[REAL, GRADED] Quiz 1- Requires Respondus LockDown Browser

Due 16 Oct at 13:54 **Points** 100 **Questions** 16
Available until 16 Oct at 13:54 **Time limit** 50 Minutes
Requires Respondus LockDown Browser

Instructions

This exam is ONLY to be taken in Tech Auditorium during your scheduled exam time. Any other testing environment without explicit permission from the instructor will be considered a violation of our academic honesty policy.

- Please remember that you only get 1 attempt on this exam. You will have 50 minutes to complete it from the time that you start unless you have testing accommodations that allow for extra time.
- **Quiz Glossary (THIS LINK WORKS IN THE LOCKDOWN BROWSER)** **(https://bain-cs111.github.io/course-files/quizzes/q3_glossary_compact.pdf)**

In part 1, if you get the answer *exactly* right, canvas will mark it as CORRECT automatically. If there is any slight variation it will mark it as INCORRECT. After the exam is over, we will be going through each question and doing manual grading. For example, if you write (list of str) and Canvas expected (listof string), canvas will automatically mark your answer wrong. However, when we go in to manually grade, we would mark the above answer as fully correct.

In part 2 and part 3, we will be grading the answers completely by hand. What we are looking for is that you understand what was wrong with the code and know how to fix it. We can utilize line number(s) as a reference point, so that you can indicate where the error occurred, but more important will be your rewrite. We will grade these with the same leniency and partial credit that we would in a printed exam.

If you encounter any issues while taking the exam, please raise your immediately!

This quiz was locked 16 Oct at 13:54.

Attempt history

	Attempt	Time	Score
LATEST	Attempt 1	47 minutes	100 out of 100

Score for this quiz: **100** out of 100

Submitted 16 Oct at 13:48

This attempt took 47 minutes.

Question 1

0 / 0 pts

RULES

1) You may not communicate with anyone (electronically or in person) during this exam. You must turn off your phone and other nearby devices after successfully logging into Canvas.

2) You may not use any resources during this exam. You may only use your computer in the Lockdown browser. You may not use any printed/written notes/books or any other devices/tools while taking this exam with TWO EXCEPTIONS:

- You may use a printed Quiz glossary (remember you can use the electronic version by clicking on the link provided in each set of instructions and opening it in a new tab).
- You may use scratch paper and a pen/pencil to take notes and sketch out solutions.

****

The following statement is an excerpt from the Northwestern Provost's Office "Academic Integrity: A Basic Guide"

A. Basic Standards of Academic Integrity

Registration at Northwestern requires adherence to the University's standards of academic integrity. These standards may be intuitively understood, and cannot in any case be listed exhaustively; the following examples represent some basic types of behavior that are unacceptable:

1. **Cheating:** using unauthorized notes, study aids, or information on an examination; altering a graded work after it has been returned, then submitting the work for regrading; allowing another person to do one's work and submitting that work under one's own name; submitting identical or similar papers for credit in more than one course without prior permission from the course instructors.

2. **Plagiarism:** submitting material that in part or whole is not entirely one's own work without attributing those same portions to their correct source.
3. **Fabrication:** falsifying or inventing any information, data or citation; presenting data that were not gathered in accordance with standard guidelines defining the appropriate methods for collecting or generating data and failing to include an accurate account of the method by which the data were gathered or collected.
4. **Obtaining an Unfair Advantage:** (a) stealing, reproducing, circulating or otherwise gaining access to examination materials prior to the time authorized by the instructor; (b) stealing, destroying, defacing or concealing library materials with the purpose of depriving others of their use; (c) unauthorized collaborating on an academic assignment (d) retaining, possessing, using or circulating previously given examination materials, where those materials clearly indicate that they are to be returned to the instructor at the conclusion of the examination; (e) intentionally obstructing or interfering with another student's academic work (f) recycling one’s own work done in previous classes without obtaining permission from one’s current instructor or (g) otherwise undertaking activity with the purpose of creating or obtaining an unfair academic advantage over other students' academic work.
5. **Aiding and Abetting Academic Dishonesty:** (a) providing material, information, or other assistance to another person with knowledge that such aid could be used in any of the violations stated above; (b)providing false information in connection with any inquiry regarding academic integrity; or (c) providing(including selling) class materials to websites that sell or otherwise share such materials – including homework, exams and exam solutions, submitted papers or projects, as well as original course materials (for example, note packets, power point decks, etc.). In addition to violating Northwestern’s policies on academic integrity, such conduct may also violate University policies related to copyright protection.
6. **Falsification of Records and Official Documents:** altering documents affecting academic records; forging signatures of authorization or falsifying information on an official academic document, grade report, letter of permission, petition, drop/add form, ID card, or any other official University document.
7. **Unauthorized Access to computerized academic or administrative records or systems:** viewing or altering computer

records, modifying computer programs or systems, releasing or dispensing information gained via unauthorized access, or interfering with the use or availability of computer systems or information.

If you agree to follow these rules, please type "I agree" in the following blank

Correct!

I agree

orrect Answers

"I agree"

I agree

Give the type of each of the following expressions. If more than one expression is given, assume that each one is executed, in order, and give the type of the value for the **last** expression.

- If it is a **primitive type** such as a number, string, Boolean or image (picture), just give the type.
- If it is a **record type (struct)**, just give the name of the record type. For example, if it's an album object, just say "album"
- If it is a **list**
  - If all the elements of the list are the same type, say "(listof *type*)" where *type* is the type of data in the list. For example (list 1 2 3) is a `(listof number)`.
  - If it is a with different types of data, say `(listof any)`
  - If you know the result is specifically the empty list, which has no elements and therefore no element type, just say empty list.
  - If you know the result is a **list** but you don't know the type of data in it, just say "list" and we will give you partial credit.
- If the result is a **function**, give its argument and return types. That is, write the type(s) of its argument(s) followed by an arrow and the type of its result. If the procedure accepts any type of value for an argument, just say "any". For example:
  - The type of the **sin** function is: `number -> number`
  - The type of the **integer?** function is: `any -> boolean`
  - The type of the **<** function is: `number number -> boolean`

  - The type of the **square** function is: `number string color -> image`
- If you know the expression’s value is a function but don’t know its argument or return types, just say **function**, and we will give you partial credit.
- If executing it would produce an **exception**, say “Exception." You don't need to specify the type of exception.

Quiz Glossary (THIS LINK WORKS IN THE LOCKDOWN BROWSER) (https://bain-cs111.github.io/course-files/quizzes/q3_glossary_compact.pdf)

Question 2

5 / 5 pts

```
(λ (p)
  (filter p (list "a" "b" "c")))
```

Correct!

(string -> boolean) -> (listof string)

orrect Answers

(string -> boolean) -> (listof string)

Question 3

5 / 5 pts

```
(define-struct quiz (questions points))
(define q1 (make-quiz (list "part 1" "part 2" "part 3") 100))
(quiz-questions q1)
```

Correct!

(listof string)

orrect Answers

(listof string)

(listof strs)

list of strings

listof string

listof str

(listof str)

listof strs

(listof strings)

Question 4

5 / 5 pts

```
(define awkward
  (lambda (a-list)
    (foldl (λ (x)
              (* (+ x 3) 10))
           0
           a-list)))
(awkward (list 0 1 2 3))
```

Correct!

exception

orrect Answers

exception

error

Question 5

5 / 5 pts

```
(iterated-beside (lambda (n)
                   (circle (* n 10) "outline" "red"))
                 7)
```

Correct!

image

orrect Answers

shape

circle

img

image

picture

Question 6

5 / 5 pts

```
(define x 30)

(define function-mcfunction
  (lambda (num)
    (local [(define x "Science!")]
      (string-append "Computer " x))))

(function-mcfunction 50)
```

Correct!

string

orrect Answers

string

str

"Computer Science!"

Question 7

5 / 5 pts

```
(+ (1) 7)
```

Correct!

exception

orrect Answers

exception

Question 8

5 / 5 pts

```
(define high-ground
  (lambda (x y)
    (if (> y x)
        "it's over"
        404)))
(high-ground 10 11)
```

Correct!

string

orrect Answers

string

it's over

"it's over"

str

Question 9

5 / 5 pts

```
(define-struct quiz (questions points))
(define q1 (make-quiz (list "part 1" "part 2" "part 3") 100))
q1
```

Correct!

quiz

orrect Answers

struct

quiz

Question 10

5 / 5 pts

```
(list 1 2 3 4 5)
```

Correct!

(listof number)

orrect Answers

(listof number)

(listof num)

Question 11

5 / 5 pts

```
(lambda (x)
  (= (string-length x)
     5))
```

Correct!

string -> boolean

orrect Answers

string -> boolean

Each of the following questions shows some code being executed (with a line number on the left of each line) at the Racket prompt, along with the output or error it generated, and the intended output that the programmer wanted. Give the **correction** to the code to produce the desired result.

It is highly recommended that you copy and paste the code and then correct it as your answer (you can leave the line numbers in). If you correctly identify the error and fix it you will get full credit.

What we are looking for is that you understand what was wrong with the code and know how to fix it. If you are unsure of what the problem is, then you can provide an explanation of your thoughts. We will accept the line number(s) as a reference point, so that you can indicate where the error occurred, but more important will be your rewrite. We will grade these with the same leniency and partial credit that we would in a printed exam.

Quiz Glossary (THIS LINK WORKS IN THE LOCKDOWN BROWSER) (https://bain-cs111.github.io/course-files/quizzes/q3_glossary_compact.pdf)

Question 12

10 / 10 pts

Interaction

```
(map (* 2)
     (list 1 2 3 4 2 3 4 2))
```

Actual Output:

```
map: first argument must be a function that expects one argument, given 2
```

Desired Output:

```
(list 2 4 6 8 4 6 8 4) ; a list with each number doubled
```

Your answer:

```
(map (lambda (num) (* num 2))
     (list 1 2 3 4 2 3 4 2))
```

Stop. Lambda time!

Question 13

10 / 10 pts

Interaction

```
(define-struct animal (name type age food))

(define my-pets
  (list (make-animal "figaro"   "cat"   15 "milk")
        (make-animal "mr bunny" "bunny" 3  "grass")
        (make-animal "molly"    "dog"   9  "dog treats")))

(define is-young?
  (lambda (anim)
    (< (animal-age anim)
        9)))

(define has-all-young-animals
  (ormap is-young? my-pets))

has-all-young-animals
```

Actual Output:

```
#true
```

Desired Output:

```
#false ; because I have 1 or more old (not young) animals.
```

Your answer:

```
(define-struct animal (name type age food))

(define my-pets
  (list (make-animal "figaro"   "cat"   15 "milk")
        (make-animal "mr bunny" "bunny" 3  "grass")
        (make-animal "molly"    "dog"   9  "dog treats")))

(define is-young?
  (lambda (anim)
    (< (animal-age anim)
       9)))

(define has-all-young-animals
  (andmap is-young? my-pets))

has-all-young-animals
```

Should use andmap, not ormap.

ormap asks "do any of these things meet the condition." we want andmap!

Question 14

10 / 10 pts

Interaction

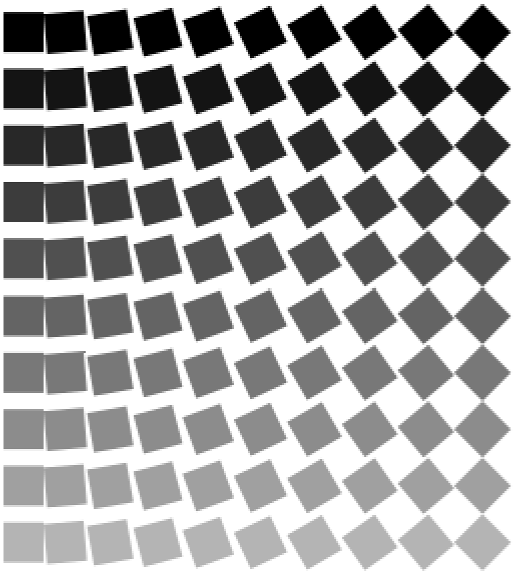
```
(define grid-of
  (lambda (func rows columns)
    (iterated-above (λ (rn)
                      (iterated-beside
                       (λ (cn)
                         (func rn cn))
                       columns))
                    rows)))

(grid-of (λ (row col)
           (local [(define grey (* row 20))]
             (rotate (* 5 col)
                     (square 20
                             "solid"
                             color grey
                             grey
                             grey))))
         10
         10)
```

Actual Output:

```
square: expects only 3 arguments, but found 6
```

Desired Output:



(a picture with rotated grey boxes should be rendered above)

Your answer:

```
(define grid-of
  (lambda (func rows columns)
    (iterated-above (λ (rn)
                      (iterated-beside
                       (λ (cn)
                         (func rn cn))
                       columns))
                    rows)))

(grid-of (λ (row col)
           (local [(define grey (* row 20))]
             (rotate (* 5 col)
                     (square 20
                             "solid"
                             (color grey
                                    grey
                                    grey)))))
         10
         10)
```

take a careful look at the color function!

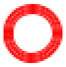
Question 15

10 / 10 pts

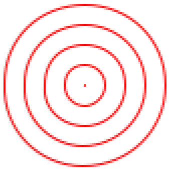
Interaction

```
(define target
  (lambda (x)
    (iterated-overlay (lambda (n)
                        (circle (+ n 10) "outline" "red"))
                      x)))
(target 5)
```

Actual Output:

 (overlayed circles of radius 10, 11, 12, 13, and 14)

Desired Output:

 (overlayed circles of radius 0, 10, 20, 30, 40)

Your answer:

```
(define target
  (lambda (x)
    (iterated-overlay (lambda (n)
                        (circle (* n 10) "outline" "red"))
                      x)))
(target 5)
```

Should we be adding 1 to each radius or should we be doing some other math here...

The following questions show a function definition. In the blank below, please provide one valid test (`check-expect`) of the procedure.

[Quiz Glossary (THIS LINK WORKS IN THE LOCKDOWN BROWSER)](https://bain-cs111.github.io/course-files/quizzes/q3_glossary_compact.pdf) (https://bain-cs111.github.io/course-files/quizzes/q3_glossary_compact.pdf)

Question 16

10 / 10 pts

```
; super-sum: (listof (listof number)) -> number
; takes in a list of lists of numbers and
; returns the sum of ALL the individual elements.
(define super-sum
  (lambda (the-list-of-lists)
    (apply + (apply append
                    the-list-of-lists))))
```

In the blank below, please provide one valid test (check-expect) of the procedure.

Your answer:

```
(check-expect (super-sum (list (list 1 2 3 4)
                               (list 2 4 6 8)
                               (list 1 1)
                               (list 2.5 2.5 2.5 2.5)))
              42)
```

Sample solution

```
(check-expect (super-sum (list (list 1 2) (list 3 4)))
              10)
```

Quiz score: **100** out of 100