CS118
Algorithm Development Exercises

1. Suppose you have a set 10 of boxes on a table. Eight of the boxes are empty, but in one of the boxes is a **peach** and in another box is an **apple**. You would like to swap the peach and the apple – have the apple be where the peach is currently and have the peach end up being where the apple is currently. Unfortunately, you've broken one arm and cannot hold both of them concurrently. To avoid contamination, the fruit should never be placed anywhere other than a box, and each box may contain only one fruit. Provide a written ENGLISH description of the process for swapping the contents of the two boxes.

For the rest of the exercises, assume that you can only move one item at a time and that each box may contain a maximum one piece of fruit.

2. Provide a symbolic description that follows the written description from #1. Assume that at the beginning of the exercise the apple is in box 5 and the peach is in box 8. All other boxes are empty.

Use arrows (→) to indicate the action of placing into a box; and indicate a box or contents of a box by its number.

For example, taking the contents from box 3 and moving those contents to box 4 would look like this: 3 → 4


Apple


Banana


Cherry


Durian


Emblic (Indian Gooseberry)

3. Suppose now that you have five fruits: apple, banana, cherry, durian, and emblic. You decide to give a name for each box that will hold a specific fruit – so you have boxes A, B, C, D, and E. Show the symbolic description of putting the fruit into its associated box – just use the name of the fruit to represent the fruit itself.

4a. Each of the labeled boxes, A-E, contains a fruit although it may or may not be the correct fruit. Box X is empty to begin. Write an English algorithm with sub-tasks which guarantees that each of boxes A-E contains a fruit which is NOT the correct fruit. This method will involve sub-tasks which involve testing. Performing the sub-tasks (in the correct order) should guarantee the original goal is met. Describe this method in English, one sub-task on a line. Since we know how to swap any two boxes, your English sub-tasks should use this ability simply by saying (for example) "swap box A with box B". The first sub-task is provided:

if A contains an apple, then swap box A with box B

4b. Your English description in #4a should have used "swap" - for example, "swap box A with box B". Now rewrite #4a here, but breakdown each "swap" task into child-tasks – provide English descriptions for the three steps that actually happen to accomplish a "swap" between boxes. To be able to do this, you'll need an empty box designated as box X.

When writing algorithms, we often have to make decisions – these involve testing (comparisons).

To describe the decision process symbolically, we need to describe what is being compared:

to describe *being the same as* use two equal signs (==);
to describe *not being the same as*, use bang equal (!=).

For example, if I want to describe the decision process for checking to see if box A contains an apple, I would write:

```
A == apple
```

If I want to describe the decision process for checking to see if box A does *not* contain an apple, I would write:

```
A != apple
```

To perform an action based upon the comparison, use a colon (`:`). So if I want to place the contents of A into B *only if* A contains an apple, I would write:

```
A == apple: A → B
```

I can perform multiple actions by separating them by commas:

```
A == apple: A → B, X → A
```

I can *nest* decisions, also – this means to make another decision based upon a previous decision. For example, if I want to move the contents of B to X but only if both A and B do not contain an apple:

```
A != apple: B != apple: B → X
```

4c. Use the previous symbolic language with the new additions to it shown in the box to the left to provide the symbolic form of the algorithm specified in 4a.

5a. For this exercise, each of the labeled boxes begin by containing the corresponding fruit. Describe in English a method that when executed will guarantee that each of the original five boxes contains a fruit, but each <u>does NOT</u> contain the correct fruit associated with that box. [This is different from #4 in that the starting assumption is different.]

5b. What are the problem-solving tools [INPUT, OUTPUT, DEFINE, TEST, REPEAT] and information used by each sub-task?

5c. Use the sub-tasks you created in #5a to provide a symbolic algorithm. Be sure that each step of the symbolic algorithm matches the English version, and that it ends up performing the exact sub-task specified.

English	Symbolic

6. Suppose that somebody has come along and mixed up all of the fruits. Each of the five boxes (A, B, C, D, and E) contains a fruit, but it is not the correct fruit. Box X is empty. Write an English algorithm with sub-tasks which guarantees the fruits end up in the correct boxes. For each sub-task, provide also the symbolic form of the algorithm in the column to the right.

7. Each of the labeled boxes, A-E, and X, may contain one of the fruits; however, exactly one box is empty – and it may be any of the boxes. (Note in the decision syntax description provided above there is no syntax that tests for an empty box – so don't attempt to compare a box to 'empty'.) Your assignment is to write an English algorithm with sub-tasks which makes X the empty box. Use as few steps in your algorithm as you can. For each sub-task, provide also the symbolic form of the algorithm. (Not all rows need be filled – do only as many tasks as required.)

English	Symbolic