higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

MARKING GUIDELINE

NATIONAL CERTIFICATE (VOCATIONAL)

INTRODUCTION TO SYSTEM DEVELOPMENT

(First paper)

NQF LEVEL 2

11 March 2020

This marking guideline consists of 5 pages.

Copyright reserved

SECTION A

QUESTION 1

1.1 Apple
1.2 Functionality
1.3 Linux
1.4 Graphical
1.5 Variable
1.6 System
1.7 Validation
1.8 Language translators
1.9 Proprietary
1.10 Unicode

(10 × 1) **[10]**

QUESTION 2

2.1 False
2.2 True
2.3 True
2.4 False
2.5 False
2.6 True
2.7 False
2.8 False
2.9 True
2.10 False

(10 × 1) **[10]**

TOTAL SECTION A: 20

QUESTION 3

3.1
- An operating system
- A utility program
- A language translator (3)

3.2
- Multi-user
- Multi-processing
- Multi-tasking
- Multi-threading
- Real time (5)

3.3
- Pre-release stage
- Testing phase
- Post-release (3)

[11]

Copyright reserved

QUESTION 4

4.1 4.1.1 Object-oriented programming – is a modular programming language.

4.1.2 Visual programming languages – lets the user specify programs in a two-dimensional way by manipulating elements graphically rather than specifying them textually.

4.1.3 Event-driven programming is a programming technique for creating responsive programs.

4.1.4 Rapid application development – is a software application that helps a programmer design a program faster by generating the source code automatically.

(4 × 2) (8)

4.2
- Opportunistic or accidental reuse
- Systematic or deliberate reuse

(2)
[10]

QUESTION 5

5.1 Variable – is a named item that stores a value in a program.

5.2 Data types – are the different forms of data that a computer can recognise.

5.3 Constant – is a named item with a value that does not change.

5.4 Procedures – is a group of instructions that tell a computer how to do a task.

5.5 Functions – is a procedure that always takes in a value or values, works with the values and returns a single value as a result.

(5 × 2) **[10]**

QUESTION 6

6.1
- Flow charts
- Nassi-Schneidermann diagram
- Decision tables
- Decision trees
- Use case diagrams in UML

(Any 4 × 1) (4)

6.2
- NOT operator
- AND
- OR

(3)

6.3 6.3.1 Terminal symbol
6.3.2 Processing symbol
6.3.3 Decision symbol
6.3.4 Module symbol
6.3.5 IO symbol
6.3.6 The connector line

(6 × 1) (6)

6.4 T = 3 * 5 ^ (13-11) mod 4 + 3 \ 3✓✓
= 3 * 5 ^ **2** mod 4 + 3 \ 3✓
= 3 * **25** mod 4 + 3 \ 3 ✓
= **75** mod 4 + 3 \ 3✓
= 75 mod 4 + **1**✓
= **3** + 1✓
= **4**✓
T = 4 (8)

6.5

I	P	O
length✓ width✓	Area = length * width ✓✓ Perimeter = (length*2) + (width*2)✓✓	perimeter✓ area✓

(8)

6.6 Substitute:
TRUE OR FALSE AND NOT TRUE✓
TRUE OR FALSE AND **FALSE**✓
TRUE OR **FALSE**✓
TRUE✓✓ (6)
[35]

QUESTION 7

7.1
- Ensure that the indentation style is consistent throughout the program
- Indent code within a block
- Avoid excessive indentation (3)

7.2
- They can detect and correct errors while these are still inexpensive to correct.
- They improve the quality of the product.
- They provide a good resource for improving software products.
- They establish a good metrics database of planning, as well as reliability and failure predictions. (4)

7.3
- Use a test plan
- Use test data
- Do syntax checking (3)

7.4
- Feasibility – is an evaluation to determine whether a project or product will be a profitable investment.
- Viability – is the ability of the project or product to perform the tasks required, and to adapt to the changing environment. (2 × 2) (4)

[14]

TOTAL SECTION B: 80
GRAND TOTAL: 100

Copyright reserved