UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B.Sc. Examination 2012

COMPUTING AND INFORMATION SYSTEMS

IS52018B Software Projects: Software Engineering and Research Methods

Duration: 1 hour 30 minutes

Date and time:

There are THREE questions on this paper. You should answer no more that TWO questions. Full marks will be awarded for complete answers to a total of TWO questions. The marks for each part of a question are indicated at the end of the part in [.] brackets.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Question 1.

a) Describe three ways in which the reliability of software differs from that of hardware. **[5]**

b) What, conceptually, are Unified Modelling Language (UML) diagrams and how can they be used? **[3]**

c) Consider the task to design a Dormitory management system software. We think of a dormitory as a general kind of area: two specific cases are Accommodation and Canteen. The Canteen contains a number of seats, a number of tables and allows users to reserve a seat. An accommodation has three parts: a Room, an Officer and a Receptionist.

- The Receptionist should have a budget and provide functions to calculate balances and manage the budget;
- The Officer works with one or more Student, providing services such as searching for a free room, telling the price for using a room, taking money, and giving out keys;
- The Student should have money in order to make payments if he needs a room, which he can indicate by requesting a room;
- The Room contains rooms, which may be occupied or not, and functions reserve an unoccupied room, and find occupation status;
- The Accommodation uses no more than two Officers and manages all the Rooms in the Dormitory by keeping info for the free rooms, and checking the availability of free rooms.

Develop a UML class diagram for this software Dormitory system (the nouns in capital letters should be classes), and suggest class attributes with names and type, as well as methods with name and return type. Illustrate associations, aggregation and generalisation relationships. **[17]**

Question 2.

a) i) Explain briefly the main difference between UML state machine and sequence diagrams. **[4]**

ii) When should we use UML sequence diagrams in the software modelling process? **[3]**

b) What are the components of a label of a UML state machine diagram. What does each indicate? **[4]**

c) Develop a UML state machine diagram for processing student in a software Dormitory management system by modelling a scenario for the behaviour of the officer:

- The officer is initially in a waiting state;
- When a student arrives and requests a room the officer has to search for a free room using the system. This is a new state, called *room searching* state;
- There are two possible continuations from this room searching state: either the system returns to the initial waiting state if a free room has not been found, or when a free room has been found the system informs the user about the room price;
- This leads the system to the *serving* state where if the student has money the officer takes the money, or if the student has no money it rejects accommodation and goes back to the initial waiting state. If the officer receives payment the system moves to the *payment made* state, and after that gives the key to the student.

Finally, the scenario terminates by returning to the initial waiting state where it started. Design the state machine diagram using states, transitions and labels. **[14]**

Question 3.

a) Describe three approaches to computing the cyclomatic complexity of a program. **[9]**

b) The following function needs to be tested:

```
void Compute( int n, int *List )
{
   int i, j, k, change, temp;
   k = 0; change = 1;
   while (( k < n-1 ) && change )
   {
      change = 0; k++;
      for ( j = 0; j < n-k; j++ )
      {
         if ( List[ j ] > List [ j+1 ] )
         {
            temp = List[ j ]; List[ j ] = List[ j+1 ]; List[ j+1 ] = temp;
            change = 1;
         }
      }
   }
}
```

i) Draw the flow graph of this function. **[4]**

ii) Determine the cyclomatic complexity of the flow graph. **[4]**

iii) Determine the basis set of linearly independent testing paths. **[8]**