

**The University of Melbourne**  
**Department of Computing and Information Systems**  
**COMP30020/COMP90048 Declarative Programming**  
**Sample mid-semester test**  
**Semester 2, 2016**

Write your enrolment number here:

**Reading Time:** Five (5) minutes.

**Writing Time:** Forty-five (45) minutes.

**This paper has 6 pages including this cover page.**

**Authorized Materials:** No materials are authorized. Calculators are *not* permitted.

**Instructions to Invigilators:** Students will write all of their answers on this exam paper. *Students may not remove any part of the paper from the examination room.*

**Instructions to Students:**

- Start by filling in the box above with your student number. Unidentified test papers yield no marks for anyone.
- This test counts for 10% of your final grade. *All questions should be answered* in the boxes provided on the paper. Only material written inside the boxes will be marked.
- Answers should be kept short and should display good programming style.
- The reverse side of any page may be used to make rough notes, or prepare draft answers.
- Unreadable answers will be deemed wrong.
- Use a blue or black pen or pencil.
- You do not need to write comments in your code, but you may include comments if you feel that they would assist the examiner in understanding your code.

**Library:** This paper may *not* be held by the Baillieu Library.

Question	1	2	3	4	Total
Marks					

**Question 1** (2 marks)

For each of the following Haskell expressions, write down its **type** (which may be a function type or may include type class constraints) or say that it represents a type error.

- A    `[]`
- B    `(++)`
- C    `map`
- D    `show`
- E    `(. length)`

A: <code>[]</code>
B: <code>[a] -&gt; [b] -&gt; [c]</code> <code>"[a] -&gt; [a] -&gt; [a]"</code>
C: <code>(a -&gt; b) -&gt; [a] -&gt; [b]</code>
D: <code>(Readable a) =&gt; a -&gt; "a"</code> <code>"Show a =&gt; a -&gt; String"</code>
E: <code>[a] -&gt; Int</code> <code>"(Int -&gt; a) -&gt; [b] -&gt; a"</code>

**Question 2** (2 marks)

Briefly explain what this Haskell function does. Give a type declaration for `q2`.

```
q2 _ [] = 0
q2 a (b:bs) = c + q2 a bs
  where c = if a == b then 1 else 0
```

<code>q2 :: Eq a =&gt; a -&gt; [a] -&gt; [Int]</code>	<code>"q2 :: (Eq a, Num b) =&gt; a -&gt; [a] -&gt; b"</code>

**Question 3 (3 marks)**

Write a function `merge` that takes two sorted lists as input and returns a sorted list of all the elements of both lists by interleaving them as necessary to put them in order. Include a type declaration for your function.

[illegible]

**Question 4 (3 marks)**

Given the following binary search tree (BST) data type

```
data Tree a = Empty | Node (Tree a) a (Tree a)
```

write a Haskell function `insert` to insert an element into a tree. Recall that in a BST, every element appearing in the first (left) subtree of a node must be smaller than the element of that node, and every element in the second (right) subtree must be larger. Note that this tree type does not have values, just keys. Include a type declaration for your `insert` function. You may use any function in the Haskell prelude, but not functions defined in libraries.

[illegible]

## Overflow answers

If you do need to use this page, indicate **CLEARLY** in your previous answer that you have continued onto this page. Without such an indication, it is possible that this part of your answer will be overlooked.

[illegible]