

Computer Science - Problem Solving Framework

Applying the input/processing/storage/output model to the problem!

Givens?

What is given to you in the problem description? e.g. ARRAYS, VARIABLES, DATA STRUCTURES, EXISTING FUNCTIONS/METHODS

Inputs?

These will include any data that is a “given” NOT necessarily literal user input

What needs to be input by the user?

Does the user input require any validation?

Outputs?

What variables need to be output?

What messages need to be output?

Storage?

What data needs to be stored? How does it need to be stored?

What data is required but NOT GIVEN? New variables/data structures

What intermediate variables do we need to calculate the output?

Processing?

What processing is required to create the required outputs from the inputs?

Sequence: Inputs before outputs; intermediate before final e.g. we need to find the Max and Min before we calculate the range

Selection: What logical conditions are required? Which structure(s) best represent all cases in a problem? if ... else if ... else, case of

Calculation: e.g. Average = SumTotal/Count

Iteration: e.g. while, for, repeat until

13. The names of students attending a science fair were recorded in a stack data structure as each one arrived.

...
Troy
Mia
Jane
Rick
Ryan
Abed
Zara
Sophie

The first item stored in the stack was “Sophie”.

Note that “Troy” is currently in position 0 in the stack.

- (a) Construct the pseudocode that will search the stack for a specific name, and output its position in the stack. You may assume that all names in the stack are unique.

[5]

```
GIVENS:      STACK
             STACK.push();
             STACK.pop();
             STACK.isEmpty();

INPUTS:      SPECIFIC_NAME;
OUTPUTS:     "Not found";      NAME_POSITION
STORAGE:     boolean found = false;
             int current_position = 0;
SEQUENCE:    input SPECIFIC_NAME;
             search STACK;
             OUTPUT;
SELECTION:   If (STACK.pop() == SPECIFIC_NAME)
             If found else
CALCULATION: NAME_POSITION = CURRENT_POSITION;
             CURRENT_POSITION ++;
REPETITION: WHILE NOT STACK.isEmpty() AND NOT FOUND
```