27

Engg. SWE 8th Semester

15 May 2023

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **SUMMER SEMESTER, 2021-2022**
DURATION: 3 HOURS **FULL MARKS: 150**

SWE 4805: Software Verification and Validation

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

---

1. Suppose you are developing a new software system, SHEBA, for a hospital that allows doctors and nurses to manage patient information and appointments. The system needs to be reliable, secure, and user-friendly, as patient's health and safety depend on its proper functioning.

   a) Define Software Verification and Validation. What aspects of developing SHEBA can motivate a practitioner to study software verification and validation? $2 + 3$ (CO2) (PO1)

   b) Describe the activities that you will perform in the verification and validation steps of developing SHEBA. 10 (CO2) (PO1)

   c) Considering the scenario, which functionalities of the system will you test during black, white, and gray box testing? Justify your answer. $3 + 3 + 4$ (CO1) (PO1)

2. a) How can model-based testing be used to verify and validate non-functional requirements, such as performance and reliability? $2 + 3$ (CO2) (PO1)

   b) Describe three types of automated test case generation approaches with their pros and cons. 10 (CO2) (PO2)

   c) Analyze the following myths of formal method and propose corresponding facts: 10 (CO2) (PO1)
      i. Formal methods replace the traditional engineering design methods.
      ii. Formal methods are not used on real, large-scale software.

3. Consider the module of a large system written in C programming language shown in Code Snippet 1 and answer the subsequent questions:

```
int largestNumber(int a, int b, int c, int d) {
    int result = a;
    if(!( a < b || a < c || a < d )) result = a;
    else if(!( b < c || b < d )) result = b;
    else if(!( c < d )) result = c;
    else result = d;
    return result;
}
```

Code Snippet 1: A C program for finding the largest number for Question 3.

   a) Write a test plan having one (1) test case of each of the following criteria: 5 (CO1) (PO1)
      i. Statement coverage
      ii. Decision/branch coverage

iii. Condition coverage

iv. Multiple condition

v. Boundary Value Analysis

In your test plan, only consider the input and its corresponding expected output.

b) Create three types of mutants considering the code snippet and calculate the mutation score using your test plan. (CO (PO

c) After performing the test mentioned in Question 3.b), you found some errors and changed the code accordingly. What are the tests that you need to perform to verify and validate the given module as well as the whole system? 10 (CO2) (PO2)

4. Consider the Alloy code shown in Code Snippet 2 and answer the subsequent questions:

```
abstract sig FSObject {
    parent: lone Directory
}
sig File extends FSObject{}
sig Directory extends FSObject{
    entries: set Entry
}
sig Entry {
    name: one Name,
    object: one FSObject
}
sig Name {}
one sig Root extends Directory {}{no parent}
```

Code Snippet 2: Alloy code of a file system for Question 4.

a) Describe Code Snippet 2 in plain English. 5 (CO1) (PO1)

b) Explain the following facts in plain English. 5 × 2 (CO2) (PO1)

i. `File + Directory = FSObject`

ii. `all fs:Entry.object | fs in Directory => fs not in fs.^parent`

iii. `FSObject = Root.*(entries.object)`

iv. `all d: Directory | d not in d.^(object[entries])`

v. `all d: Directory, e1, e2: d.entries | e1 != e2 implies object[e1].parent != object[e2].parent && e1.name != e2.name`

c) Assuming all the facts mentioned in Question 4.b) are incorporated, analyze the following assertions independently and justify whether a counterexample is found or not: 5 × 2 (CO2) (PO2)

i. `Root.^(entries.object)= FSObject-Root`

ii. `all disj d1,d2:Directory| # (d1.entries & d2.entries)= 0`

iii. `all d:Directory, disj e1, e2:d.entries| e1.name != e2.name`

iv. `one fs:FSObject | # fs.parent = 0`

v. `all d:Directory, o: entries[d].object| o.parent = d`

IUT has developed a system to maintain the examination routine. However, it is not properly verified and validated. The specification of the system are given below:

Every exam comprises a collection of details, including the courses for which the exam will be conducted, the date on which it will be held, and information pertaining to seating arrangements. The seating arrangement contains three pieces of information: the student taking the exam, the room, and the assigned seat where the student will be seated for the exam. Every seat has a unique number. Rooms are identified by a unique room number. They belong to an academic building, have a capacity, and have a set of labelled seats. Each room is configured in a manner where ten students can sit in a single row. A student can be identified by a unique ID and studies in a specific semester. Each student also belongs to a specific department, program, batch, and has taken some courses. A course is identified by a code and name, while a semester is associated with a program which has a list of offered courses. The department can be CSE, EEE, CEE, or MPE. Every department has some programs, where every program has some required courses. For example, CSE department has two programs, B.Sc. in CSE and B.Sc. in SWE.

Apart from these, there are the following constraints:

- All students must belong to a valid department and program for an exam.
- In the exam hall, students of every program can sit only for the courses that they are enrolled in their current semester.
- All students in an exam must fit with in the rooms' capacity.
- No two programs of any department can have the identical offered course list as a whole.
- In any given room, it is prohibited to seat two students of the same program of any department consecutively.

IUT has hired and instructed you to verify and validate the following statements:

- No student is assigned to two exams in the same day.
- It is allowed to have 70% similarities of the offered courses of any two programs of the same department, but 30% for the different departments.
- For an assigned seat of a program's student, there is no other student around him/her.
- Each room can only be occupied by students from a single batch.
- For any program, there must not be any duplication of courses of different semesters.

a) Analyze the specification and constraints and write the signatures with their associated fields and facts. 5+10 (CO1) (PO2)

b) Write the given statements as assertions to verify and validate in Alloy. Justify whether a counterexample is found or not for each one. Assume that all the facts mentioned in Question 5.a) are incorporated. 5 × 2 (CO2) (PO2)

6. Consider a prototype of a virtual networking system that comprises of nodes and messages. Each node possesses a distinctive name, IP address, port number, physical location, and status that indicates whether it is active, inactive, or suspicious. In this system, the status of the node is determined by the activity of the node. Any node within the system is capable of sending and receiving messages, with each message containing a designated source and destination node, timestamp, and a payload. To facilitate easy monitoring and tracking of the prototype, three locations, namely Dhaka, Rajshahi, and Chattogram, have been predetermined.

   a) Analyze the specification and write the signatures with their associated fields and facts. 5 (CO1) (PO2)

   b) Write the following constraints in Alloy as facts, considering the signatures and fields that you have designed in Question 6.a). 5 × 2 (CO2) (PO1)

      i. In the network, all messages must have a valid source and destination address.

      ii. All messages must be sent between nodes within the same location.

      iii. Active nodes in the network must be able to send and receive messages.

      iv. Suspicious and inactive nodes in the network must not be able to send and receive messages.

      v. All nodes must be within a certain distance from at least one other node in the network.

   c) Write the following statements as assertions to verify and validate in Alloy. Justify whether a counterexample is found or not for each one. Assume all the facts mentioned in Question 6.b) are incorporated. 5 × 2 (CO2) (PO2)

      i. All nodes must be able to send and receive messages.

      ii. There are some messages whose source and destination nodes are the same.

      iii. There might be some active nodes whose IP addresses, ports, and locations are not valid.

      iv. All nodes with an inactive or suspicious status must not have a location.

      v. All messages sent to a node in Rajshahi must have a timestamp between 1000 and 2000 with a "checksum: OKAY" payload.