Candidate Number

G6021

THE UNIVERSITY OF SUSSEX

BSc and MComp FINAL YEAR EXAMINATION
January 2020 (A1)

Comparative Programming

Assessment Period: January 2020 (A1)

Assignment Project Exam Help

DO NOT TURN OVER UNTIL INSTRUCTED TO BY THE LEAD INVIGILATOR

https://powcoder.com

Candidates should answer TWO questions out of THREE. If all three questions are attempted only the first two answers will be marked.

Add WeChat powcoder

The time allowed is TWO hours.

Each question is worth 50 marks.

At the end of the examination the question paper and any answer books/answer sheets, used or unused, will be collected from you before you leave the examination room.

1. (a) Define *multiple inheritance* and state any problems associated with this concept. Briefly show how these problems can be circumvented, using examples to support your argument. [10 marks]

   (b)
   - Describe briefly the *call-by-name* and *call-by-value* strategies of evaluation of functions, highlighting the differences between them. [5 marks]
   - Consider the two functions `zero` and `loop` below:

     ```
     zero :: Integer -> Integer
     zero x = 0

     loop :: Integer -> Integer
     loop x = loop (x + 1)
     ```

     What would happen if the expression `zero (loop 10)` were to be evaluated using each of the *call-by-name* and *call-by-value* strategies? Justify your answer. [5 marks]

   (c) Consider the following definition of the function `iterate_until`:

Assignment Project Exam Help

   ```
   iterate_until p f x = if (p (f x)) then (f x)
                         else iterate_until p f (f x)
   ```

https://powcoder.com

   Give a polymorphic type for this function. [10 marks]

   (d) i. Define a non-terminating term in PCF, and include types in your answer. Can you define such a term in the pure lambda calculus and in the simply typed lambda calculus? [10 marks]

Add WeChat powcoder

   ii. Show how to define the function `iterate_until` (given in Part 1(c)) as the fix point of a functional in PCF. [10 marks]

2. (a) Consider the λ-term $t = vw(\lambda xy.vx)$.

   i. Write t in full, inserting all the missing parentheses and λ's.

   ii. List the *free variables* and *bound variables* of t.

   [10 marks]

   (b) Give the β-reduction graph of the λ-term $(\lambda xy.x)(II)I$, where $I = \lambda x.x$. Underline all redexes in the graph. [15 marks]

   (c) Define and explain the term *disagreement set* as used in the unification algorithm as part of the algorithm for type reconstruction. Include in your answer an example to illustrate the concept. [10 marks]

   (d) Build a type derivation to find the type of $\lambda x.(\lambda y.y)x$. [15 marks]

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

3. This question is about comparing paradigms for adding an element at the end of a list.

   (a) Consider the following logic program defining the insertion of an element at the end of a list.

   ```
   insert(X,[],[X]).
   insert(X,[S1|S],[S1|S2]) :- insert(X,S,S2).
   ```

   Draw the SLD-resolution tree for the query:

   `:- insert(1,[2],Y).`

   Indicate the answers that Prolog finds for the query above. [15 marks]

   (b) Write a polymorphic Haskell function `insert` to add an element at the end of a list. Include in your answer:

   - The type of your function.
   - A reduction graph of `insert 2 [1]`.

   [15 marks]

Assignment Project Exam Help

   (c) Compare the different paradigms studied in this module for adding an element to the end of a list. Your answer should incorporate features of different paradigms such as Prolog difference lists, accumulating parameters, objects, and any other appropriate concept to justify your answer. [20 marks]

https://powcoder.com

Add WeChat powcoder