|  |
| --- |
| **FURTHER EDUCATION AND TRAINING CERTIFICATE: INFORMATION TECHNOLOGY: SYSTEMS DEVELOPMENT**  **ID 78965 LEVEL 4 – 165 CREDITS** |
| **LEARNER WORKBOOK**  **SAQA: 14917**  **EXPLAIN COMPUTER ARCHITECTURE CONCEPTS** |

**Learner Information:**

|  |  |
| --- | --- |
| **Details** | **Please Complete this Section** |
| Name & Surname: | Mila Ngewu |
| Organisation: | Nelson Mandela Bay Ihub |
| Unit/Dept: | ICT |
| Facilitator Name: | Anneline Nombeko |
| Date Started: | 17 October 2023 |
| Date of Completion: | 18 October 2023 |

**Copyright**

All rights reserved. The copyright of this document, its previous editions and any annexures thereto, is protected and expressly reserved. No part of this document may be reproduced, stored in a retrievable system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission.

**OVERVIEW**

|  |  |  |  |
| --- | --- | --- | --- |
| **About the Learner Workbook** | This Learner Exercise Workbook has been designed and developed to evaluate learners’ level of understanding of the  **EXPLAIN COMPUTER ARCHITECTURE CONCEPTS.** It forms part of a series of Learner Workbooks that have been developed for **FURTHER EDUCATION AND TRAINING CERTIFICATE: INFORMATION TECHNOLOGY: SYSTEMS DEVELOPMENT ID 78965 LEVEL 4 – 165 CREDITS** | | |
| **Purpose** | The purpose of this Learner Exercise Workbook is to evaluate learners understanding on the specific outcomes and/or assessment criteria of the following SAQA Registered Unit Standards: | | |
| **US NO** | **US TITLE** | **LEVEL** | **CREDITS** |
| **14917** | **EXPLAIN COMPUTER ARCHITECTURE CONCEPTS** | **4** | **7** |
| **Context** | This assessment represents the Formative Assessment component of the **FURTHER EDUCATION AND TRAINING CERTIFICATE: INFORMATION TECHNOLOGY: SYSTEMS DEVELOPMENT ID 78965 LEVEL 4 – 165 CREDITS** and should be completed in the classroom/training room. | | |
| **Resources** | The following are resources needed for this assessment:   1. Learner Guide; and 2. Assessment Preparation. | | |
| **Instructions to Facilitators** | Facilitators will be required to:   * Explain the completion of the workbook to each learner; and * Interview the learner on similar questions, should he/she not be able to write. | | |
| **Instructions to Learners** | Learners will be required to:   * Complete the workbook as per the instructions; * Ensure that all questions are completed; * Ensure that the completion of the workbook is their own work; * Ensure that all annexure are attached to the workbook and clearly referred to; | | |
| **Assessment Time** | Learners are required to complete this assessment within the allocated time frame of.... hours. | | |
| **Total Mark** | This formative assessment carries a total mark of **\_ points**. In order to meet the pass mark, learners are required to achieve a minimum of **80%** of the total marks. | | |
| **Equipment** | Learners are required to have the following equipment in order to complete this workbook:   * Pen and Pencil; * Ruler; and * Exam Pad – for additional paper. | | |

# **GENERAL INFORMATION**

|  |  |  |
| --- | --- | --- |
| **LEARNER DETAILS** | | |
| **Learner Full Names** | **Ngewu Mila Mihlali** | |
| **Learner ID No.:** | **9909106615084** | |
| **Organisation:** | **Nelson Mandela Ihub** | |
| **Unit/Dept:** | **9909106615084** | |
| **Contact Details:** | **Telephone /Cell Numbers:** | **Email Address:** |
| **0823655804** | **0823655804** |
| **WORKSHOP DETAILS** | | |
| **Workshop Venue:** | **Nelson Mandela Bay iHub** | |
| **Facilitator Name:** | **Anneline Nombeko** | |
| **Date Started:** | **17 October 2023** | |
| **Date Completed:** | **18 October 2023** | |

# **ASSESSMENT PREPARATION CHECKLIST**

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPTION** | **YES** | **NO** | **COMMENTS/CONTINGENCY** |
| This assessment is a formative assessment and it is based on the outlined unit standard/s for the **Explain computer architecture concepts** module. | ✓ |  |  |
| Your assessment evidence for **Explain computer architecture concepts** module needs to be submitted on....... (day) of...............(month)...........(year) at the following address/place................................................................ | ✓ |  |  |
| You will be assessed based on the outlined Unit Standards. The assessment activities are linked to specific outcomes/assessment criteria of the outlined Unit Standards. | ✓ |  |  |
| To determine your competence level, the following are the methods to be used for this assessment:   1. ..................................................... 2. ..................................................... |  |  |  |
| To be declared competent on **Explain computer architecture concepts** module (formative assessment), you should have obtained at least 80% of the total mark of this assessment. | ✓ |  |  |
| You will be provided with detailed feedback on your performance of this assessment as follows:   1. Written Feedback 2. Verbal Feedback | ✓ |  |  |
| Should you be declared “not yet competent” on this assessment, you will be entitled for re-assessment opportunity/ies. | ✓ |  |  |
| You will be required to re-submit evidence (only for areas) you were declared not yet competent. A date for re-submission will be agreed with the assessor. | ✓ |  |  |
| You will be entitled to lodge an appeal should you not be satisfied with the assessment decision of your assessment. | ✓ |  |  |
| You will be required to provide the assessor feedback on assessment procedure – this is to assist in improving the assessment practices. | ✓ |  |  |
| Your results of assessment and portfolio of evidence information will not be provided to any person without your written consent. | ✓ |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learner’s Declaration** | | | |
| I Mila Ngewu herewith declare that I am ready for the assessment, that we have reviewed the assessment preparation and plan, I understand the assessment process and I am happy that the assessment will be conducted in a fair manner. | | | |
| **Learner Signature:** | **Date:** | **Facilitator Signature:** | **Date:** |
|  | **17 October 2023** |  | **18 October 2023** |

**Learning Unit1**

**Explain computer architecture concepts**

**UNIT STANDARD NUMBER :** 14917

**LEVEL ON THE NQF :** 4

**CREDITS :** 7

**FIELD :** Physical, Mathematical, Computer and Life Sciences

**SUB FIELD :** Construction Information Technology and Computer Sciences

|  |  |
| --- | --- |
| **PURPOSE:** | This unit standard is intended:  to provide a fundamental knowledge of the areas covered  for those working in, or entering the workplace in the area of Information Systems & Technology Management.  as additional knowledge for those wanting to understand the areas covered  People credited with this unit standard are able to:  Explain computer architecture elements  Explain the organisation of a computer  Describe the design constraints in the design of instruction sets for computers  The performance of all elements is to a standard that allows for further learning in this area. |
| **LEARNING ASSUMED TO BE IN PLACE:** | |
| The credit value of this unit is based on a person having the prior knowledge and skills to:  be able to apply the principles of Procedural Computer Programming | |

|  |
| --- |
| **SESSION 1.**  **Explain computer architecture elements.** |
| **Learning Outcomes** |
| * 1. The explanation identifies the functions of elements which make up computer architecture. * 2. The explanation outlines the functions of elements which make up computer architecture. * 3. The explanation distinguishes categories of each element and outlines their features. * 4. The explanation identifies examples of the application of architecture elements. |

|  |  |
| --- | --- |
| http://3.bp.blogspot.com/_0EodaYtqevU/TMun5XOj03I/AAAAAAAAAIU/lzrnWelQjgc/s1600/group-discussion.jpg | **Answer the following questions according to the instructions provided** |

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **1** | **Define computer architecture.** | **3** |

Computer architecture encompasses the design and integration of hardware, firmware, and software components to create a functional and efficient computing system. The hardware forms the physical foundation, firmware provides low-level control and instructions embedded in the hardware, and software consists of programs and instructions that users and applications interact with to accomplish tasks on the computer.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **2** | **Identify the functions of elements which make up computer architecture.** | **7** |

Hardware

Central Processing Unit (CPU):

Function: Executes instructions stored in memory, performs calculations, and manages data processing.

Memory (RAM):

Function: Stores data and instructions that are actively used by the CPU during operations.

Firmware:

BIOS/UEFI (Basic Input/Output System/Unified Extensible Firmware Interface):

Function: Initializes hardware during the boot process, loads the operating system, and provides a basic set of instructions for hardware control.

Device Firmware (e.g., in routers, printers):

Function: Controls specific hardware devices, providing low-level functionality and interfacing between hardware and software.

Software:

Operating System:

Function: Manages hardware resources, provides a user interface, and facilitates communication between software applications and hardware.

Application Software:

Function: Enables users to perform specific tasks, such as word processing, web browsing, and graphic design.

Utilities:

Function: Perform system maintenance tasks, manage resources, and optimize system performance.

In short this is how these 3 elements work, hardware provides the physical foundation and resources, firmware offers low-level control and initialization, and software delivers the high-level instructions and functionality that users interact with to accomplish various tasks on a computer system. These elements work together cohesively to create a functional and efficient computing environment.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **3** | **What is the idea of software simulaton? Discuss the advantages and disadvantages.** | **5** |

Software simulation is a technique used to model the behavior of a system or process through the execution of computer software. It involves creating a virtual representation of the real-world system or process to analyze its performance, behavior, or outcomes.

**Advantages of Software Simulation:**

Cost-Effective:

Advantage: Compared to real-world testing or physical prototypes, software simulation is often more cost-effective. It allows for multiple scenarios to be tested without the need for expensive equipment or resources.

Time Efficiency:

Advantage: Simulation enables the rapid testing of various scenarios in a relatively short amount of time. This can be especially beneficial in situations where real-time experimentation would be impractical or time-consuming.

**Disadvantages of Software Simulation:**

Simplification and Abstraction:

Disadvantage: Simulations often involve simplifications and abstractions of real-world systems. While this is necessary for computational efficiency, it can lead to a lack of accuracy in certain scenarios.

Modeling Challenges:

Disadvantage: Developing accurate models for simulation can be challenging. If the model does not accurately represent the real-world system, the simulation results may be unreliable.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **4** | **What types of binding occur at language definition, language implementation, at translation** **and execution?** | **6** |

key types of binding associated with each stage:

1. Language Definition:

Static Binding (or Early Binding): In static binding, the association between a name and its entity is determined at compile-time or during the language definition phase. This means that the binding is fixed before the program is executed. For example, in statically-typed languages, the data types of variables are often determined at compile-time.

1. Language Implementation:

Static Binding: The language implementation phase also involves static binding. The compiler or interpreter determines the bindings between names and entities before the program is executed. This includes decisions related to variable types, function signatures, and other static aspects of the program.

1. Translation (Compilation or Interpretation):

Static Binding: The translation process, whether compilation or interpretation, often involves static binding.

Dynamic Binding (or Late Binding): Some languages, especially those that are dynamically typed or support dynamic features, involve dynamic binding. In dynamic binding, the association between a name and its entity is determined at runtime.

1. Execution:

Dynamic Binding: During program execution, dynamic binding comes into play in some programming languages.

Run-time Binding: This term is often used interchangeably with dynamic binding. It refers to the process of associating names with entities during the actual execution of the program. In dynamically-typed languages, variables may take on different types at runtime, and function calls may be resolved dynamically.

|  |
| --- |
| **SESSION 2.**  **Explain the organisation of a computer.** |
| **Learning Outcomes** |
| * 1. The explanation identifies the purpose of computer components. * 2. The explanation outlines how components achieve their outcomes in terms of their relationships, and the structure of the computer. |

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **5** | **Briefly describe and explain the explanation the purpose of computer components.** | **15** |

Computer components are the physical parts or elements that make up a computer system. Each component has a specific function, and when combined, they work together to perform various tasks. Here is a brief description of some essential computer components and their purposes:

Central Processing Unit (CPU):

Purpose: Often referred to as the brain of the computer, the CPU executes instructions stored in memory. It performs calculations, controls data flow, and manages other hardware components.

Memory (RAM - Random Access Memory):

Purpose: RAM provides temporary storage for data and instructions that are actively being used or processed by the CPU. It allows for quick access to information, improving the computer's speed and responsiveness.

Storage Devices (Hard Drives, SSDs):

Purpose: Storage devices store data persistently, including the operating system, software applications, and user files. Hard drives and SSDs are common types of storage devices.

Motherboard:

Purpose: The motherboard is the main circuit board that connects and facilitates communication between various hardware components, including the CPU, memory, storage devices, and expansion cards.

Input Devices (Keyboard, Mouse, etc.):

Purpose: Input devices allow users to interact with the computer by providing a means to input data and commands. Keyboards and mice are common examples.

Output Devices (Monitor, Printer, etc.):

Purpose: Output devices display information to the user or produce physical output based on computer processes. Monitors and printers are examples of output devices.

Graphics Processing Unit (GPU):

Purpose: The GPU is specialized hardware designed to handle graphics and image processing tasks. It is crucial for rendering images and videos, especially in gaming and multimedia applications.

Power Supply Unit (PSU):

Purpose: The PSU provides electrical power to the computer components. It converts electrical power from an outlet into the appropriate voltages needed by the various components.

Networking Components (Network Interface Cards, Wi-Fi Adapters):

Purpose: Networking components enable the computer to connect to networks, including local area networks (LANs) or the internet. They facilitate communication and data transfer between computers.

Cooling Systems:

Purpose: Cooling systems, including fans and heat sinks, help dissipate heat generated by the CPU and other components. They prevent overheating and ensure the proper functioning of the computer.

BIOS/UEFI (Basic Input/Output System/Unified Extensible Firmware Interface):

Purpose: BIOS/UEFI initializes hardware during the boot process, loads the operating system, and provides a basic set of instructions for hardware control.

Operating System:

Purpose: The operating system (OS) manages hardware resources, provides a user interface, and facilitates communication between software applications and hardware.

These components work together in a coordinated manner, allowing computers to execute a wide range of tasks, from basic operations to complex computations, and to provide a platform for various applications and user interactions.

|  |
| --- |
| **SESSION 3.**  **Describe the design constraints in the design of instruction sets for computers.** |
| **Learning Outcomes** |
| * 1. The description identifies the constraints, and outlines the issues involved. * 2. The description outlines how the constraints have been accommodated, by using examples. |

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **6** | **Demonstrate understanding of Classification of instruction sets** | **5** |

A Complex Instruction Set Computer (CISC) incorporates numerous specialized instructions, many of which are infrequently employed in practical programs. On the other hand, a Reduced Instruction Set Computer (RISC) streamlines the processor by including only commonly used instructions, relegating less common operations to subroutines. Although theoretically, Minimal Instruction Set Computers and One Instruction Set Computers are significant types, they are not currently implemented in commercial processors. Another variation is Very Long Instruction Word (VLIW), in which the processor receives multiple encoded instructions retrieved in a single instruction word.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **7** | **Identify examples of operations common to many instruction sets** | **7** |

Arithmetic Operations:

Addition (ADD): Adds two numbers together.

Subtraction (SUB): Subtracts one number from another.

Multiplication (MUL): Multiplies two numbers.

Division (DIV): Divides one number by another.

Logical Operations:

AND Operation (AND): Performs a bitwise AND operation between two values.

OR Operation (OR): Performs a bitwise OR operation between two values.

NOT Operation (NOT): Inverts the bits of a value.

XOR Operation (XOR): Performs a bitwise exclusive OR operation between two values.

Comparison Operations:

Compare (CMP): Compares two values without altering them, setting flags based on the result.

Conditional Branch (BEQ, BNE, etc.): Branches based on the result of a previous comparison.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **8** | **Identify characteristics common in CISC hardware architectures** | **7** |

Rich Instruction Set:

CISC architectures have a large and diverse set of instructions that can perform complex operations in a single instruction. This includes arithmetic operations, logic operations, and more specialized instructions.

Variable-Length Instructions:

Instructions in CISC architectures can have variable lengths. Some instructions may be quite long and complex, encompassing multiple operations in a single instruction.

Memory Access Modes:

CISC architectures often support multiple addressing modes, providing flexibility in how operands are accessed from memory. Common addressing modes include direct, indirect, indexed, and register modes.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Questions Description** | **Mark** |
| **9** | **Describe the design constraints in the design of instruction sets for computers with reference to Instruction length** | **5** |

Instruction set design for computers involves various constraints to ensure efficient and practical operation. One crucial constraint is instruction length, which directly impacts factors like memory usage, execution speed, and decoding complexity. Shorter instructions can reduce memory requirements and improve execution speed but may limit the complexity and functionality of instructions. Conversely, longer instructions can provide more capabilities but may require more memory and lead to slower execution. Striking the right balance in instruction length is a key challenge in instruction set design, as it influences the overall performance and efficiency of a computer architecture.