NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF COMPUTER SCIENCE
JULY SUPPLEMENTARY EXAMINATIONS 2001

SUBJECT: PROGRAMMING AND PROGRAM DESIGN
CODE: SCS1201

INSTRUCTIONS TO CANDIDATES

LIBRARY USE ONLY

1. Answer at least one question from each section
2. A total of 4 questions are to be answered
3. Each question carries 25 marks

3 HOURS

SECTION A

1. A library system is to be designed. Develop an IPO (input, Process, Output) for the basic operation of the library system. [25]

2. A simple banking system is required with the following operations;

   i) Open a deposit account. [6]
   ii) Open a current account [6]
   iii) Close an account [8]
   iv) Make a deposit or withdrawal [5]

   Develop an IPO for these operations.

SECTION B

3. A chain store file contains records each containing a department code, record type indicator and other fields. A type 1 record contains the start-of-day each figure. A program is required that will detect errors and store correct data in an accepted file. If the data for a department are valid, they are written to the accepted file. If invalid in any way, the data are rejected and the following message id displayed:

   INCORRECT DATA FOR DEPARTMENT nnnnn

   Data for a department are valid if:

i) There are exactly 2 records for the department, a type 1 and a type 2 record, for the department, a type 1 and a type 2 record in that order.
ii) The value of both cash fields in the range of $100 to $100 000.
iii) The value of cash at the end of day on the type 2 record is strictly greater than the value of cash at the start of the day on day on the type 1 record.

The records are sorted into ascending order of record type indicator within department code.

Design the program to detect the above errors and produce appropriate output by following these steps:

a) Construct logical data structures for

- the chain store file [4]
- the accepted file [3]

b) identify correspondences and produce a program structure n.
c) List the elementary operations and conditions.
d) Allocate these to the program structure.
e) Revise the program structure to include body boxes in necessary.
f) Produce schematic logic.
g) Revise the schematic logic to introduce as appropriate POSIT, ADMIT and QUIT.
h) Classify any side effects.
i) Amend the operations list and condition list to cater for side effect processing and QUITs.
j) Revise the schematic logic to incorporate side effect processing.

[b-j 2 marks each]

4. The following are the informal requirements for oil terminal control system, a simple system to control the allocation of berths to tankers in an oil terminal.

An oil terminal has a number of berths at which tankers can discharge their cargoes. When an approaching tanker asks for permission to clock, the controller will ask the system to allocate the berth for it to use. If no berth is free, the system will tell the controller so, and the tanker will be queued in the approach to the terminal. The system assumes that the there is enough queuing space for any number of tankers. On clocking, a tanker occupies the allocated berth and unload its cargo. When it is ready to leave, the controller will notify the system so that the berth is ready for

reuse, and the tanker is detected from the system. A tanker leaving a berth might mean that a queuing tanker can come and occupy it. The system will identify the tanker at the head of the queue to the controller, and allocate the berth to the tanker. The system has enquiry facilities so that the controller can get information about which tankers, and which berths are free.

a) Identify ambiguities and inconsistencies in these requirements, and record them in the form of questions directed to the person supplying the requirements. [8]

b) Obtain or invent classification, and then construct a tabular description of these requirements. [8]

c) Write a Z specification to formalise the requirements. [9]

SECTION C

5. a) What are the characteristics of JSP. [8]

b) Outline the stages of JSP. [8]

c) Specifically, what must be done in caring up with a computer program. [9]

6. Develop an algorithm for a merge sort or bubble sort. [25]

END OF QUESTION PAPER