UNIVERSITY OF SWAZILAND

Faculty of Science

Department of Computer Science

MAIN EXAMINATION 2012

Title of paper: **PROGRAMMING LANGUAGES**

Course number: **CS343**

Time allowed: **3 hours**

Instructions to candidates:

This question paper consists of **FIVE (5)** Questions. Answer any **FOUR (4)** questions. Marks are indicated in the square brackets.

All questions carry equal marks.

THIS EXAMINATION PAPER SHOULD NOT BE OPENED UNTIL PERMISSION HAS BEEN GRANTED BY THE INVIGILATOR

QUESTION 1

a) State any 5 reasons why we study concepts of programming languages. [5]

b) i) State 2 uses of parameters. [2]

ii) With the aid of appropriate sample/example programs explain the difference between actual parameters and formal parameters. [6]

iii) What is the difference between a function and a procedure? [4]

c) Over the last four decades what factor do you think has had the most influence on the design of programming languages? [2]

d) Language implementation can rely on compilers or interpreters. What are the differences between compilation and interpretation? [6]

QUESTION 2

a) Briefly describe the following terms, as they are understood by a **PROLOG** programmer:

i. Fact;
ii. Rule;
iii. Query.
iv. Unification [8]

b) **PROLOG** programs are often described as being declarative in nature. Explain what we mean when we say program statements are declarative. [3]

c) Briefly describe the characteristic features of the following:

ii) Functional programming languages
iii) Logic programming languages [3marks each]

State 3 applications of each of these types of programming languages. [3]

d) University of Swaziland has decided to automate the process it uses for checking students' degree programmes. The University needs to be able to check that students have the correct pre-requisites for any paper that they wish to take. The rules for pre-requisites for papers in the BSc in Physical Education, Department of Surfing can be expressed in PROLOG as:

```
/*

prereq(Course1, Course2) succeeds if Course2 is a pre-requisite for Course1

*/
prereq(surf113, surf112).

prereq(surf212, surf113).
prereq(surf213, surf113).
prereq(surf215, surf113).
prereq(surf216, surf113).

prereq(surf311, surf215).
prereq(surf312, surf215).
prereq(surf312, surf216).
prereq(surf313, surf212).
prereq(surf313, surf216).
prereq(surf314, surf212).
prereq(surf314, surf216)
prereq(surf315, surf213).
prereq(surf315, surf215).
prereq(surf316, surf213).
prereq(surf316, surf215).
prereq(surf389, surf311).
```

Unfortunately the University has employed a Surfing graduate, rather than a Computer Science graduate, and he has written the following PROLOG code:

```
/*

requires(Course1, Course2) succeeds if Course2 is a requirement for Course1.
Course2 is a requirement for Course1 if it is a pre-requisite for a course
which is a requirement for Course1 or if is a pre-requisite for Course1.

*/

requires(Course1, Course2) :-
      requires(Course1, Other),
      prereq(Other, Course2).
requires(Course1, Course2) :-
      prereq(Course1, Course2).
```

Explain how PROLOG will attempt to answer the following query, and what the result will be:

?- requires(surf389, surf112). [5]

QUESTION 3

a) Briefly describe the **GOTO** statement. Also explain why it is discouraged by structured programming languages. [4]

b) i) Briefly explain the purpose of parse trees. [3]

ii) BNF is said to be a metalanguage. What is a metalanguage? [2]

iii) Given the following **BNF** grammar:

<exp> **::=** ***<term>*** **+** ***<exp>*** **|** ***<term>*** **-** ***<exp>*** **|** ***<term>***
<term> **::=** ***<factor>*** * ***<term>*** **|** ***<factor>*** **/** ***<term>*** **|** ***<factor>***
<factor> **::= (** ***<exp>*** **) | a | b | c | d | 1 | 2 | 3 | 4**

Construct the parse tree for the expression **a – b*(c+d)**. [6]

c) Consider a program with 3 procedures, ***f, g*** and ***h***, which carry out the following steps:

f	***G***	***H***
1. Assign 1 to *x*	*1.* Assign 3 to *x*	*1.* Assign 6 to *y*
2. Assign 2 to *y*	*2.* Assign 4 to *y*	*2.* Display *x*
3. Call *g*	*3.* Call *h*	
4. Display *x*	*4.* Display *y*	

i. Assuming that ***x*** and ***y*** are global variables, write down the values displayed when ***f*** is called, in their order that they appear on screen. [6]

ii. Answer question (a) assuming that *x* and *y* are dynamically scoped local variables. [4]

QUESTION 4

a) What are the primary differences between static and dynamic binding. [2]

b) State 2 reasons why the **λ - calculus** is of interest to computer scientists. Evaluate the following expression: **((λx. x*x) ((λx. x-1) 4))** [4]

c) State and describe the 3 defining properties of an object. [6]

d) Rewrite the expression $\frac{b^2-4ac}{2a}$ in prefix form, i.e., using (+), (-), (*) and (/). [4]

e) With the aid of appropriate examples, explain the following terms:

i. Message passing [2]
ii. Lazy evaluation [2]
iii. Type checking [2]
iv. Functional abstraction [3]

QUESTION 5

a) Distinguish between: Axiomatic and denotational semantics. [8]

b) Briefly explain the single main difference between:

i. Machinelanguage and Assembly language
ii. Statements and expressions
iii. Selection control structure and repetition control structure [6]

c) State any 2 advantages of formal descriptions of semantics. [2]

d) Write a Haskell script that can be used to evaluate the expression: [6]

$$x = \frac{\sqrt{b^2 - 4ac}}{2a}$$

e) What is the output of executing the Haskell code: [3]

map (+3) [1 . . 5]

<<End of Question Paper>>