University of Swaziland
Final Examination
JULY 2013

Title of paper : Programming Languages

Course number : CS343

Time Allowed : Three(3) hours

Instructions :

- ***Each question carries 20 marks***
- ***Answer any five (5) questions from questions 1 to 6.***

This paper may not be opened until permission has been granted by the invigilator

QUESTION 1

(a)

(i) Briefly explain the purpose of parse trees. [2]

(ii) BNF is said to be a meta language. What is a matter language. [2]

(iii) Given the following BNF grammar:

(iv) <exp> ::= <term> + <exp> | <term> - <exp> | <term>

<term> ::= <factor> * <term> | <factor> / <term> I| <factor>

<factor> ::= (<exp>) | a | b | c | d |1 | 2 | 3 | 4

Construct the parse tree for the expression **a - b*(c+d).**

[6]

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

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

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

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

QUESTION 2

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

(b) Briefly explain the main difference between

(i) Compiler and Interpreter [2]

(ii) Statement and expression [2]

(c) Write a Haskell script that can be used to <u>evaluate</u> the expression:

$\mathbf{X =(\sqrt{b^2} -4ac) / 2a}$ [5]

(d) What is the output of executing the Haskell code:

map (+3) [1..5] [3]

QUESTION 3

(a) What are the primary differences between static and dynamic binding. [3]

(b) Discuss any two reasons for programming in a high-level language rather than low-level language. [4]

(c) Consider a language with 4 operators : ∨, ∧, > and <, that take numerical operands. Their syntactic properties are as follows:

Operator	Precedence	Arity	Fixity	Associativity
∨	0 (high)	1	postfix	Left
<	1	2	infix	Left
∧	2	2	infix	Right
>	3	1	Prefix	Right

Fully parenthesize the following expressions:

i. 1 < 2 ∧ 3 < 4 [2]

ii. > > 1 ∨ ∨ [2]

iii. (> 1 < (>2) < (<3)) ∨ < 4 [3]

iv. >1 <2∧3 ∧ 4∨∧ (>5∨<6∧7) [6]

QUESTION 4

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

(b) With the aid of examples, briefly describe 5 kinds of user defined types. [5]

(c) Define operator overloading [2]

(d) Explain why operator overloading is impossible in untyped languages. [3]

(e) With the aid of examples, define type safety [3]

(f) With aid of examples in C++/Java, define overloading polymorphism and parametric polymorphism. [5]

QUESTION 5

(a) With the aid of examples in Haskell, define the following terms:

i. Higher-Order function [2]

ii. Tail-recursive function [2]

(b) Define referential transparency and explain one of its benefits. [4]

(c) Based on the following type signature, describe the Haskell function ***fun*** as completely as possible:

Fun :: Integer -> Char -> [(Integer, Char)] [4]

(d) Rewrite the following Haskell function using pattern matching: [2]

```
describe x =
    if x == 0 then "Zero"
   else "non-zero"
```

(e) Prove that the following lambda-calculus function evaluates to 14, showing all steps:

(λx. ((λy.x*y) 2)) ((λx.x+((λy.y) 3)) 4) [6]

QUESTION 6

Define the following functions in Haskell including the type signature of each function.

(a) A function that, given two lists of identical length consisting of floating-point numbers, returns a list whose n-th element is the productof the n-th elements of the given lists. For example, given [5, 2.2, -3.3] and [-1.1, 1, -1], then the result is [-5.5, 2.2, 3.3] [4]

(b) A function that, given a string, returns the number of upper-case characters in the string. [8]

(c) A tail-recursive version of the following function that counts the number of elements in a list (but do not use Haskell's built-in length): [8]

```
count lst =
   if lst = = [] then 0
   else  1 + count (tail lst)
```