University of California, Berkeley – College of Engineering
Department of Electrical Engineering and Computer Sciences
Fall 2010   Instructors: Dan Garcia and Brian Harvey   2010-09-20

CS10 Quest

Last Name	*Solutions*
First Name	*Quest*
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)	**Jon** **Luke**
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.
- Question 0 (1 point) involves filling in the front of this page and putting 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	0	1	2	3	4	5	6	7	8	Total
Points	1	8	8	8	8	7	5	15	20	80
Score										

Question 1 : In a hybrid car, the electric motor can also serve as a generator to recharge the battery, by letting the wheels turn the motor instead of the other way around. This takes energy away from the wheels, slowing the car. To make this happen, you push the brake pedal. But you're not really applying the brakes -- you're charging the battery from the car's energy of motion. In one sentence, why didn't they make a separate control for this, instead of giving an extra meaning to the brake pedal?

> **8** points for mentioning that user only sees car slowing down; mechanism is below the abstraction barrier. [Has to point out that battery charging isn't inherently user-visible.]
> **6** points for mentioned that it keeps the abstraction that the brake pedal slows the car.
> **4** points for mentioning "abstraction."
> **2** points for saying that it keeps things simple.

Question 2 : In one sentence, what was the main point of Luis von Ahn's *Games With a Purpose (GWAP)* project, and what was one example?

> To have humans help solve problems that are hard for computers to solve (and to train AI algorithms) by playing games and generating data. [5 pts] Example: *ESP* for labeling images, *Verbosity* for finding out commonsense facts (e.g., sock is a kind of clothing), *Peekaboom* for locating objects in images, or *Phetch* which annotates images with descriptive paragraphs [3 pts]

Question 3 : Aside from the commonly-cited (and debated) increase in violent behavior, what is another negative aspect of video games?

> Addiction (e.g., "gamer's wife"), repetitive-stress injuries, and time taken away from academics has led to a reduction in performance for many youth. [8 pts]

Question 4 : In one sentence, refute either of the following statements with a simple example from the reading: *"Technology is Good, with no downsides"* or *"Technology brings opportunities with no risks"*

> Many answers are possible, but the best answers cite a specific example from one of the readings, e.g., "Blown to Bits talks about a teenage girl who killed herself because she was dumped by a nonexistent MySpace friend created by an adult to harass her." A less good example would be, e.g., "identity theft," without citing anything from the readings.

> **8** points for explicitly citing a reading, or being really specific.
> **7** points for anything else except...
> **4** points for being much too vague or providing no detail at all.

Question 5 : Of the four we mentioned, which is the most powerful programming paradigm?

> All of them are equally powerful.

Question 6: Draw the shape that results from the following code:

> The correct answer for this question would have been to draw a square (4 sides).
>
> Points for this question were given cumulatively. **1** point was given for correctly expressing the value of the first repeat statement (or showing that you understood it later). **2** points were given for getting the mod's correct. **1** point was given for expressing an understanding of the number of times the second repeat loop executed. **1** point for drawing the correct shape (a square).

Question 7: You record the growth of a pandemic flu going around your city in the following table comparing the *total number of people with the flu (i.e., infected)* with that day.

Day (input)	0	1	2	3	4
Infected (output)	1	3	9	27	81

a) Write a block `Flu` to estimate how many people will be infected on any given day, based on the assumption that the number infected on any given day will be three times that of the day before. Your answer should be written in such a way that would allow another CS10 student to translate it into proper Scratch code. Here is an example of a call to `Flu`:

> There are many possible solutions to this problem. One perfectly acceptable answer would be:

```
Flu (Day)
script variables (Infected) ▶
set Infected ▾ to 1
repeat (Day)
    set Infected ▾ to (3 * Infected)
report (Infected)
```

10 points if you communicated the idea of needing to find an exponent of 3. **7** points if you performed some sort of (incorrect) repeated operation on a value. **5** points if you only performed a single math operation. **3** points for creating some sort of block.

b) If, say, on day 15 the number infected is (shudder) exactly the same as the city's population, on what day will it be *a third* of the city's population?

5 points for day 14.

Question 8 : The block below tries to find all the numbers in a list that are greater than a specific number, and put them in another list. However, there is one bug.


```
find all numbers larger than (num) in (list) and put them into (other list)
set index ▾ to 1
repeat (length of (list))
    if (item (index) of (list) > (num))
        add (item (index) of (list)) to (other list)
    change index ▾ by 1
```

a) What would the buggy block produce for the `other list` (which starts out empty) if you provided the list below as the input `list` and the number `4` as `num`? Write your answer beside the list.

The correct answer would have been a list with a single element (5).

6 points if you had a list containing only the number 5. If you had a list of five elements, all of which were five, you got **4** points; all other answers got **no** points.

b) As the block is currently written, for a given `num`, write one sentence that describes *all* the `list`s that will **not** trigger the bug.

All lists in which any elements bigger than `num` are listed first. (because as soon as a number is not bigger than it, it stops incrementing `index` and nothing else gets copied over)

8 points for the correct answer. Many of you noted that the bug would not be triggered if the list were already in decreasing order. While this is true, it does not describe *all* the possible lists that would not trigger the bug; you got **6** points for this answer. You got **4** points if your answer showed that you had an idea, but the answer was not precise or concise.

c) Describe how you would fix this so the block works as desired.

Move "`change index by 1`" down one "element", i.e., out of the `if` (but still in the `repeat`).

6 points for the correct answer. Many of you asked that the "if" block be replaced with an "if-else" block, and that the "`change index by 1`" block be placed inside the "else"-bracket. This does not quite work: it would cause the index variable to remain at the position of the first number that was greater than `num`, without progressing. You got **4** points for this answer.

By the way, though you were not docked off for this, many answers assumed, for the entire question, that `num` was hardwired to 4. This is not true in general; it was only true for the first part.