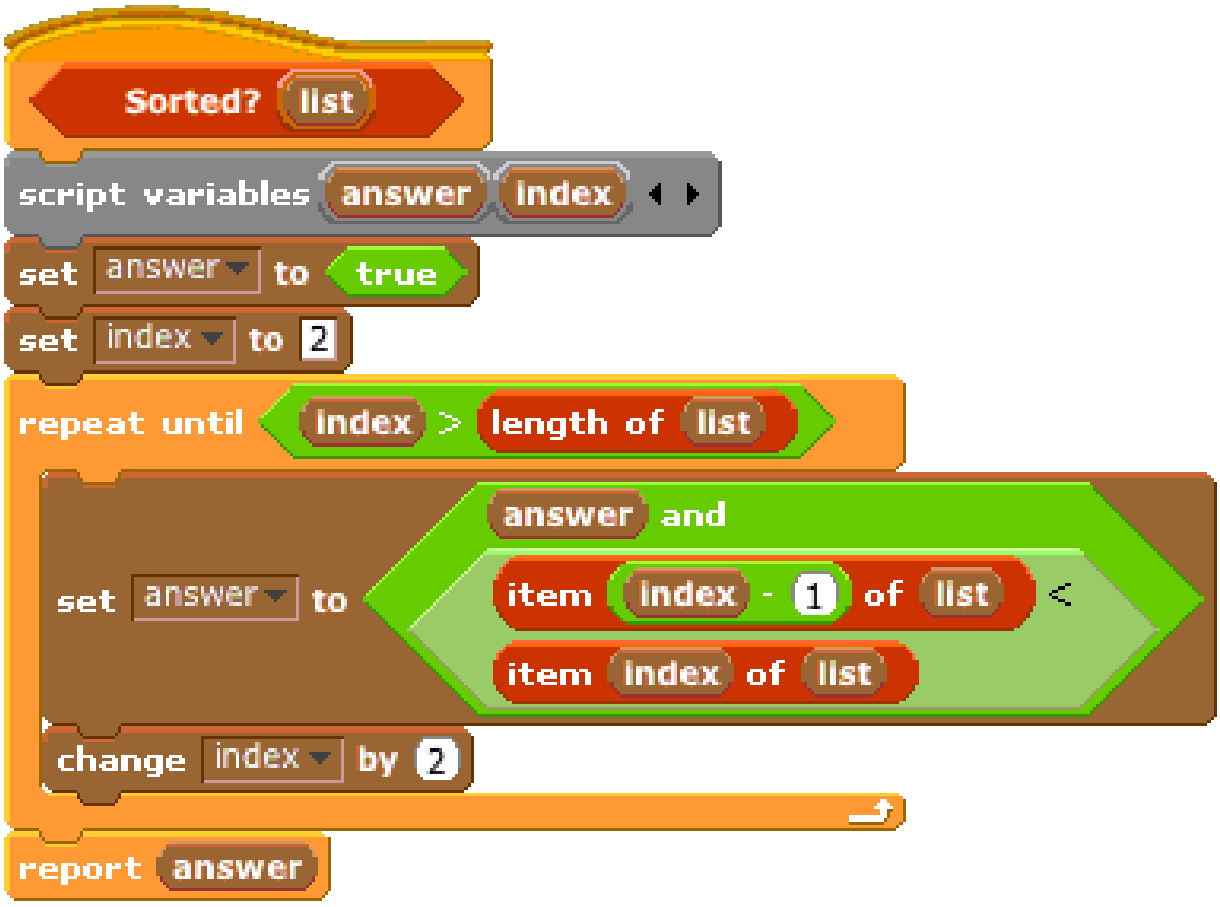
Range of `foo`:

Question 7a : Describe what happens in the following "play" between the Duck and the Ball which start as shown below, and whose scripts are shown below.



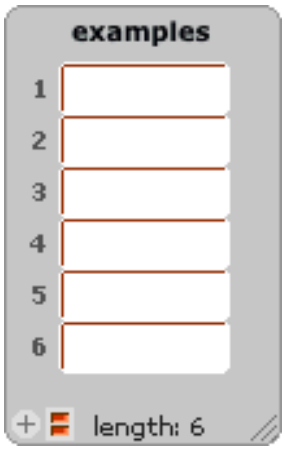


Question 8a : We're trying to write a function that returns true when the input list is sorted in ascending order (i.e., every element is smaller than the ones after it, like the sequence "3 6 7"). Unfortunately, we think there's a bug in the code.



a) Complete the next sentence.
"The **examples** list below (fill this in, you might not need all six slots, just cross off the slots you don't need) is *the shortest list that does **not** trigger the bug* when passed in to the function. It returns (circle one) TRUE FALSE but it *should* return (circle one) TRUE FALSE ."

b) Describe what fixes you would make so the block works as desired.



University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Fall 2011 Instructor: Dan Garcia 2011-09-19

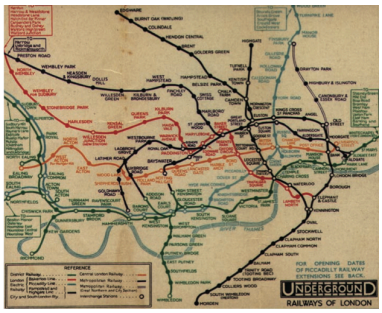
CS10 Quest

Last Name	
First Name	
Student ID Number	
cs10- Login First Letter	**a b c d e f g h i j k l m**
cs10- Login Last Letter	**a b c d e f g h i j k l m** **n o p q r s t u v w x y z**
The name of your LAB TA (please circle)	**Aijia Glenn Luke Navin Rabbit Samir**
Name of the person to your Left	
Name of the person to your Right	
All my work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS10 who have not taken it yet. ***(please sign)***	

Instructions

- Don't panic!
- This booklet contains 4 numbered pages including this cover page. Put all answers on these pages; don't hand in any stray pieces of paper.
- Please turn off all pagers, cell phones and beepers. Remove all hats and headphones.
- Fill in the front of this page and put your login on every sheet of paper.
- You have 50 minutes to complete this exam. The Quest is closed book, no computers, no PDAs, no cell phones, no calculators, but you are allowed one double-sided set of notes. There may be partial credit for incomplete answers; write as much of the solution as you can. When we provide a blank, please fit your answer within the space provided.

Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**TOTAL**
Points	**2**	**2**	**2**	**2**	**2**	**2**	**3**	**5**	**20**
Score									



Question 1b : Some argue that applying abstraction to visualizations is *always* a good thing. Consider the original 1933 London Underground map (on the left) and the redesign (on the right) – the detail removed allows riders to focus on the important aspects of their trip, i.e., the order of the stops and the connecting stations. "Hold on!" you say, "there were useful parts of the original map that were lost!" What were they?

Question 2b : In traditional 2D cell animation (like Snow White), every stage of the production pipeline was done by hand. As we have moved to 3D animated movies (like Pixar films), many stages are still done by hand. Of the four stages of the 3D graphics pipeline listed below, circle all those that *could be* automated by a computer program (to save humans tedious work):

Modeling Animation Lighting-and-Shading Rendering

Question 3b : Give one example from the reading why software is eating the world.

Question 4b : Draw lines to match the computing pioneer with what they did:

a) Ivan Sutherland	1) Created system to turn recreational into useful work
b) Alan Turing	2) Came up with programming languages completeness theory
c) Shigero Miyamoto	3) Father of information and communications theory
d) Luis von Ahn	4) Developed the 1st OOP system, the 1st graphical interface, & the 1st non-procedural language. Father of Computer Graphics!
e) Claude Shannon	5) First person elected to the Gaming Hall of Fame
f) Gordon Moore	6) Founder of Intel, observed exponential growth of the density of integrated circuits on chips.

Question 5b : In his "Program or Be Programmed" talk, Douglas Rushkoff argues that we should "direct technology or run the risk of lettering ourselves be directed by it and those who have mastered it". He goes on to say we should understand the *motivations* behind technology, which is often very different from what you think it is. "Blown to Bits" also makes this point. A naive person may think that *supermarket loyalty cards* are there to help you save money to thank you for your steady business, what do they say is their *real* motivation?

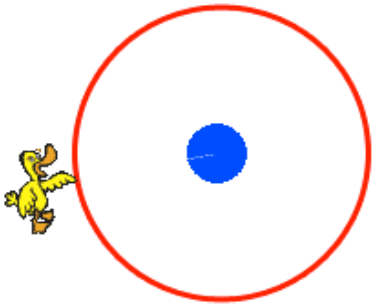
Question 6b : The function **`bar`** is used in the following way (and you have no idea what **`a`**, **`b`**, or **`c`** are). What can you say about the *domain* and *range* of **`bar`**?

Domain of **`bar`** (first argument):

Domain of **`bar`** (second argument):

Range of **`bar`**:

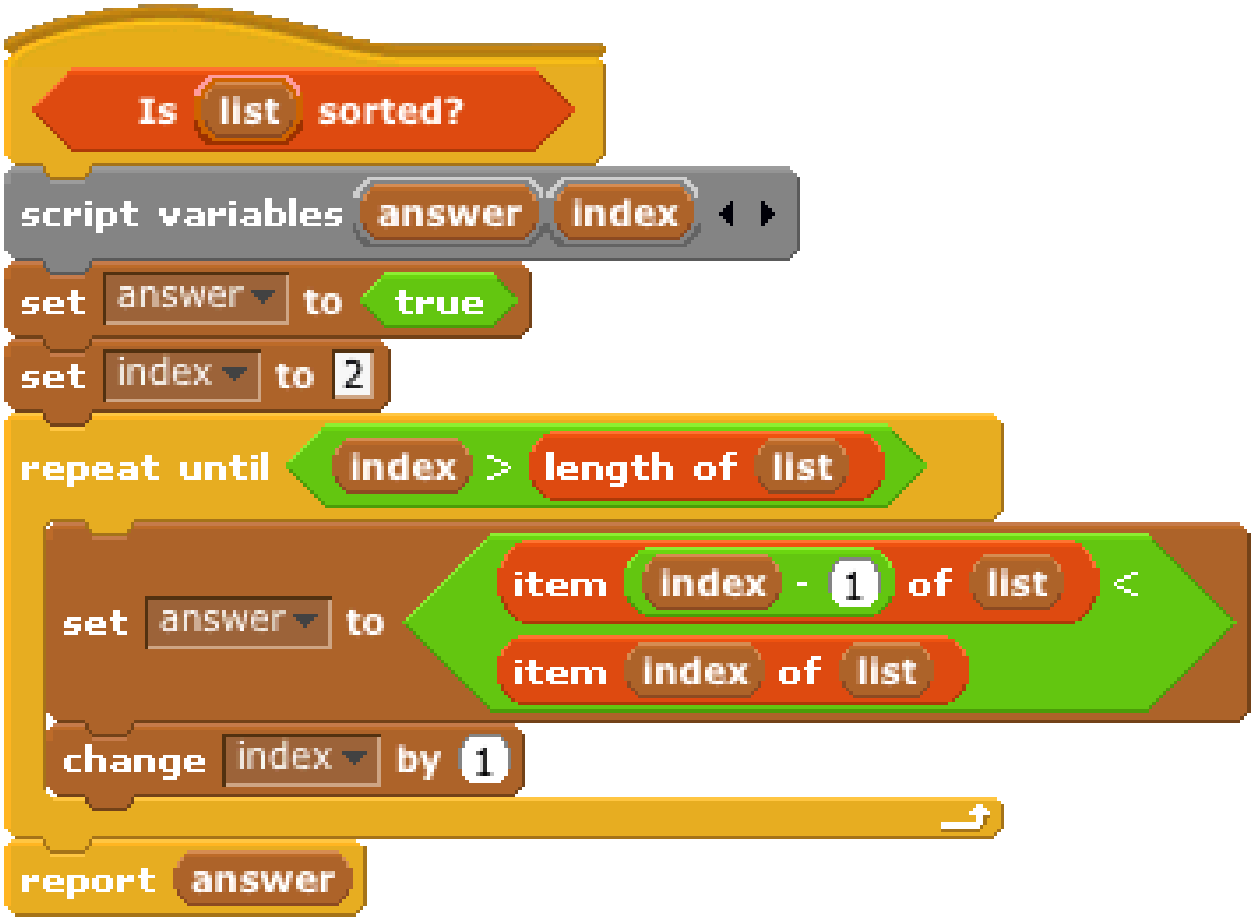
Question 7b : Describe what happens in the following "play" between the Duck and the Ball which start as shown on the right, and whose scripts are shown below (Duck on left, Ball on right).



Duck

Ball

Question 8b : We're trying to write a function that returns true when the input list is sorted in ascending order (i.e., every element is smaller than the ones after it, like the sequence "3 6 7"). Unfortunately, we think there's a bug in the code.



a) Complete the next sentence.
"The **examples** list below (fill this in, you might not need all six slots, just cross off the slots you don't need) is *the shortest list that does **not** trigger the bug* when passed in to the function. It should return (circle one) TRUE FALSE but it does return (circle one) TRUE FALSE ."

b) Describe what fixes you would make so the block works as desired.