PROGRAMMING AS PROBLEM SOLVING
COMP1100

Important notice: Code templates for programming questions in this exam can be found in the accompanying VSCodium project. You can place this Web page for the exam next to the VSCodium project window, so that you can conveniently switch between them while working on your answers. You will also want to open the Terminal tool in VSCodium, in which you can test your solution code with ghci. You can also make use of doctest for testing.

All of your answers will be auto-graded, so for coding problems you will be marked according to how many of our tests you pass. Incorrect answers that pass tests will be penalised accordingly. The auto-grader will only mark code that you upload to the exam by dragging it into the appropriate box from a file window.

Total marks: 50
Reading period: 10 minutes duration (no use of keyboard or selection of answers)
Writing period: 60 minutes duration
Permitted materials: One A4 page with notes on both sides, Unannotated paper-based dictionary.

Questions are not of equal value.
All questions must be completed on this web form.

Your work is automatically saved and recorded as you type.

This is a closed examination. You may not copy this exam.

Question 1 [20 Marks] *Multichoice Questions*

Each question is intended to have only one correct answer. Each question is worth 2.5 marks. An incorrect or missing answer is worth 0 marks, without further mark penalty.

1 i) [2.5 Marks] *Sets and Functions*

What is the sum of the sets `{ 1, 2, 3 }` and `{ 2, 3, 4 }` ?

- `{ 1, 2, 3, 4 }`
- `({ 1, 2, 3 }, Left) + ({ 2, 3, 4 }, Right)`
- `{ (1, Left), (2, Left), (3, Left), (2, Right), (3, Right), (4, Right) }`
- `{ (1, Left), (2, Left), (3, Right), (4, Right) }`

Clear

1 ii) [2.5 Marks] *Sets and Functions*

Let `A` , `B` and `C` be sets. Let `a` be an element of `A` and `b` an element of `B` . Finally, let `h :: A -> B` and `f :: A -> B -> C` be mathematical functions. Which of the following is an element of `C` ?

- `f(h)`
- `f(h(a))`
- `(f(a))(h(a))`
- `(f(b))(h(a))`

Clear

1 iii) [2.5 Marks] *Programming*

What does it mean when we say that Haskell is 'lazy'?

- Haskell can only be used in the afternoon, because in the mornings it is still asleep
- Every expression in Haskell has a type
- Expressions are not evaluated until their results are needed
- Everything in Haskell can be viewed as a function

Clear

1 iv) [2.5 Marks] *Algebraic datatypes*

Which of the following is a valid definition of a datatype in Haskell?

- `type IntPlusBool = First Int | Second Bool`
- `type IntPlusBool = Int | Bool`
- `data IntPlusBool = Left Int | Right Bool`
- `data IntPlusBool = Int | Bool`

Clear

1 v) [2.5 Marks] *Types and lists*

Which of the following is a correct type assignment in Haskell?

- `[Int] :: [a]`
- `[] :: [Int]`
- `2 :: [Int]`
- `(2 : 3 : 4) :: [Int]`

Clear

1 vi) [2.5 Marks] *Lists*

Which of the following Haskell expressions is **not** the same as `[1, 2, 3]` ?

- `[] : 1 : 2 : 3`
- `1 : [2, 3]`
- `[1] ++ [2, 3]`
- `[1..3]`

Clear

1 vii) [2.5 Marks] *Cases and guards*

Which of the following statements is true?

- Guards and cases are interchangeable
- Cases can do anything that guards can do
- Cases should always end with `"otherwise"`
- Cases are evaluated from bottom to top, while guards are evaluated from top to bottom

Clear

1 viii) [2.5 Marks] *Resolving a warning*

Consider the function `maybeHead` which is defined as follows:

```
maybeHead :: [Char] -> Maybe Char
maybeHead list = case list of
  [] -> Nothing
  x:xs -> Just x
```

When compiling this function it gives the following warning:

```
maybeHead.hs:6:5: warning: [-Wunused-matches] Defined but not used: 'xs'
  |
6 |   x:xs -> Just x
  |   ^^
```

Which of the following modifications to our code prevents this warning?

- Change `"x:xs"` to `"x:(Just x)"`
- Change `"xs"` to `"_"`
- Change `"x"` to `"_"`
- Change `"xs"` to `"ys"`

Clear

Question 2 [5 Marks] `AnySum.hs`

The following instructions apply to all programming questions below and should be read carefully.

There are **six** Haskell files that you need to complete and submit. Each file contains exactly one function to complete. You will find the template Haskell files in a folder on your desktop, and in VSCodium.

You may define helper functions if you wish. You may use any Haskell function available in the Prelude or basic libraries. The doctests in the file are intended to help you, but are not intended to be exhaustive, and do not replace your need to test your own code. The doctests are not identical to the tests that will be used to mark you.

Please submit by dragging and dropping **each** Haskell file into the white box below **each** question. Do not rename the files before submission. You should be able to see automatic test results for your files after submission. You may submit to the same question multiple times; if you do so, your previous submission will be overwritten.

These questions are auto-graded: you will be graded according to how many tests you pass. Each question is worth five marks, one mark for each passed test. After dragging your code into the box, you will see five green ticks, yellow exclamation marks, or red crosses above the box. A green tick indicates a succesful test (worth one mark). A yellow exclamation mark indicates a partially succesful test (which is worth zero marks), and a red cross indicates a failed test (worth zero marks).

The auto-grader will only mark what you upload to the exam. Therefore it is **essential** that you upload your code into this exam by dragging your file into the appropriate box. Note that **code that cannot run** will receive zero marks, even if part of the code is correct. In particular, if you import packages that are not basic libraries, your code will receive zero marks.

Complete the program `AnySum.hs` in your VSCodium project.

Drag and drop AnySum.hs below

Question 3 [5 Marks] `Temperature.hs`

Complete the program `Temperature.hs` in your VSCodium project.

Drag and drop Temperature.hs below

Question 4 [5 Marks] `Eyes.hs`

Complete the program `Eyes.hs` in your VSCodium project.

Drag and drop Eyes.hs below

Question 5 [5 Marks] `MaybeProd.hs`

Complete the program `MaybeProd.hs` in your VSCodium project.

Drag and drop MaybeProd.hs below

Question 6 [5 Marks] `Sentence.hs`

Complete the program `Sentence.hs` in your VSCodium project.

Drag and drop Sentence.hs below

Question 7 [5 Marks] `Skiponacci.hs`

Complete the program `Skiponacci.hs` in your VSCodium project.

Drag and drop Skiponacci.hs below