```
;; ********************************************************************
;; Question 1
;; a)
(define distance (+ (abs (- 9 5))
                    (abs (- 3 6))))

;; b)
(define a (sqrt (+ (sqr 52)
                   (sqr 16)
                   (- (* 2 52 16 (cos 115))))))

;; c)
(define c 299792458)
(define E (sqrt (+ (* (sqr 1) (expt c 4))
                   (* (sqr 0) (sqr c)))))

;; ********************************************************************
;; Question 2
;; a)
(define (Stirling n)
  (* (expt n (+ n 1/2))
     (exp (- 1 n))))

;; b)
(define (HM x y z)
  (/ 3
     (+ (/ 1 x)
        (/ 1 y)
        (/ 1 z))))

;; c)
(define g 9.8)
(define (height v t)
  (- (* v t)
     (* 1/2 g (sqr t))))
```

```
;; ******************************************************************
;; Question 3
;; We definitely want to see defined constants here
;; a)
(define metres/mile 1609.344)
(define seconds/hour 3600)

(define (mph->m/s mph)
  (* (/ mph seconds/hour) metres/mile))

;; b)
(define metres/Smoot 1.7018)
(define seconds/nanocentury 3.15576)

(define (mph->S/nc mph)
  (/ (* (/ mph seconds/hour)
        seconds/nanocentury
        metres/mile)
     metres/Smoot))

;; ******************************************************************
;; Question 4
;; a)
(define particip-weight 5/100)
(define assn-weight 20/100)
(define mt1-weight 10/100)
(define mt2-weight 20/100)
(define fe-weight 45/100)

(define particip-grade 100)  ; full participation marks

(define (final-cs135-grade fe-grade mt1-grade mt2-grade assn-grade)
  (+ (* particip-weight particip-grade)
     (* fe-weight fe-grade)
     (* mt1-weight mt1-grade)
     (* mt2-weight mt2-grade)
     (* assn-weight assn-grade)))

;; b)
(define target-grade 60)

(define (final-cs135-exam-grade-needed mt1-grade mt2-grade assn-grade)
  (/ (- target-grade
        (+ (* particip-weight particip-grade)
           (* mt1-weight mt1-grade)
           (* mt2-weight mt2-grade)
           (* assn-weight assn-grade)))
     fe-weight))
```