ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING
SOFG1008 - SOFTWARE ENGINEERING II
SECOND EXAMINATION - I PAO 2024

***Name:* ____ *Student ID:* ______________**

Section A

1. Program *Check* in figure below computes negative numbers from a given input stream.
   a. Draw the DD-path graph for the *Check* program. Annotate the graph showing the basic node types that occur in it. List all the independent paths in the DD-path graph. **[16%]**
   b. How many test cases would be needed for 100% Statement Coverage (C_0 metric) **[04%]**
   c. Specify the test cases which ensure the 100% Path Coverage (C_1) of the *Check* program. Use a simple form of test case specification, i.e., a table with the following headings:

Test Case ID	Input Values	Expected Values

   As a part of your answer: (i) Explain what is meant by the Path Coverage; (ii) Explain in detail what principles you have used to generate your test cases. **[14%]**
   d. Do the tests generated using the Path Testing method make you confident that you will detect most of important bugs? Why? **[06%]**

```
1   Program Check
2
3   Count, Sum, Index: Integer
4
5   Begin
6
7   Index = 0
8   Sum = 0
9   Read (Count)
10  Read (New)
11
12  While Index <= Count
13  Do
14      If New < 0
15      Then
16          Sum = Sum + 1
17      Endif
18      Index = Index + 1
19      Read (New)
20  Enddo
21
22  Print (“There were”, Sum, “negative numbers in the input
    stream”)
23
24  End
```

Section B

Justify your answer. The use of pencil or a correction in the selection invalidates the answer.

2. What is a definition-use pair? **[05%]**
   a. The association of a definition of a variable with the subsequent use of that variable.
   b. The association of an executable statement in the source code with the use of that statement in the execution of the code.
   c. The association of a comment in the code that describes the meaning of a variable with the subsequent use of that variable in the code.
   d. The association between the definition of the behaviour of the software in the specification and the code that implements that behaviour.

3. Below is the pseudo-code for a program that calculates and prints sales commissions: **[05%]**

```
00  program Calculate Commission
01  total, number : integer
02  commission_hi, commission_lo : real
03  begin
04     read ( number )
05     while number ≠ -1 loop
06          total = total + number
07          read ( number )
08     endloop
09     if total > 1000 then
10          commission_hi = 100 + 0.2 * ( total - 1000 )
11     else
12          commission_lo = 0.15 * total
13     endif
14     write ( "This salesman's commission is:")
15     write ( commission_hi )
16  end program Calculate Commission
```

The code contains data flow anomalies on lines 6 and 12 (highlighted text). Which examples of data flow anomalies are to be found on these lines? Select exactly ONE option.

a. line 6: variable "total" is not assigned a value before using it
   line 12: variable "commission_lo" is defined but subsequently not used
b. line 6: an invalid value is assigned to variable "total"
   line 12: variable "commission_lo" is redefined before it is used
c. line 6: variable "total" is out of scope
   line 12: the "hard-coded" value "0.15" should not be used
d. line 6: the variable "number" is undefined
   line 12: the variable "total" is redefined before it is used

4. You have just started designing test cases for the following user story. **[05%]**

As a customer,
I want to be able to filter search results by price range, so that I can find products within my budget more easily.
Acceptance criteria:

1. The filter should work for all versions of the application from version 3.0 upwards
2. The filter should allow the customer to set a price range with a minimum and a maximum price
3. The search results should update dynamically as the customer adjusts the price range filter

In all test cases the precondition is as follows: there are only two products available, products A and B. Product A costs $100 and product B costs $110.
Which of the following is the BEST example of a test case for this user story? Select ONE option.

a. Enter webpage and set filter to show prices between $90 and $100. Expected result: results show product A only. Set maximum price to $110. Expected result: results now include both products A and B

b. Enter webpage. Expected result: the default minimum and maximum prices are $100 and $110 respectively. Add product C to stock, with price $120. Refresh the client's webpage. Expected result: the default maximum price changes to $120
c. Enter webpage and set filter to show prices between $90 and $115. Expected result: results show both products A and B. Change currency from USD to EUR. Expected result: the filter range changes correctly to EUR values, according to the current exchange rate
d. Enter webpage with three different browsers: Edge, Chrome and Opera. In each browser set filter between $90 and $110. Expected result: results include both products A and B and the results layout is the same in all three browsers

5. Which of the following principles should be followed when introducing a test tool into an organisation? **[05%]**
   i. Assessing organisational maturity to establish whether a tool will provide expected benefits.
   ii. Requiring a quick payback on the initial investment.
   iii. Including a requirement for the tool to be easy to use without having to train unskilled testers.
   iv. Identifying and agreeing requirements before evaluating test tools.

   a. i and ii.
   b. i and iv.
   c. ii and iii.
   d. iii and iv.
6. Why are both specification-based and structure-based test design techniques needed? **[05%]**
   a. Because specification-based techniques do not provide coverage measures.
   b. Because structure-based techniques can only test code.
   c. Because both are needed to improve the chances of finding defects.
   d. Because neither can take advantage of users' and testers' experience of the type of system being tested.
7. How does software testing contribute to the quality of delivered software? **[05%]**
   a. By detecting and removing all the defects in the delivered code and ensuring that all tests adhere to the quality standards set for the project.
   b. By measuring reliability of the software and ensuring that it is always above 99.99 per cent.
   c. By detecting all deviations from coding good practice and ensuring that these are corrected.
   d. By identifying root causes of defects from past projects and using lessons learnt to improve processes and thus help to reduce the defect count.
8. Which testing is used to verify that the system can perform properly when internal program or system limitations have been exceeded **[05%]**
   a. Stress Testing
   b. Load Testing
   c. Performance Testing
   d. Volume testing
9. Which of the following is **not** true of regression testing? **[04%]**
   a. It can be carried out at each stage of the life cycle.
   b. It serves to demonstrate that the changed software works as intended.
   c. It serves to demonstrate that software has not been unintentionally changed.
   d. It is often automated.

Section C

10. Compare and contrast the following terms in the context of software testing: **[09%]**
    a. load testing
    b. stress testing
    c. volume testing
11. Suggest appropriate reliability metrics for the classes of software system below. Give reasons for your choice of metric. **[12%]**
    a. a software controlling an insulin delivery system for patients in a hospital intensive care unit.
    b. a spreadsheet editor.
    c. an automated train ticket machine control system.
    d. An operating software for a chemical plant, whose failure may cause a serious pollution incident.
    e. a system to control a basement dehumidifier.
    f. a management dashboard builder.