

**THE AUSTRALIAN NATIONAL UNIVERSITY**

*Second Semester 2006*

**COMP1100**  
**(Introduction Programming and Algorithms)**

*Writing Period: 1 hour duration*

*Study Period: 15 minutes duration*

*Permitted Materials: Any, except ANU library books and electronic devices*

*Answer ALL questions*

*Total marks: 50*

*The questions are followed by labelled blank spaces into which your answers are to be written.*

*Additional answer panels are provided (at the end of the paper) should you wish to use more space for an answer than is provided in the associated labelled panels. If you use an additional panel, be sure to indicate clearly the question and part to which it is linked.*

Name (family name first):

Student Number:

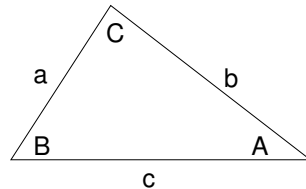
*The following are for use by the examiners.*

Q1 Mark	Q2 Mark	Q3 Mark	Q4 Mark	Q5 Mark	Total Mark

## QUESTION 1 [10 marks]

The area of a triangle with side lengths  $a$ ,  $b$ ,  $c$  and internal angles  $A$ ,  $B$ ,  $C$  may be calculated with the following formula:

$$\frac{c^2 \cdot \sin A \cdot \sin B}{2 \cdot \sin(A + B)}$$



- (a) Define a Haskell function to calculate the area of a triangle by using this formula. Include an appropriate type signature.

QUESTION 1(a)	[4 marks]
---------------	-----------

- (b) A triangle may be represented by its three internal angles.

```
type Angle = Float
type Triangle = (Angle, Angle, Angle)
```

A triangle is *obtuse* if any of its internal angles are greater than 90 degrees. Define a Haskell function:

```
isObtuse :: Triangle -> Bool
```

which checks whether a triangle is obtuse.

QUESTION 1(b)	[6 marks]
---------------	-----------

## QUESTION 2 [12 marks]

Are each of the following valid Haskell expressions? If so, what are their values? If not, why are they invalid?

(a) `[1] : [2,3,4]`

QUESTION 2(a)	[3 marks]
---------------	-----------

(b) `([["perch"], ["gourami", "cichlid"]] !! 1) !! 1`

QUESTION 2(b)	[3 marks]
---------------	-----------

(c) `map fst [(1, [1..10]), (2, ['a'..'z'])]`

QUESTION 2(c)	[3 marks]
---------------	-----------

(d) `length [(+), (-)] ++ [(*)]`

QUESTION 2(d)	[3 marks]
---------------	-----------

### QUESTION 3 [12 marks]

The following functions have been disguised by giving them nonsensical names . Describe the behaviour of each function and suggest more meaningful names.

(a) 

```
lobster :: [a] -> Int
lobster xx      = prawn 0 xx
  where prawn n []      = n
        prawn n (x:xs) = prawn (n+1) xs
```

QUESTION 3(a)

[4 marks]

(b) 

```
echidna :: Ord a => (a, a) -> [a] -> [a]
echidna _      []      = []
echidna (a, b) (x:xs)
  | x < a || x > b = echidna (a, b) xs
  | otherwise     = x : echidna (a, b) xs
```

QUESTION 3(b)

[4 marks]

```
(c) kangaroo :: [Int] -> [[Int]]
    kangaroo [] = []
    kangaroo (i:is) = wallaby i i : kangaroo is
    where wallaby 0 x = []
          wallaby n x = x : wallaby (n-1) x
```

QUESTION 3(c)

[4 marks]

#### QUESTION 4 [6 marks]

The following function definition is missing its type signature.

```
mapT2 f g [] = []
mapT2 f g ((x, y) : xs) = (f x, g y) : mapT2 f g xs
```

What is the type of mapT2?

QUESTION 4

[6 marks]

## QUESTION 5 [10 marks]

- (a) The binary search trees discussed in lectures were represented by the following data type:

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

Recall the invariant for binary search trees: for any given Node with key  $k$ , all the keys in the left hand sub-tree must be less than  $k$ , and the keys in the right hand sub-tree must be greater than  $k$ .

Write a Haskell function that takes a search tree and finds the key with the lowest value. The function should have the following type:

```
treeMinimum :: Tree a -> a
```

QUESTION 5(a)

[7 marks]

(b) The lookup function for trees may be defined as follows:

```
treeLookup :: Ord a => a -> Tree a -> Maybe a
treeLookup x Null      = Nothing
treeLookup x (Node x' left right)
  | x < x'      = treeLookup x left
  | x > x'      = treeLookup x right
  | otherwise   = Just x'
```

What is the meaning of the `Ord a` component of this function's type, and how does this relate to the definition of `treeLookup`?

QUESTION 5(b)

[3 marks]

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

---

---