

NOTE: This is an exam in Systematic Program Design. In all your work on this exam you must follow all applicable design recipes and design rules. Unless we say otherwise you must show all design recipe elements in your work.

You may use any unambiguous abbreviations in your exam. So you can abbreviate check-expect as c-e, and you can make other abbreviations as long as they are unambiguous.

Finally, please write neatly, and write in a reasonable size. It is easier for the grader to read your work and understand it if what you write is neat.

;

PROBLEM 1:

Design a function that takes 2 strings and produces the longest string. The name of the primitive that produces the length of a string is string-length.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

⚠ This page left blank for workspace if you need it.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

PROBLEM 2:

Write out the template and the template rules used for the following type comment. Follow all applicable template rules and be sure that your template and the rules used are both in the same order as the type comment.

```
;; Porridge is:  
;; - Number[<0]  
;; - "just right"  
;; - Number[>0]
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

PROBLEM 3:

The following program has one inconsistency. By which we mean one part of the program is inconsistent with the rest. To answer this problem you should NEATLY cross out JUST THAT ONE part and NEATLY write the correction beside it.

NOTE:

- You should only correct one part of the program. CHANGING MORE THAN ONE PART OF THE PROGRAM WILL BE MARKED INCORRECT.
- We expect you will spend 5–10 minutes reading and thinking and then less than a minute writing.
- Remember that a good way to review code is to systematically work through the relevant recipes, checking each part of what is written against the relevant prior recipe steps.

```
(define-struct cloud (x y sl))
;; Cloud is (make-cloud Integer Integer Integer)
;; interp. A cloud with center x and y in pixels.
;;          Clouds are square, sl is side length in pixels.

(define C1 (make-cloud 100 200 10)) ;at 100, 200; side length 10

(define (fn-for-cloud c)
  (... (cloud-x c)
        (cloud-y c)
        (cloud-sl c)))

;; Template rules used:
;; - compound: 3 fields
;; - atomic non-distinct: Integer
;; - atomic non-distinct: Integer
;; - atomic non-distinct: Integer

;; Cloud -> Integer
;; produce the area of a cloud sl^2
(check-expect (cloud-area 3) (sqr 3))

;(define (cloud-area c) 0) ;stub

;;<Template from Cloud>

(define (cloud-area c)
  (sqr (cloud-sl c)))
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

PROBLEM 4:

For this problem you should assume the existence of a new primitive type called `Date`, as well as a function called `date<?` that consumes two dates and produces true if the first is earlier than the second. Use the `D1`, `D2` and `D3` constants to write your check-expects for functions operating on dates.

First read the partial program below. Then complete the instructions described in the box at the end.

```
(define D1 (string->date "September 19, 1979"))
(define D2 (string->date "March 1, 1980"))
(define D3 (string->date "July 31, 1980"))
```

```
(define-struct student (fn fn dob id))
;; Student is (make-student String String Date Natural)
;; interp. A student with first and last name, date of birth, and ID number.
(define S1 (make-student "Hermione" "Granger" D1 84223))
(define S2 (make-student "Ron" "Weasley" D2 342342))
(define S3 (make-student "Harry" "Potter" D3 12392))
```

```
(define (fn-for-student s)
  (... (student-fn s)
        (student-ln s)
        (student-dob s)
        (student-id s)))
```

```
;; Template rules used:
;; - compound: 4 fields
```

```
;; ListOfStudent is one of:
;; - empty
;; - (cons Student ListOfStudent)
;; interp. A list of students
(define LOS1 empty)
(define LOS2 (cons S1 (cons S2 (cons S3 empty))))
#;
(define (fn-for-los los)
  (cond [(empty? los) (...)]
        [else
```

```
(... (fn-for-student (first los))  
      (fn-for-los (rest los))))))
```

```
;; Template rules used:  
;; - one of: 2 cases  
;; - atomic distinct: empty  
;; - compound: (cons Student ListOfStudent)  
;; - reference: (first los) is Student  
;; - self-reference: (rest los) is ListOfStudent
```

;

PROBLEM:

Design a function that consumes a list of students and a date, and produces all the students born before the given date.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

 This page left blank for workspace if you need it.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

🔗 This page left blank for workspace if you need it.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder