UNIVERSITY OF SWAZILAND

Faculty of Science

Department of Computer Science

FINAL EXAMINATION 2007

Title of paper: PROGRAMMING LANGUAGES

Course number: CS343

Time allowed: Three (3) hours

Instructions: Answer any five (5) of the six (6) questions.

This examination paper should not be opened until permission has been granted by the invigilator.

Question 1

a) *Precedence*, *associativity*, *arity* and *fixity* are properties of operators in programming languages. Explain the meaning of each term with the aid of examples taken from actual programming languages.

[12]

b) Give a brief explanation of *denotational semantics.*

[3]

c) Explain the main advantages of statically typed languages over dynamically typed languages.

[5]

Question 2

a) Give an overview of *by-copy* and *by-reference* parameter passing mechanisms.

[8]

b) Explain and clearly distinguish between the following kinds of *scope* of variables:

i. Global
ii. Lexical
iii. Dynamic

[8]

c) Explain the terms: *inclusion polymorphism* and *interface-implementation separation.*

[4]

Question 3

a) Consider the statement: "The semantic gap for imperative languages is narrower than for declarative languages. Therefore, programs in imperative languages run faster but programs in declarative languages are easier to understand."

Explain the underlined terms and discuss the stated conclusion.

[10]

b) Briefly describe the main properties of *objects* in the object oriented programming paradigm.

[3]

c) Explain in detail how, in C++, dynamic dispatch of member functions is implemented with the help of *vtables*.

[7]

Question 4

a) Describe in detail the structure of Lambda calculus expressions, as well as the method by which the expressions are evaluated (reduced to normal form).

[14]

b) In Lambda Calculus, functions are limited to 1 parameter and are not named. Write 2 Lambda Calculus expressions showing how these limitations can be overcome.

[6]

Question 5

a) Write a Haskell function named `countMatches` that takes 2 parameters: (i) a function of one parameter that returns a Boolean result; and (ii) a list. `countMatches` must return the number of elements of the given list for which the given function parameter returns `True`.

For example, the value of the following expression must be `3` since 3 of the listed numbers are odd:

```
countMatches odd [1, 6, 9, 3]
```

[7]

b) The value of *sin x* (where *x* is expressed in radians) can be found by adding the terms of the following series:

$$\frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

Write a Haskell function named `sine` that takes 2 parameters: (i) a real number `x`; and (ii) a positive integer `n`. This function must use the first `n` terms of the series to return an approximate value of *sin x*.

[13]

Question 6

a) The following Prolog query will request the user to input a list:

```
read(L), [A, B, C] = L, number(A), number(B),
A < B, list(C).
```

In order to make this query succeed, what conditions must be fulfilled by the input list?

[4]

b) Assume that a large number of facts have been entered into Prolog concerning parent-child relationships, of the form `parent(tom, sue)` for example, to represent that `tom` is a parent of `sue`. Furthermore, assume that facts about the gender of each individual have also been entered, of the form `male(tom)` and `female(sue)`.

i. Write a query to determine whether `ann` is the mother of both `jim` and `joe`.

[3]

ii. Define a rule named `grandparent(X, Y)` that succeeds when `X` is a grandparent of `Y`.

[3]

iii. Write a query to find out how many males are known by the system.

[3]

iv. Write a query to determine the total number of grandchildren of `rob`.

[3]

c) Write a recursive Prolog predicate `factorial(X, Y)` that binds `Y` to the factorial of `X`. Assume that `X` will always be given as a non-negative integer.

[4]