CS 571
Quiz 4 Solution

Nov 9 (Actually held on Nov 14) **Closed book**
15 points **Closed notes**

Important Reminder: As per the course Academic Honesty Statement, cheating of any kind will minimally result in receiving an F letter grade for the entire course.

Please ensure that you have filled-in BOTH your name and B-number in the bubbles on the provided grid-sheet.

For each of the following questions, select a **single** alternative on the grid-sheet.

There are 7 questions with 2-points per question; there is 1-point for submitting the quiz.

1. What will be the result of evaluating the following Haskell expression?

   ```
   length [[1, 2], [3, 4], [5]]
   ```

   (a) An error will occur.
   (b) `2`.
   (c) `3`.
   (d) `4`.
   (e) `5`.

   Answer: (c).

   `length` returns the number of elements in a list. In this case, the argument list has 3 elements (which themselves are lists). Hence the `length` will evaluate as 3.

2. What will be the result of evaluating the following Haskell expression?

```
length [(x, y) | x <- [1..10], y <- "abcde"]
```

   (a) An error will occur.

   (b) 10.

   (c) 15.

   (d) 20.

   (e) 50.

   Answer: (e).

   The list comprehension builds a list of pairs `(x, y)` with `x` ∈ `[1, 2, 3, ..., 10]` and `y` ∈ `['a', 'b', 'c', 'd', 'e']`. So there are 10 possibilities for `x` and 5 possibilities for `y`; hence there are a total of $10 \times 5 = 50$ possible pairs. Hence the length of the list of pairs will be 50.

3. What will be the result of evaluating the following Haskell expression?

```
foldl (-) 1 [1, 2, 3]
```

   (a) An error will occur.

   (b) `1.`

   (c) `-1`

   (d) `5.`

   (e) `-5.`

   Answer: (e).

   The expression will apply `-` from the left with initial value `1`. Hence the expression is equivalent to `((1-1)-2)-3` which is -5.

4. What will be the value of evaluating the following Haskell expression?

```
foldr (-) 1 [1, 2, 3]
```

   (a) An error will occur.

(b) `1.`

(c) `-1`

(d) `5.`

(e) `-5.`

Answer: (b).

The expression will apply `-` from the right with initial value `1`. Hence the expression is equivalent to `(1-(2-(3-1))` which is 1.

5. Which of the following is not a legal Haskell expression?

   (a) `[1, 2, 3]`.

   (b) `['a', 'b'] ++ "c"`.

   (c) `[1, 2, [1]]`.

   (d) `[['a', 'b'], "cd"]`.

   (e) `[[1], [2]]`.

   Answer: (c).

   (c) cannot be typed since lists must be of homogeneous type, but the elements of (c) are `NUM`'s and a list of `NUM`. (a) is a simple list of `Num`, (b) is equivalent to `"abc"` with type list of `Char`, (d) is equivalent to `["ab", "cd"]` with type list of list of `Char`, and (e) is a list of list of `Num`.

6. What will be the result of evaluating the following Haskell expression?

```
foldr (++) "x" ["abc", "de", "f"]
```

   (a) An error will occur.

   (b) `"abcdef"`.

   (c) `"fedcba"`.

   (d) `"xabcdef"`.

   (e) `"abcdefx"`.

Answer: (e).

The `foldr` folds the `++` append operation over the list and is equivalent to `"abc" ++ "de" ++ "f" ++ "x"` resulting in `"abcdefx"`.

7. Given sets $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$, which of the following is **not** a function from A to B.

   (a) $\{\}$.

   (b) $\{(1, a), (2, a), (3, a), (4, a)\}$.

   (c) $\{(1, a), (2, b), (3, c), (4, d)\}$.

   (d) $\{(1, b), (2, b), (1, d), (4, d)\}$.

   (e) $\{(1, a)\}$.

   Answer: (d).

   An essential property for a function is that it must map a single element in the domain to a unique element in the range; however, (d) maps `1` to both `b` and `d`. The other alternatives do not have similar violations of this function property. Note that (a) is a function which is undefined over all elements of its domain.