0/20 Questions Answered

COMP1130 Mid-Semester Exam

STUDENT NAME

Search students by name or email...

Q1 Acknowledgment

0 Points

Australian National University

COMP1130 Mid-Semester Exam, Semester 1 2022

If you have moved to COMP1100, or intend to move to COMP1100, attempt all questions except those related to the lambda calculus.

You must acknowledge the following **integrity pledge** before proceeding. Please read carefully and check all the boxes.

- [ ] I am committed to being a person of integrity.
- [ ] I pledge, as a member of the ANU community, to abide by and uphold the standards of academic integrity outlined in the ANU statement on honesty and plagiarism, I am aware of the relevant legislation, and understand the consequences of breaching those rules.
- [ ] I will not communicate in any way with anyone else during this exam. This includes asking questions in any online forum.
- [ ] I acknowledge that this exam is protected by copyright and that copying or sharing any of its content will violate that copyright.

Read and check off the following instructions:

1. This examination is timed.

- [ ] Note the remaining time at the top right of this screen. Set an alarm for yourself if you need one.

2. Permitted materials. This is an open book exam. You might in particular find the course Website and the Prelude documentation useful.

- [ ] You may use any documentation you wish but **all work must be your own**.

Save Answer

Q2 Sets and Functions

2 Points

Given sets A and B, consider the right injection function

$$inj_{right} :: B \to A + B$$

Which of the following statements is **true**?

- For any element b of B, $inj_{right}(b) = b$.
- If b and c are different elements of B, then $inj_{right}(b)$ and $inj_{right}(c)$ are different elements of the codomain.
- There are valid inputs to inj_{right} that do not have a defined output.
- inj_{right} can be defined for some choices of sets A and B, but cannot be defined for some other choices.
- inj_{right} has domain $B \to A$.

Save Answer

Q3 Sets and Functions

2 Points

Given sets A, B, and C, consider the mathematical functions

$f :: A \to B$
$g :: B \to C$
$h :: B \to C \to A$

Which of the following compositions is a well-defined mathematical function?

- $f \,.\, g$
- $f \,.\, h$
- $g \,.\, h$
- $h \,.\, f$
- $h \,.\, g$

Save Answer

Q4 Haskell

2 Points

Which word can **not** be used to correctly describe Haskell?

- Declarative
- High-level
- Imperative
- Lazy
- Pure

Save Answer

Q5 Basic Types

2 Points

Which type can have any number of elements (up to the physical limits imposed by a computer and operating system)?

- `Bool`
- `Char`
- `Int`
- `Integer`
- `Word`

Save Answer

Q6 Types

2 Points

Which of the following is a valid Haskell type definition?

- `data MyType = foo Double`
- `data MyType = Foo Double`
- `MyType = foo Double`
- `MyType = Foo Double`
- `type MyType = foo Double`
- `type MyType = Foo Double`

Save Answer

Q7 Algebraic Datatypes

2 Points

Suppose `TypeA` is a user-defined finite type containing **3** elements, and `TypeB` is a user-defined finite type containing **4** elements. How many elements are in the type defined below?

```
data TypeC = Bar TypeA | Baz TypeB TypeB | Qux
```

- ○ 19
- ○ 20
- ○ 28
- ○ 29
- ○ 35

Save Answer

Q8 Guarded Expressions

2 Points

NOTE: An error was detected in this question. All students who sat the exam received 2 marks for this question.

Consider the four possible definitions below of a function

```
calc :: Int -> Int -> Int
```

A

```
calc x y
  | x < y      = 4 * x - y
  | x > y      = 3 * y
  | otherwise = x + y
```

B

```
calc x y
  | x < y      = 4 * x - y
```

```
  | x > 2 * y = x + y
  | otherwise = 3 * y
```

C

```
calc x y
  | x > 2 * y = x + y
  | x < y     = 4 * x - y
  | otherwise = 3 * y
```

D

```
calc x y
  | x > 2 * y = x + y
  | x > y     = 3 * y
  | otherwise = 4 * x - y
```

Three or four of them will produce the same output as each other, given the same input. Which one (if any) will behave **differently** on some inputs?

- ○ **A**
- ○ **B**
- ○ **C**
- ○ **D**
- ○ They all produce the same output as each other, given the same input.

Save Answer

Q9 Case Expressions

2 Points

Consider the type definition

```
data Element = Air | Earth | Fire | Water
```

and consider the four possible definitions below of a function

```
transform :: Element -> Element
```

A

```
transform ele = case ele of
  Air   -> Earth
  Earth -> Fire
  _     -> ele
```

B

```
transform ele = case ele of
  Air   -> Earth
  Earth -> Fire
  e     -> e
```

C

```
transform ele = case ele of
  Air   -> Earth
  Water -> ele
  _     -> Fire
```

D

```
transform ele = case ele of
  Air   -> Earth
  Water -> Water
  _     -> Fire
```

Three or four of them will produce the same output as each other, given the same input. Which one (if any) will behave **differently** on some inputs?

- ○ **A**
- ○ **B**
- ○ **C**
- ○ **D**
- ○ They all produce the same output as each other, given the same input.

Save Answer

Q10 Recursion

2 Points

Consider the following five functions, all with type `Int -> String`.

```
rec1 n
   | n >= 0     = "."
   | otherwise = '1' : rec1 (abs n)
```

```
rec2 n
   | n >= 0     = "."
   | otherwise = '2' : rec2 (n+1)
```

```
rec3 n
   | n == 100     = "."
   | n > 100      = '0' : rec3 (n-1)
   | otherwise    = '3' : rec3 (n+2)
```

```
rec4 n
   | n >= 100   = "."
   | otherwise = '4' : rec4 (n+2)
```

```
rec5 n
   | n >= 100   = "."
   | otherwise = '5' : rec5 (n^2)
```

Which one of them will **loop forever** (or until the computer runs out of memory) on some input(s)?

- O `rec1`
- O `rec2`
- O `rec3`
- O `rec4`
- O `rec5`

Save Answer

Q11 Recursion and Lists

2 Points

Suppose we have a function

```
recFunc :: [Int] -> Int
```

which is defined by recursion on its input list. Which statement about this function will be **true**?

- If `recFunc` has a base case, then it will certainly terminate on all inputs
- The code for the definition of `recFunc` will contain a call to the function `recFunc`
- `recFunc` will use an accumulator
- `recFunc` will be split into exactly two cases: one where the input is the empty list, and one where the input is non-empty
- Running `recFunc` might cause some function to run multiple times, with a different input each time.

Save Answer

Q12 Programming Questions

30 Points

There are **four** Haskell files that you need to complete and submit. Note that some files include more than one function to complete. Each function is worth **5** marks.

Please download the template Haskell files here.

You can find all programming questions on your dashboard.

Please submit by uploading **each** Haskell file to **each** question.

Q12.1 SameSign.hs

5 Points

Submit `SameSign.hs` here

Save Answer

Q12.2 Places.hs

10 Points

Submit `Places.hs` here

Save Answer

Q12.3 OddsOverEvens.hs

5 Points

Submit `OddsOverEvens.hs` here

Save Answer

Q12.4 StringTransformers.hs

10 Points

Submit `StringTransformers.hs` here

Save Answer

Q13 Lambda Calculus

2 Points

Consider the term

$$(x((\lambda y.\, y\, x)z))y$$

According to the conventions for parentheses in the lambda calculus, which of the following terms has the same meaning?

- $x((\lambda y.\, y\, x)z)y$
- $(x(\lambda y.\, y\, x)z)y$
- $(x(\lambda y.\, y\, x\, z))y$
- $x(\lambda y.\, y\, x)z\, y$
- $x(\lambda y.\, y\, x\, z)y$
- $(x\, \lambda y.\, y\, x\, z)y$

Save Answer

Q14 Lambda Calculus

2 Points

Consider the term

$$(\lambda v.\lambda w.\, v\, x(\lambda y.\, v\, y))(\lambda z.\, z\, w)$$

Which of the following terms is alpha-equivalent?

- $(\lambda v.\lambda u.\, v\, x(\lambda y.\, v\, y))(\lambda z.\, z\, u)$
- $(\lambda v.\lambda w.\, v\, u(\lambda y.\, v\, y))(\lambda z.\, z\, w)$
- $(\lambda v.\lambda w.\, v\, x(\lambda z.\, v\, z))(\lambda z.\, z\, w)$
- $(\lambda w.\lambda w.\, w\, x(\lambda y.\, w\, y))(\lambda z.\, z\, w)$
- $(\lambda z.\lambda w.\, z\, x(\lambda y.\, v\, y))(\lambda z.\, z\, w)$

Save Answer

Q15 Lambda Calculus

2 Points

Consider the term

$$(\lambda x.\, x(\lambda y.\, y\, x))(\lambda z.\, y\, z)$$

Which of the following terms is (alpha-equivalent to) the result of applying beta-reduction as many times as possible?

- $y(\lambda w.\, w\, y)$
- $y(\lambda w.\, w(\lambda z.\, y\, z))$
- $y\, y$
- $y(\lambda y.\, y\, y)$
- $y(\lambda y.\, y(\lambda z.\, y\, z))$
- $y(\lambda z.\, y\, z)$

Save Answer

Q16 Lambda Calculus

2 Points

Suppose that, **instead** of the encoding shown in lectures, we encoded pairs $\langle A, B \rangle$ as

$$\lambda x.\lambda y.\, x\, y\, A\, B$$

Which of the following terms is **not** a valid encoding of `fst`?

- $\lambda x.\, x(\lambda x.\, x)(\lambda x.\lambda y.\, x)$
- $\lambda x.\, x(\lambda x.\lambda y.\, y)(\lambda x.\lambda y.\, x)$
- $\lambda x.\, x(\lambda x.\lambda y.\lambda z.\, x\, z\, y)(\lambda x.\lambda y.\, y)$
- $\lambda x.\, x(\lambda x.\lambda y.\lambda z.\, y)(\lambda x.\lambda y.\, y)$

Save Answer

Q17 Lambda Calculus

2 Points

Suppose we had correct encodings of

- Booleans (**True**, **False**, and **if ... then ... else**)
- lists, including
  - a function **null** that returns **True** on an empty list and **False** on a non-empty list
  - a function **tail** that returns the tail of a non-empty list
- natural numbers, including
  - the numbers 0 and 1
  - plus $(+)$
- Turing's fixed point combinator Θ

Which of the following is a correct definition of a length function?

- $\Theta(\lambda x.\lambda y.\ \text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$
- $\Theta(\lambda y.\lambda x.\ \text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$
- $\lambda x.\ \Theta(\lambda y.\ \text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$
- $\lambda x.\ \lambda y.\ \Theta(\text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$
- $\lambda y.\ \Theta(\lambda x.\ \text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$
- $\lambda y.\ \lambda x.\ \Theta(\text{if}\ (\text{null}\ x)\ \text{then}\ 0\ \text{else}\ (1 + (y\ (\text{tail}\ x))))$

Save Answer

Save All Answers

Submit & View Submission ›