higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

MARKING GUIDELINE

NATIONAL CERTIFICATE (VOCATIONAL)

COMPUTER PROGRAMMING
NQF LEVEL 4
(First paper)

27 November 2024

This marking guideline consists of 7 pages.

Copyright reserved

SECTION A

QUESTION 1

1.1 B
1.2 B
1.3 D
1.4 A
1.5 C

(5 × 1) **[5]**

QUESTION 2

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

(10 × 1) **[10]**

QUESTION 3

3.1 File Transfer Protocol
3.2 Third Normal Form
3.3 Transfer Control Protocol
3.4 Secure Socket Layer
3.5 Document Object Model

(5 × 1) **[5]**

QUESTION 4

4.1 runtime
4.2 <!-- -->
4.3 String.Contains()
4.4 ===
4.5 20px

(5 × 1) **[5]**

TOTAL SECTION A: 25

Copyright reserved

SECTION B

QUESTION 5

5.1 A **base class** (also known as a superclass) is a general class that provides common properties and methods which other classes can inherit. ✓ It is the "parent" class in the class hierarchy✓, designed to provide shared behaviour to its subclasses.✓ It can be instantiated to create objects, ✓ unless it is declared as an abstract class. ✓

A **subclass** is a class that inherits from a base class. ✓ It extends the functionality of the base class✓, adding or modifying properties and methods. ✓ The subclass is the "child" class in the class hierarchy. ✓
[OR ANY OTHER RELEVANT DIFFERENCES] (Any 2 × 2) (4)

5.2 5.2.1 Cohesion in Object-Oriented Programming (OOP) refers to the degree to which the elements (methods, attributes) of a class are related and work together to accomplish a single, well-defined task✓. In other words, a class is said to have high cohesion if all of its components (methods and fields) are focused on a single responsibility or purpose.✓ (2)

5.2.2 Encapsulation is the concept of wrapping data and methods together into a single a class and controlling access to the inner workings of that class✓. It enhances security, integrity, and maintainability by hiding the internal state and providing controlled access through well-defined interfaces. This helps ensure that the object behaves in a consistent and predictable way. ✓ (2)

5.2.3 Polymorphism allows objects of different classes to be treated as instances of the same class through a common interface, typically a base or parent class✓. In simple terms, polymorphism enables a single action to behave differently depending on the object that performs it. ✓ (2)

[OR ANY OTHER RELEVANT VARIATION OF THE ABOVE]

5.3 5.3.1

```
Sub New ✓ (ByVal value1 As Decimal✓, ByVal value2 As
Decimal✓, ByVal value3 As Decimal✓)
    NormalHours = value1✓
    OvertimeHours = value2✓
    HourlyRate = value3✓
End Sub
```

[ANY PARAMETER / ARGUMENT NAMES CAN BE USED] (7)

5.3.2

```
Public ReadOnly Property PaymentDue As Decimal✓
  Get✓
   Return✓((NormalHours+OvertimeHours)*HourlyRate) ✓
  End Get
End Property
```

(4)

5.4 5.4.1 An **object list** generally refers to a collection or list✓ that stores multiple objects of a specific type. ✓
[ANY OTHER ACCEPTABLE VARIANT OF THE ABOVE] (2)

5.4.2 personList.Add✓ (New Person✓ ("James Longwind", 28) ✓) (3)
[(28, “James Longwind”) is also acceptable] **[26]**

QUESTION 6

6.1 6.1.1 One-to-Many✓ (1)

6.1.2 A one-to-many relationship in a database means that a single record in one table✓ can be related to multiple records in another table, while each record in the second table✓ relates to only one record in the first.
[OR ANY OTHER VARIANT OF THE ABOVE DEFINITION] (2)

6.1.3 (PK) `rental_customers.custid` ✓
(FK) `tools.custid` ✓ (2)

6.1.4
- Data integrity
- Security
- Data availability
- Efficiency
- Scalability
- Concurrency control (Any 4 × 1) (4)

6.1.5
```
SELECT ✓
    rental_customers.custid, ✓
    rental_customers.firstname,
    rental_customers.surname,
    tools.tool_name,
    tools.date_out,
    tools.returned_yn✓
FROM rental_customers ✓
JOIN✓ tools ON ✓
    tools.custid = rental_customers.custid ✓
WHERE✓ tools.returned_yn='NO' ✓ {='N' or 'n' or 0 (for false), etc.}
ORDER BY✓
    rental_customers.surname✓
```

NOTE: Variations include using INNER JOIN, table aliases and since both tables contain the `custid` column one can also use the USING statement i.e., `INNER JOIN tools USING(custid)`

(11)
[20]

QUESTION 7

7.1 It contains metadata about the document, such as links to stylesheets, scripts, and the document title
[OR ANY OTHER RELEVANT VARIANT OF THE ABOVE] (1)

7.2 Block-level elements start on a new line and take up the full width available (e.g., `<div>`, `<p>`), ✓while inline elements do not start on a new line and only take up as much width as necessary (e.g., `<span>`, `<a>`). ✓
[OR ANY OTHER RELEVANT VARIANT OF THE ABOVE] (2)

7.3 7.3.1 `<script>` (1)
7.3.2 `<link>` (1)
7.3.3 `<nav>` (1)
7.3.4 `<section>` (1)

7.4 The CSS box model consists of four parts: content, padding, border, and margin. ✓ It defines how elements are sized and spaced on the page. ✓
[OR ANY OTHER RELEVANT VARIANT OF THE ABOVE] (2)

7.5 `margin: 0 auto;` (1)

7.6 It controls the stacking order of elements that overlap. Elements with a higher `z-index` appear in front of elements with a lower `z-index`. (1)

7.7 JavaScript is used to add interactivity, manipulate HTML and CSS✓
and handle events dynamically in a web page. ✓ (2)

7.8 1. incorrect comment indicator (should be // or /* */) ✓
2. missing comma (,) between parameters ✓
3. `message` is not declared (or undefined) ✓
5. missing closing parentheses “)”✓

(Any 3 × 1) (3)
[16]

QUESTION 8

8.1 **Source Credibility**: The credibility of the source is one of the most important indicators of reliability. Reliable sources are typically associated with institutions, organizations, or individuals that have expertise and authority in their respective fields. ✓✓

Accuracy and Verifiability: Reliable information is accurate, meaning it is fact-based, free from errors, and consistent with other reputable sources. The content should be based on verifiable facts, meaning that it can be cross-checked with other trustworthy sources. ✓✓

Timeliness: The timeliness of the information refers to how recent or up-to-date the content is. In fast-evolving fields like technology, science, and current events, information that is old may no longer be accurate or relevant. ✓✓

Objectivity and Bias: Reliable information is typically objective and presents a balanced view of the topic, free from bias or undue influence from external sources. Bias can distort facts to promote a particular viewpoint, agenda, or product. ✓✓

Purpose and Intended Audience: The purpose of the content affects its reliability. Information intended to inform or educate is typically more reliable than content aimed purely at entertaining or selling something. The intended audience also matters because information targeting experts may be more detailed and accurate than content aimed at casual readers**.** ✓✓

Design and Functionality: While design and functionality alone do not determine the reliability of information, a well-designed website often signals professionalism and attention to detail. Trustworthy websites tend to have a clear structure, easy navigation, and professional presentation. ✓✓

Domain and URL: The domain name can also give clues about the reliability of the information. Certain domains are more trustworthy because they are typically regulated or associated with reputable entities. ✓✓

Peer Review and Editorial Process: Content that has undergone a peer review or editorial process is typically more reliable. Peer-reviewed articles in academic journals are evaluated by experts in the field before publication. Similarly, websites that have a clear editorial process ensure the quality of their information. ✓✓

User-Generated Content: User-generated content, such as posts on forums, social media, and open-access platforms (like Wikipedia), can be useful but should be approached cautiously. While user-generated content can provide diverse perspectives, it is often not fact-checked and may be inaccurate. ✓✓

Plagiarism and Misinformation: Reliable sources provide original content or properly cite their sources. Watch out for plagiarism (content copied without crediting the original author) and misinformation (false or misleading information). ✓✓ (Any 2 × 2) (4)

8.2 Planning and Requirement Analysis
Wireframing and Prototyping (Design Phase)
Front-End Development
Back-End Development
Content Creation and Management
Testing and Quality Assurance (QA)
Deployment and Launch
Maintenance and Updates
Search Engine Optimization (SEO) (Any 2 × 1) (2)

8.3 Explains the Purpose and Functionality of the Code
Provides Insight into the Code's Structure and Logic
Facilitates Easier Maintenance and Debugging
Improves Collaboration and Onboarding
Documents APIs and Usage for External Developers
Supports Code Reusability
Clarifies Configuration and Environment Setup
Helps Ensure Consistency and Adherence to Best Practices
Supports Future Proofing and Scalability
Documents Compliance and Legal Requirements
(Any 3 × 1) (3)

8.4 **ADVANTAGES**

- Increased Efficiency and Productivity
- User-Friendly Interface for Non-Coders
- Responsive Design and Mobile Optimization
- Version Control Integration
- SEO and Performance Optimization

DISADVANTAGES

- Lack of Control and Flexibility in WYSIWYG Editors
- Over-Reliance on Templates and Frameworks
- Performance and Code Bloat
- Higher Cost of Professional Tools
- Learning Curve for Advanced Tools
- Platform and Compatibility Issues
- SEO and Performance Limitations in Drag-and-Drop Builders

(Any 2 × 2) (4)
[13]

TOTAL SECTION B: 75
GRAND TOTAL: 100

Copyright reserved