

University of Swaziland  
Department Of Computer Science  
DECEMBER 2013

*Title of paper:*                      *C under Unix*

*Course number:*                      *CS344*

*Time Allowed:*                      *Three (3) hours*

***Instructions:***

- *Answer ALL Questions in section A.*
- *Answer any two(2) questions in section B.*

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

- **Questions 1 and 3 are based on the following case description.**

### **Hospital Database Management System**

A Hospital maintains a database for assisting with administration of its wards and operating theatres, and for maintaining information relating to its patients, surgeons and nurses. The following is a description of the information in the database:

Most patients are assigned to a ward on admittance and each ward may contain many patients. However, consultants (senior surgeons) at the hospital may treat private patients who are also assigned to wards. The information to be recorded about a patient includes a unique medical number, name address, sex and age. A list of all admitted patients is kept and patients are added to this list on admittance and removed upon discharge.

Each ward has a unique ward number and name. A list of all the wards is kept. A nurse may or may not be assigned to a ward and he/she cannot be assigned to more than one ward. A ward may have many nurses assigned to it. Nurses are identified by staff number. The name and telephone number of the nurse is also recorded. A list of all nurses is kept and nurses are added to this list on employment and removed on termination of employment.

For each patient the medical history is recorded which include information such as the date of admission, a description of the treatment received and its cost, the medication given to the patient and the cost to the client., and the discharge date.

### **SECTION A**

- **ANSWER BOTH QUESTIONS IN THIS SECTION.**

#### **QUESTION 1 – 25 MARKS**

Using UML notation, draw a class diagram for the hospital database management system. Make sure to show attributes, multiplicities and aggregation/compositions, where appropriate. [25]

#### **QUESTION 2 – 25 MARKS**

- (i) Using C++ or C# or Java notation, define a template, class called **MyList**, that implements an **un-ordered** list. You are free to use either an array-based or pointer-based implementation. Your definition must contain all required code for the **constructor** function/methods, and methods to **insert**, **delete** and **display** all elements in the list. [16]
- (ii) Write a short main program that demonstrates how you may use the definition in (i) above to read 5 values from standard input, add the values to a list, and then display all the values on standard output. [5]
- (iii) What would you need to change in the above definition if you wanted your list to be **Ordered** instead of being **Un-ordered**. [4]

### **SECTION B**

- **ANSWER ANY TWO(2) QUESTIONS FROM FROM QUESTIONS 3 TO 5**

#### **QUESTION 3 – 25 MARKS**

- (i) Add design attributes to capture the association shown in the diagram obtained in question 1. [5]
- (ii) Using C++/C# or Java notation, write suitable class definitions for the hospital database system. For each class write suitable code for the required function/method(s), and at least one function/method to display information about each object of the class. [20]

**QUESTION 4 – 25 MARKS**

- (i) Explain the meaning of the following object-oriented terms
- (a) Encapsulation. [1]
  - (b) Inheritance. [1]
  - (c) Polymorphism. [1]
  - (d) Message Passing. [1]
- (ii) With the aid of examples, explain the main difference between the following terms.
- (a) Overloading and Overriding. [2]
  - (b) Concrete and Abstract class. [2]
  - (c) Private and Protected members. [2]
- (iii) Consider the following sequence
- $$\text{SUM}(N) = \sum 1/N!$$
- where  $N!$  is the factorial of  $N$  defined as follows
- $$N! = \begin{cases} 1 & \text{if } N = 0 \\ N * (N-1)! & \text{if } N > 0 \end{cases}$$
- Using C++ or C# or Java
- (a) write an iterative definition of the factorial function [4]
  - (b) write a recursive definition of the factorial function. [4]
  - (c) write a recursive definition of the summing function. [7]

