RQF LEVEL 4

SWDDA401

SOFTWARE DEVELOMENT

Data Structure and Algorithm Fundamentals

TRAINER'S MANUAL

October, 2024

DATA STRUCTURE AND ALGORITM FUNDAMENTALS

KOICA Korea International Cooperation Agency

TQUM TVET Quality Management Project

2024

AUTHOR'S NOTE PAGE (COPYRIGHT)

Copies available from:

- *HQs: Rwanda TVET Board-RTB*
- *Web: www.rtb.gov.rw*
- **KIGALI-RWANDA**

Original published version: October 2024

ACKNOWLEDGEMENTS

The publisher would like to thank the following for their assistance in the elaboration of this training manual:

Rwanda TVET Board (RTB) extends its appreciation to all parties who contributed to the development of the trainer's and trainee's manuals for the TVET Certificate IV in Software Development, specifically for the module**" SWDDA401: Data Structure and Algorithm Fundamentals."**

We extend our gratitude to KOICA Rwanda for its contribution to the development of these training manuals and for its ongoing support of the TVET system in Rwanda

We extend our gratitude to the TQUM Project for its financial and technical support in the development of these training manuals.

We would also like to acknowledge the valuable contributions of all TVET trainers and industry practitioners in the development of this training manual.

The management of Rwanda TVET Board extends its appreciation to both its staff and the staff of the TQUM Project for their efforts in coordinating these activities.

This training manual was developed:

Under Rwanda TVET Board (RTB) guiding policies and directives

Under Financial and Technical support of

COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

PRODUCTION TEAM

Authoring and Review

ISHIMWE Gilbert

NIZEYIMANA Martin

Validation

KWIZERA Olivier
SEKABANZA Jean de la Paix

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARELIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

MANIRAKIZA Jean de Dieu

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe
SUA Lim
SAEM Lee
SOYEON Kim
WONYEONG Jeong
SHYAKA Emmanuel

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

ACRONYMS

FIFO: First in First Out

JS: JavaScript

KOICA: Korea International Cooperation Agency

LIFO: Last in First out

RTB: Rwanda TVET Board

SWD: Software Development

TQUM: TVET Quality Management Project

TVET: Technical and Vocational Education and Training

INTRODUCTION

This trainer's manual includes all the methodologies required to effectively deliver the module titled **"Data Structure and Algorithm Fundamentals".** Trainees enrolled in this module will engage in practical activities designed to develop and enhance their competencies.

The development of this training manual followed the Competency-Based Training and Assessment (CBT/A) approach, offering ample practical opportunities that mirror real-life situations.

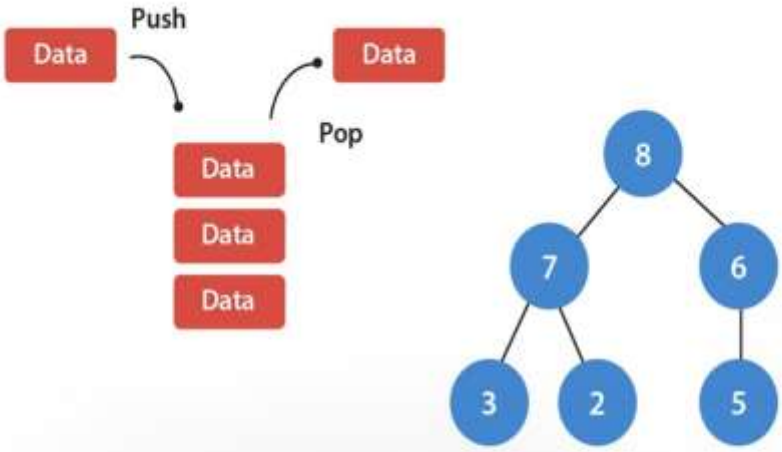
The trainer's manual is organized into Learning Outcomes, which is broken down into indicative content that includes both theoretical and practical activities. It provides detailed information on the key competencies required for each learning outcome, along with the objectives to be achieved.

As a trainer, you will begin by asking questions related to the activities to encourage critical thinking and guide trainees toward real-world applications in the labour market. The manual also outlines essential information such as learning hours, didactic materials, and suggested methodologies.

This manual outlines the procedures and methodologies for guiding trainees through various activities as detailed in their respective trainee manuals. The activities included in this training manual are designed to offer students opportunities for both individual and group work. Upon completing all activities, you will assist trainees in conducting a formative assessment known as the end learning outcome assessment. Ensure that trainees review the key reading and the points to remember section.

MODULE CODE AND TITLE: SWDDA401 DATA STRUCTURE AND ALGORITHM FUNDAMENTALS

Learning Outcome 1: Apply Algorithm Fundamentals
Learning Outcome 2: Apply Data Structure
Learning Outcome 3: Implement Algorithm using JavaScript


Push
Data
Data
Pop
Data
Data
Data
8
7
6
3
2
5

Indicative contents

1.1 Conversion of number systems

1.2 Description of logic gates and expressions

1.3 Use of data types on variables

1.4 Application of JavaScript operators

1.5 Write an algorithm

Key Competencies for Learning Outcome 1 : Apply Algorithm Fundamentals

Knowledge	Skills	Attitudes
• Description of number system • Description of logic gates and expressions • Description of the use of data types on variables	• Converting numbers from one numerical system to another • Applying JavaScript operators • Developing an algorithm • Drawing a flowchart	• Being critical thinker • Being innovative • Being attentive • Being creative • Being practical oriented • Being details Oriented • Having teamwork spirit

Duration: 30 hrs

Learning outcome 1 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe correctly number system based on number system types
2. Convert correctly Number systems according to the base conversion methods
3. Describe well Logic gates and expressions based on Boolean algebra
4. Apply effectively Data types according to their intended use
5. Apply appropriately Operators based on data type
6. Write Algorithm properly based on problem to be solved
7. Draw properly program flowchart based on best practice

Resources

Equipment	Tools	Materials
● Computer	● Visual paradigm ● Edraw max ● Lucid chart	● N/A

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- N/A

Indicative content 1.1: Conversion of Numbers System

Duration: 6 hrs

Theoretical Activity 1.1.1: Description of key concepts of number systems

Notes to the trainer:

- Trainer may use small groups for describing key concepts of number systems

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What is number system?

ii. Discuss about the following concepts of number systems:

a) Decimal base

b) Binary base

c) Hexadecimal base

d) Octal base

iii. Discuss application of number system

Step 2: Ask trainees to present their findings to the whole class

Step 3: Provide expert view and address any question if any.

Step 4: Ask trainees to read the key reading 1.1.1 in the trainee manual

Points to Remember

- A number system is a mathematical notation for expressing numbers of a given set, using digits or other symbols in a consistent manner.
- There are different types of number system, such as:
  - ✓ Decimal numbers
  - ✓ Binary numbers
  - ✓ Hexadecimal base
  - ✓ Octal base

Practical Activity 1.1.2: Converting a number from one numeral system to another

Notes to the trainer

- This activity may be done individually

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:

i. Convert 157 from decimal to binary number
ii. Convert 1101.1012 into decimal system.
iii. Convert 100 100 000 from base 2 to octal
iv. Convert 0110 1110 1101 from binary to hexadecimal
v. Convert hexadecimal number (3DE)16 to a binary number
vi. Convert an octal number (345)8 to a binary number
vii. Convert 62010 into hexadecimal

Step 2: Demonstrate to trainees how to convert a number from one numeral system to another and explain steps to follow while demonstrating

Step 3: Ask trainees to do the provided task and monitor

Step 4: Verify whether numbers conversion is properly performed

Step 5: Ask them to read the Key readings 1.1.2 in their manuals

Points to Remember

- **Conversions Between Number Bases**
  - ✓ **Binary to Decimal**: Summing each bit multiplied by 2 raised to its positional power.
  - ✓ **Decimal to Binary**: Repeatedly dividing by 2 and noting remainders.
  - ✓ **Hexadecimal to Binary**: Direct conversion since one hex digit maps to four binary digits.

- ✓ **Binary to Hexadecimal**: Grouping binary digits in sets of four and converting.

Application of learning 1.1.

RXtc astronomy is a branch of the Rwanda national institute of science. RXtc astronomy deals with celestial objects, space, and the physical universe as a whole. In 2022 the institute have sent spaceship called Ω-spaceship to the moon. And waited for the message that confirm the arrival of the spaceship and coordinates.
After one month the Ω spaceship have sent a series of binary codes which contain a message. You have been recruited to a team of scientists working on deciphering the message sent. Your task is to convert these binary codes into decimal base, binary base, Hexadecimal base and Octal decimal base to help the researchers analyze the message.

Notes: The Ω-spaceship provided the following list of binary numbers.
Binary Numbers:

1. 01001110
2. 11101.001
3. 00110111
4. 10101010
5. 011.11111
6. 10000000
7. 01010101
8. 11001100
9. 00011001

After conversion share your findings with the researchers' team, who will attempt to interpret the message based on the converted numbers.

SN	Criteria	Indicators	Observation	
			Yes	No
1.	Number systems are correctly converted	1.1 Numbers received		
		1.2 Conversion Base Chosen		
		1.3 Number converted		
2.	Result communication	2.1 Findings recoded		
		2.2 Findings Analysed		
		2.3 Findings communicated		

Indicative content 1.2: Description of Logic Gates and Expressions

Duration: 6 hrs

Theoretical Activity 1.2.1: Description of logic gates and expressions

Notes to the trainer:

- Trainer may use small groups for describing logic gates and expressions

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to provide answers on the following questions:

i. What is meant by logic gates?
ii. Describe the types of logic gates.
iii. Discuss about representation of Boolean logic gates.
iv. Discuss application of Boolean logic gates

Step 2: Ask trainees to present their findings in front of whole class

Step 3: Provide expert view on presented content

Step 4: Address any questions or concerns

Step 5: Ask them to read the Key readings 1.2.1 in their manuals

Points to Remember

- A Logic gate is an essential component of digital circuits. It takes one or more binary inputs and produces an output based on the logic rule it operates.
- There are 3 types of logic gates: **Basic Gates** : (OR, AND, and NOT Gates), **Universal Gates**:(NAND, and NOR Gates) and **Derived Gates**: (XOR Gates, and XNOR Gates).
- In computer science, the role of logic circuits is fundamental and profound. They form the core building blocks of digital systems and contribute to the functionality of every type of digital device, from **calculators** and **watches**, to **powerful computer processors** and **complex digital networks**.

Practical Activity 1.2.2: Applying Boolean logic gates

Notes to the trainer

- This activity may be done individually

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:

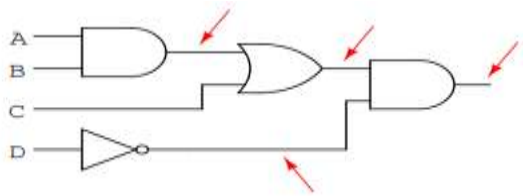
i. Using the above laws, simplify the following expression: (A + B)(A + C)

ii. Convert the following Boolean expression into logic circuit:

a) A(B + CD)

b) $(\overline{AB})B\overline{C}$

iii. Convert the following logic gate circuit into a Boolean expression, writing Boolean sub-expressions next to each gate output in the diagram:



iv. Demonstrate (show) by the truth table the following De Morgan theorems.

a) $\overline{a+b} = \overline{a}.\overline{b}$

b) $\overline{ab} = \overline{a} + \overline{b}$

Step 2: Demonstrate to trainees how to apply Boolean logic gates and explain steps to follow while demonstrating

Step 3: Ask trainees to do the provided task and monitor

Step 4: Verify whether numbers conversion is properly performed

Step 5: Ask them to read the Key readings 1.2.2 in their manuals

Points to Remember

- The procedure to detect output states using a truth table consists of the following steps:

1. Create a truth table with a number of input columns equal to the number of inputs in your logic circuit. Add one more column to denote the output.
2. The number of rows in the truth table would equal 2n, where n is the number of inputs.
   For example, if you have 2 inputs, you'll have 22=4 rows. Two zeros and two ones
3. Fill in the table with all possible combinations of inputs.
4. Based on the logic gate or combination of gates in your logic circuit, fill in the output for each row.

Application of learning 1.2.

A team of software engineers team needs to design an algorithm that decides when and how these functions should be activated based on multiple input conditions.

To optimize the decision-making process, the team decides to use logic gates and Boolean expressions to define the conditions under which different actions will be taken. By applying the fundamentals of logic gates, they can create a simplified model that efficiently controls the automation system. The engineers need to represent which logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) to use and how to combine them to represent complex logical expressions.

SN	Criteria	Indicators	Observation	
			Yes	No
1	Boolean logic gates are well represented	Logic gates Identified		
		Gates represented		
		Complex logical expressions represented		

Indicative content 1.3: Use of Data Types on Variables

Duration: 6 hrs.

Theoretical Activity 1.3.1: Description of data types in JavaScript

Notes to the trainer:

- Trainer may use small groups for describing data types in JavaScript

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Discuss on the following the following terms as used in JavaScript:

a) Variables

b) Data types

Step 2: Ask trainees to present their findings to the whole class.

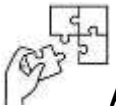
Step 3: Provide expert view on presented content

Step 4: Address any questions or concerns

Step 5: Ask them to read the Key readings 1.3.1 in their manuals

Points to Remember

- In JavaScript, Variable is a container for Storing Data. Also, we can say that variable is a memory zone used to store and manage data.
- Data type is an attribute associated with a data that tells a computer system how to interpret its value.
  Data types are primarily used in computer programming, in which variables are created to store data. Each variable is assigned a data type that determines what type of data the variable may contain.



Application of learning 1.3.

Movie Ticket Booking Form

The application based on where a user is interacting with a simple JavaScript-based webpage to fill out a form for a movie ticket booking. The form captures basic information like the user's name, age, contact number, whether they want snacks, and their preferred movie format. In this scenario, we can apply only JavaScript primitive types to handle the form data.

The webpage contains a form with the following fields:

1. **Name** (String)
2. **Age** (Number)
3. **Contact Number** (String)
4. **Snacks** (Boolean)
5. **Preferred Format** (String)
6. **Booking Reference ID** (Symbol)
7. **Total Cost** (Number)
8. **Coupon Code** (String, optional)
9. **Booking Status** (Null or String, initially null indicating no status)
10. **Customer email** (String)
11. **Address** (String)

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Java Script data type is well used	1.1. Variables are identified		
		1.2. Types of data types identified		
		1..3. Primitive data type used on a variable		

Indicative content 1.4: Application of JavaScript Operators

Duration: 6 hrs

Practical Activity 1.4.1: Applying JavaScript Operators

Notes to the trainer

- Trainer may use individualized facilitation technique for Applying JavaScript Operators

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:
Write a program that will prompts the user to enter two numbers and an operation (+, -, *, /) and then performs the calculation based on the input.

Step 2: Demonstrate to trainees how to Apply JavaScript Operators

Step 3: Ask trainees to do the provided task by applying the knowledge learn from trainer's demonstration

Step 4: Provide expert view on presented content

Step 5: Address any questions or concerns

Step 6: Ask them to read the Key readings 1.4.1 in their manuals

Points to Remember

- JavaScript operators are special symbols used in scripts to perform operations on operands, such as arithmetic calculations, logical comparisons, or value assignments. It plays a crucial role in controlling the flow and processing of data within the language.
- There are different types of JavaScript operators:

  ✓ Arithmetic Operators
  ✓ Assignment Operators
  ✓ Comparison Operators

- ✓ String Operators
- ✓ Logical Operators
- ✓ Bitwise Operators
- ✓ Ternary Operators

Application of learning 1.4.

An application based on Creating a simple calculator application that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on numbers. To do so represents numerical values, including integers and floating-point numbers, used to store the operands and results of calculations. Represents textual data, used to display the input and output values, as well as any error messages. Represents true or false values, used to control conditional logic (e.g., checking for invalid input or handling division by zero).

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	JavaScript operators are well Applied	Assignment operators are applied		
		Arithmetic operators are applied		
		Logical operators are applied		
		Relational operators are applied		
		Compound operators are applied		
		Conditional operators are applied		
		Bitwise operators are applied		

Indicative content 1.5: Write an Algorithm

Duration: 6 hrs

Theoretical Activity 1.5.1.1: Description of key concepts of Algorithm

Notes to the trainer:

- Trainer may use small groups to describe key concepts of Algorithm fundamentals

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

1. Define the following terms:
   a) Language
   b) Programming language
   c) Source code
   d) Machine code
   e) Algorithm
   f) IDE

iv. Differentiate compiler from interpreter

v. Discuss about types of algorithms

vi. Describe characteristics/qualities of a good algorithm

Step 2: Asks any Trainees to write answers provided on flipchart/paper.

Step 3: Asks trainees to discuss the provided answer and choose correct answers.

Step 4: Provides expert view and clarifies ideas by using didactic materials.

Step 5: Ask trainees to read the key reading 1.5.1 in the trainee manual

Points to Remember

- Language is a mode of communication that is used to share ideas, opinions with each other.
- Programming language is a formal language used by computer programmer to create software programs and instruct a computer
- Source code is the set of instructions written by a programmer using a computer programming language

- Machine code is a computer language that is directly understandable by a computer's CPU.
- Algorithm is a step-by-step procedure (well-defined instructions) to solve a given problem in order to get an expected result.
- An IDE is a set of tools like text editor, compiler, debugger etc., that all work together to facilitate software programming.
- Compiler is defined as a software that transforms an entire set of source code into object code and saves it as a file before executing it. Conversely, an interpreter converts and executes source code line by line without saving it and points out errors along the way.

Practical Activity 1.5.2: Developing an algorithm using structured English

Notes to the trainer

- Trainer may use individualized facilitation techniques for Developing an algorithm using structured English.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:
By using structured English Write an algorithm of the program that will help a user to calculate the sum and the average of three numbers.

Step 2: Demonstrate to trainees how to Develop an algorithm using structured English

Step 3: Ask trainees to do the provided task following what trainer demonstrated

Step 4: Provide expert view on presented content

Step 5: Address any questions or concerns

Step 6: Ask them to read the Key readings 1.5.2 in their manuals

Points to Remember

- Structured English is a method of writing down the steps of an algorithm using plain English but with added precision and structure.
- Rules for Writing Structured English includes: Use **simple, short sentences**. **No ambiguity**: Be as precise as possible. Include **conditional statements** and **iteration**

explicitly. Indent blocks for clarity, especially when dealing with loops or conditional logic. And Avoid language that could be misinterpreted or misunderstood.

Practical Activity 1.5.3: Developing an algorithm using pseudocode

Notes to the trainer

- Trainer may use individualized facilitation technique for Developing an algorithm using pseudocode

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:

By using pseudocode, write an algorithm of the program that will help a user to calculate the sum and the average of three numbers.

Step 2: Demonstrate to trainees how to Develop an algorithm using structured pseudocode

Step 3: Ask trainees to do the provided task following what trainer demonstrated

Step 4: Provide expert view on presented content

Step 5: Address any questions or concerns

Step 6: Ask them to read the Key readings 1.5.3 in their manuals

Points to Remember

- Developing an algorithm using structured pseudocode involves creating a representation of the algorithm that looks similar to actual programming code but is written in a simplified, language-agnostic format.
- An algorithm is made mainly of the following parts: The variable declaration line. The beginning of an algorithm. the instruction's part. And The end

Theoretical Activity 1.5.1.2: Description of program flowchart

Notes to the trainer:

- Trainer may use small groups to describe program flowchart

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. What is flowchart
ii. Talk about advantages of flowchart
iii. Describe elements of flowchart
iv. Tools used to draw flowchart
v. Discuss flowchart best practices

Step 2: Ask trainees to present their findings in front of the class.

Step 3: Provides expert view and clarifies ideas by using didactic materials.

Step 4: Ask trainees to read the key reading 1.5.1.2 in the trainee manual

Points to Remember

- Flowchart is diagrammatic /Graphical representation of sequence of steps to solve a problem.
- Flowchart has symbols such as:
  - ✓ Oval: Start and End points.
  - ✓ Rectangle: Process or action steps.
  - ✓ Diamond: Decision points.
  - ✓ Parallelogram: Input/Output operations.
  - ✓ Arrow Lines: Indicate the flow direction.

Practical Activity 1.5.4: Designing a flowchart

Notes to the trainer

- Trainer may use individualized facilitation techniques for designing a flowchart

 **Key steps:**

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to read the following task:
By using structured English, write an algorithm of the program that will help a user to calculate the sum and the average of three numbers.

Step 2: Demonstrate to trainees how to develop an algorithm using structured English

Step 3: Ask trainees to do the provided task following what trainer demonstrated

Step 4: Provide expert view on presented content

Step 5: Address any questions or concerns

Step 6: Ask them to read the Key readings 1.5.4 in their manuals

 **Points to Remember**

- Steps to draw a basic flow chart in draw.io:
    1. Create a new blank diagram
    2. Add shapes to the drawing canvas
    3. Move, resize, rotate, and delete shapes
    4. Connect shapes
        - Draw a floating connector
        - Draw a fixed connector
        - Change the path of a connector
    5. Add labels
    6. Style your flow chart
    7. Export and share your flow chart

 **Application of learning 1.5.**

Drawing a Flowchart for a Simple ATM Withdrawal Program

You have been asked to document the logic of a basic ATM (Automated Teller Machine) withdrawal program. The goal is to create a flowchart that visually represents how a user interacts with the ATM to withdraw money. The ATM has features for PIN authentication, balance checking, and processing a withdrawal.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	the logic of a basic ATM withdrawal program is well documented	Flow chart is created		
		PIN is Authenticated		
		Balance is checked		
		Withdrawal is processed		

Learning outcome 1 end assessment

Theoretical assessment

Section 1: Read the Read the following statement related to data structure and algorithm fundamentals, and choose the letter corresponding to the correct answer

1. **Convert the binary number 1101 to its decimal equivalent.**
   a) 11
   b) 12
   c) 13
   d) 14
   Answer: c) 13
2. **Convert the decimal number 29 to its binary equivalent.**
   a) 11101
   b) 10101
   c) 11011
   d) 10011
   Answer: a) 11101
3. **What is the hexadecimal equivalent of the binary number 10110110?**
   a) B6
   b) 76
   c) A6
   d) C6
   Answer: a) B6
4. **Convert the octal number 345 to its decimal equivalent.**
   a) 229
   b) 229
   c) 221
   d) 231
   Answer: d) 229
5. **What is the result of converting the hexadecimal number 2A to binary?**
   a) 101010
   b) 110010
   c) 100101
   d) 111000
   Answer: a) 101010

6. **Which logic gate has an output of 1 only when both of its inputs are 1?**
   a) OR
   b) AND
   c) NOT
   d) XOR
   Answer: b) AND
7. **The output of an OR gate is 1 if:**
   a) Both inputs are 0
   b) At least one input is 1
   c) Both inputs are 1
   d) Both inputs are different
   Answer: b) At least one input is 1
8. **If you want to build a circuit that outputs 1 when either input A is 1 or both inputs B and C are 1, which combination of gates would you use?**
   a) OR and NOT
   b) AND and OR
   c) AND and NAND
   d) XOR and NOR
   Answer: b) AND and OR
9. **Which of the following is a universal gate?**
   a) AND
   b) OR
   c) XOR
   d) NAND
   Answer: d) NAND
10. **What is the decimal equivalent of the hexadecimal number 7F?**
    a) 127
    b) 126
    c) 124
    d) 121
    Answer: a) 127
11. **Which logic gate can be used to invert a binary input?**
    a) AND
    b) OR
    c) NOT
    d) XOR
    Answer: c) NOT

Section 2: Read the following statement related to data structure and algorithm fundamentals, and answer by True if the statement is correct and False if the statement is wrong

1. In JavaScript, the keyword var is used to declare a variable.
   Answer: True
2. JavaScript variables are case-sensitive, meaning myVar and myvar are considered the same variable.
   Answer: False
3. A JavaScript variable name cannot start with a number.
   Answer: True
4. In JavaScript, a variable declared with const can be reassigned later in the code.
   Answer: False
5. JavaScript supports both let and const for block-level variable declarations.
   Answer: True
6. The + operator in JavaScript can be used for both addition and concatenation.
   Answer: True
7. The === operator checks for equality without type coercion in JavaScript.
   Answer: True
8. The assignment operator in JavaScript is represented by ==.
   Answer: False
9. In JavaScript, the && operator returns true only if both operands are true.
   Answer: True
10. JavaScript has a ternary operator represented by? That can be used for conditional expressions.
    Answer: True

Practical assessment

In an educational setting, teachers and administrative staff often need to calculate students' performance based on their marks in different subjects. This calculation involves determining the total marks obtained and the average marks to assess whether a student has passed or failed according to predefined criteria.

However, performing these calculations manually for a large number of students can be time-consuming and prone to errors. To streamline this process and minimize errors, developing a simple program or algorithm becomes essential. This scenario requires designing an algorithm using pseudocode and a flowchart to automate the calculation of total and average marks for a student based on their scores in five different subjects.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Number systems are well Converted	1.1. Number systems are converted from decimal base to other system and vice versa		
		1.2. Number systems are converted from hexadecimal decimal base to other system and vice versa		
		1.3. Number systems are converted from decimal base to other system and vice versa		
2	Logic gates and expressions are well applied	2.1. Boolean expressions are converted into logic circuit		
		2.2. Logic circuits are converted into Boolean expressions		
		2.3. Truth table applied		
3	Data types are well used on variables	3.1. Variables are identified		
		3.2. Data types are identified		
		3.3. Data types are used on variables		
4	JavaScript operators are well Applied	4.1. Assignment operators are applied		
		4.2. Arithmetic operators are applied		
		4.3. Logical operators are applied		
		4.4. Relational operators are applied		
		4.5. Compound operators are applied		
		4.6. Conditional operators are applied		
		4.7. Bitwise operators are applied		
5	Algorithm is well Written	5.1. Algorithm is developed using structured English		
		5.2. Algorithm is developed using structured pseudo code		
		5.3. Flowchart is Designed		

END

Further information to the trainer

BOOKS

Karumanchi, N. (2011 (2nd Edition)). *Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles.* Hyderabad, India: CareerMonk Publications.

Robert Sedgewick, K. W. (2011 (4th Edition)). *Algorithms.* Boston, MA, USA: Addison-Wesley.

Skiena, S. S. (2020 (3rd Edition)). *The Algorithm Design Manual.* New York, USA: Springer.

Thomas H. Cormen, C. E. (2009 (3rd Edition)). *Introduction to Algorithms.* Cambridge, MA, USA: The MIT Press.

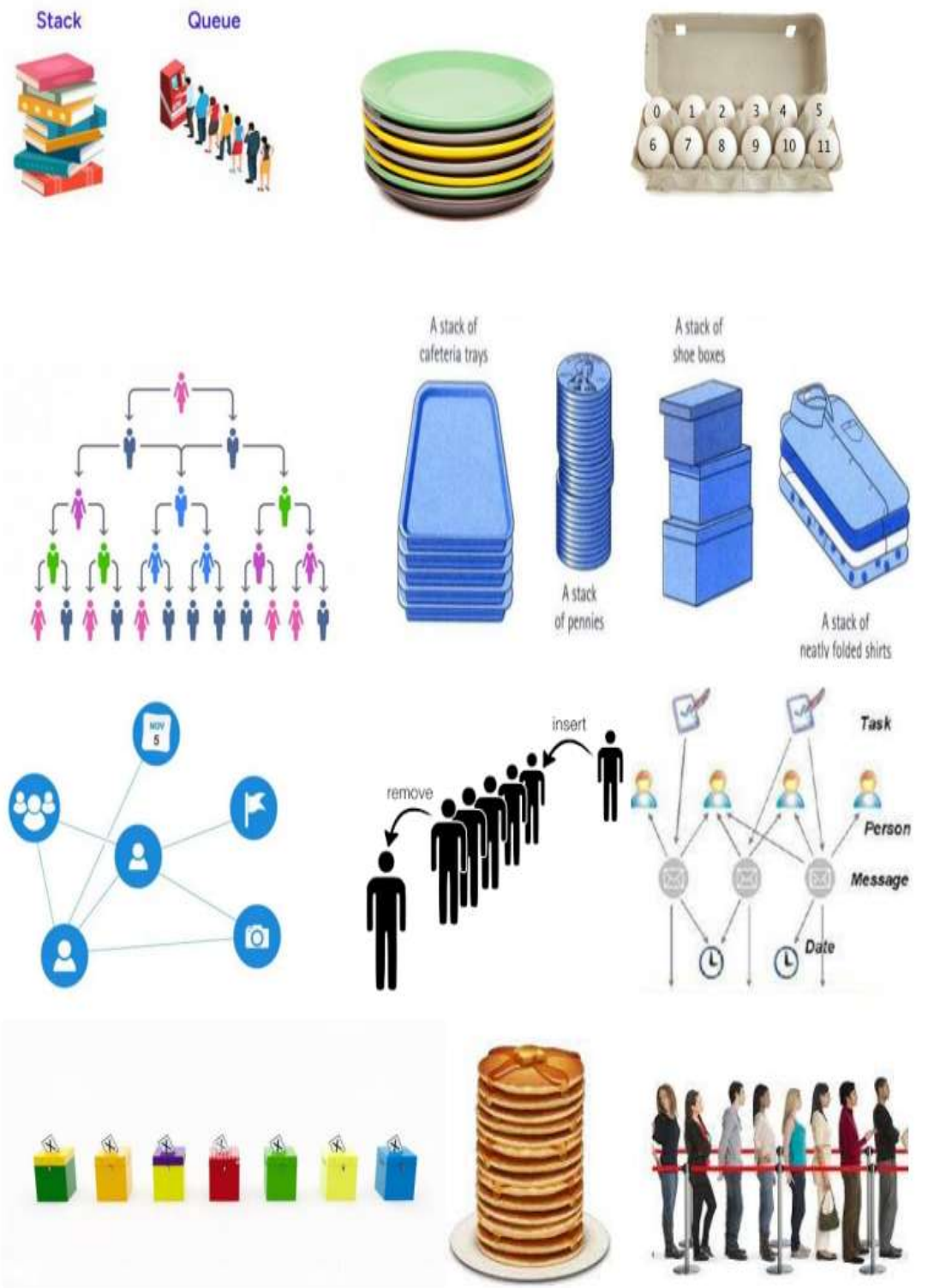
WEB LINKS

drawio. (2023). *doc/getting-started-basic-flow-chart*. Retrieved from drawio: https://www.drawio.com/doc/getting-started-basic-flow-chart

geeksforgeeks. (2023). *javascript-data-types*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/javascript-data-types/

tutorialspoint. (2024). *programming_methodologies/programming_methodologies_flowchart_elements.htm*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_flowchart_elements.htm

w3schools. (n.d.). */js/js_datatypes.asp*. Retrieved from w3schools: https://www.w3schools.com/js/js_datatypes.asp


Stack
Queue
0 1 2 3 4 5
6 7 8 9 10 11
A stack of cafeteria trays
A stack of pennies
A stack of shoe boxes
A stack of neatly folded shirts
NOV 5
insert
remove
Task
Person
Message
Date

Indicative contents
2.1 Identification of data structure concepts
2.2 Application of linear data structures and their operations
2.3 Application of non-linear data structure and their operations

Key Competencies for Learning Outcome 2: Apply Data Structure

Knowledge	Skills	Attitudes
● Description of data structure concepts. ● Description of algorithm design ● Classification of sorting algorithms ● Classification of Searching techniques	● Applying search technics ● Applying sorting techniques ● Applying linear data structure and their operations ● Applying non-linear data structure and their operations	● Being attentive. ● Having sprit of time management

Duration: 45 hrs.

Learning outcome 2 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Identify clearly Data structure concepts based on intended use.
2. Apply properly Linear Data Structures based on their operational complexity.
3. Apply properly Non-Linear Data Structures based on their operational complexity.

Resources

Equipment	Tools	Materials
● Computer	● Python tutor ● VisuAlgo	● N/A

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- N/A

Indicative content 2.1: Identification of Data Structure Concepts

Duration: 25 hrs.

Theoretical Activity 2.1.1.1: Description of data structure concepts

Notes to the trainer:

- Trainer may use small group for describing data structure concepts.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity by asking trainees to answer the following questions:

i. Explain deeply what is data structure?
ii. Describe Characteristics of Data structures?
iii. Discuss about types of Data structures?
iv. Explore classifications of data structures?
v. Discuss list representation
vi. Describe what structure is in programming.

Step 2: Ask trainees to present their findings to the whole class.

Step 3: Address any questions or concerns

Step 4: Ask them to read the Key readings 2.1.1.1 in their manual

Points to Remember

- **Data structures:** A data structure is a specialized format for organizing, processing, retrieving and storing data.
- Classifications of data structures data structures are classified into two categories: **Linear** and **Non-linear**
- **List**: is a sequence of data, types of list we have :**array based list** and **linked list**
- **List operations** refer to the various actions you can perform on lists to manage and manipulate their elements. List operation include **insertion, deletion, traversal, searching, Sorting, update.**
- **Structure** (often referred to as a **struct**) is a composite data type that groups together variables (often called members or fields) under a single name.

Theoretical Activity 2.1.1.2: Description of data structure algorithm

Notes to the trainer:

- Trainer may use small group for describing data structure concepts
- Avail physical objects, images, videos as didactic materials is required.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity by asking trainees to answer the following questions:

i. Explain deeply searching techniques
ii. Describe sorting techniques
iii. Discuss about classification of sorting algorithms
iv. Describe two types of algorithm complexity

Step 2: Ask trainees to discuss on provided questions in their small groups.
Step 3: Ask trainees to present their findings to the whole class.
Step 4: Provide expert view to the presented content.
Step 5: Address any questions or concerns
Step 6: Ask them to read the Key readings 2.1.1.2 in their manuals

Points to Remember

- There are two main types of algorithms include: **Searching Algorithm** and **Sorting Algorithm**
- Searching techniques are methods used to find specific elements within a data structure, such as an array or list. Includes: **Linear Search** and **Binary Search.**
- Sorting techniques are methods used to arrange elements in a list or array into a specified order includes **Selection Sort, bubble sort, Insertion Sort** and **Quick Sort.**
- The performance of the algorithm can be measured in two factors such as **Time complexity** and **Space complexity** here the **Time complexity an algorithm** is the amount of time required to complete the execution and **Space complexity** is the amount of space required to solve a problem and produce an output.

Practical Activity 2.1.2: Selecting appropriate data structure algorithm

Notes to the trainer

- Trainer may use individual based facilitation technique for Selecting appropriate data structure algorithm.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Ask trainees to perform the task described below:

In a bank, a Teller has role of saving the clients. He/she has organize the banknotes of same values together and coins of same values together that he/she has serve to clients. When a client arrives in the bank, he/she chooses a line to join in the front of a Teller and the line is formed according to the arrival of clients.

Mr.one wants to pay 28450rwf as school fees to the bank and he must organize banknotes and coins before deposit money to facilitate the teller to not mix the banknotes and coins.

1. Observe the above figure and describe what you see
2. Where a new arrived client will line up
3. Using above figure who is going to be served first by teller 1or teller 2
4. Discuss how one is going to organize banknotes and coins before depositing.
5. Discuss how the teller is going to keep money deposited by one

Step 2: Demonstrate to trainees how to select an appropriate algorithm to an existing problem. While demonstrating explain the procedures of selecting an appropriate algorithm

Step 3: Ask trainees to perform a given task referring to what they learnt from step 2.

Step 4: Monitor the activity and provide assistance where is needed.
Step 5: Ask trainees to present their task's results to trainer.
Step 6: Provide feedback and repeat until perfect results achieved
Step 7: Ask trainees to read key readings 2.1.2 in their trainee manual.
Step 8: Ask trainees to perform application of learning 2.1

Points to Remember

- **Merge sort is** a sorting algorithm that follows the divide-and-conquer approach.
- **Quicksort** is a sorting algorithm based on the Divide and Conquer algorithm that picks an element as a pivot and partitions the given array around the picked pivot by placing the pivot in its correct position in the sorted array**.**
- **Heap Sort** is a comparison-based sorting algorithm that uses a binary heap data structure.
- **Radix Sort** is a non-comparison-based sorting algorithm that sorts numbers by processing individual digits.
- **Counting Sort** is a non-comparison-based sorting algorithm suitable for sorting integers within a specific range.
- **Bucket Sort** is a non-comparison-based sorting algorithm that distributes elements into several buckets and then sorts each bucket individually (using a different sorting algorithm, like Insertion Sort).
- **Shell sort** is a generalized version of the insertion sort algorithm. It first sorts elements that are far apart from each other and successively reduces the interval between the elements to be sorted.

Application of learning 2.1.

In a market inventory system, you need to arrange a list of products by their prices in ascending order. Describe how you would apply the bubble sort technique to accomplish this. What would be the steps involved?

Solution:

Bubble sort is a simple comparison-based sorting technique where each pair of adjacent elements is compared, and they are swapped if they are in the wrong order.

Steps Involved

- Start at the beginning of the list.
- Compare the first two products' prices.
- If the first product is more expensive than the second, swap them.
- Move to the next pair and repeat the comparison and swapping process.
- Continue this process until the end of the list.
- Repeat the entire process for all elements, ignoring the last sorted element in each pass.

Finally: The list will be sorted in ascending order by price after several passes.

Indicative content 2.2: Application of Linear Data Structures and their Operations

Duration: 8 hrs

Theoretical Activity 2.2.1: Description of operations of linear data structure

Notes to the trainer:

- Trainer may use individual based facilitation technique for selecting appropriate a linear data structure.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity by asking trainees to answer the following questions:

i. Define a linear data structure.
ii. Identify Data Structure Operations
iii. Explain the application of linear data structure and examples.

Step 2: Ask trainees to present their findings to the whole class.

Step 3: Address any questions or concerns

Step 4: Ask them to read the Key readings 2.2.1 in their manual

Points to Remember

- **A linear data structure** is a type of data structure where elements are arranged sequentially or linearly, one after another.
- The following are some of the most frequently used operations: **Traversing, Searching, Inserting, Deleting, Sorting and Merging**
- Linear Data Structures are simpler and involve direct, sequential relationships **(example: Arrays, Linked Lists, Stacks, and Queues).**

Practical Activity 2.2.2: Selecting appropriate linear data structures

Notes to the trainer

- Trainer may use individual based facilitation technique for selecting appropriate a linear data structure.

Key steps:

While delivering this activity, pass through the following steps:

Step 1:Ask trainees to perform the task described below:

A market analyst is reviewing the last few transactions to identify the most recent ones. Which linear data structure would you use to review these transactions in reverse order, and why?

Step 2: Demonstrate to trainees how to select an appropriate algorithm to an existing problem. While demonstrating explain the procedures of selecting an appropriate algorithm

Step 3: Ask trainees to perform a given task referring to what they learnt from step 2.
Step 4: Monitor the activity and provide assistance where is needed.
Step 5: Ask trainees to present their task's results to trainer.
Step 6: Provide feedback and repeat until perfect results achieved
Step 7: Ask trainees to read key readings 2.1.2 in their trainee manual.
Step 8: Ask trainees to perform application of learning 2.1

Points to Remember

- **Arrays:** An array is a collection of elements identified by index or key, where each element is of the same data type and stored at contiguous memory locations.
- **Linked List:** A linked list is a linear data structure where each element (node) contains a data part and a reference (or link) to the next node in the sequence.
- **Stack:** A stack is a linear data structure that follows the Last In, First Out (LIFO) principle.

- **Queue:** A queue is a linear data structure that follows the First In, First Out (FIFO) principle.
- **A stack overflow** occurs when you try to push an item onto a stack that is already full
- **A stack underflow** occurs when you try to pop an item from an empty stack.

Application of learning 2.2.

A grocery store implements a system where the last product scanned at checkout can be removed if the customer changes their mind. Which linear data structure would be best for this functionality, and how would it operate?

Solution:

To handle the removal of the last scanned product at checkout, a **stack** is the best linear data structure.

How It Works:

- Each product scanned is pushed onto the stack.
- If the customer decides to remove the last scanned product, it is simply popped off the stack.

Finally: This setup ensures that the last scanned product can be removed efficiently, reflecting the real-time decision-making process at the checkout.

Indicative content 2.3: Application of Non-Linear Data Structures and their Operations

Duration: 7 hrs

Theoretical Activity 2.2.1: Description of application for non- linear data structure

Notes to the trainer:

- Trainer may use individual-based facilitation technique to select a linear data structure appropriate.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity by asking trainees to answer the following questions:
  i. Define a non-linear data structure.
  ii. Discus to examples of non-linear data structure

Step 2: Ask trainees to present their findings to the whole class.

Step 3: Address any questions or concerns

Step 4: Ask them to read the Key readings 2.3.1 in their manual

Points to Remember

- **A non-linear data structure** is a type of data structure where data elements are not arranged in a sequential or linear order**.**
- **Examples of non-linear data structures are** Trees**,** Graphs, Tables

Practical Activity 2.3.2: Applying appropriate linear data structures

Notes to the trainer

- Trainer may use individual based facilitation technique for applying appropriate a non-linear data structure.

 **Key steps:**

While delivering this activity, pass through the following steps:

Step 1: **Ask trainees to perform the task described below:**

You want to ensure that all market locations in a city are connected so that goods can be transferred between any two locations, either directly or indirectly. Describe how you would check if your graph is fully connected.

Step 2: Demonstrate to trainees how to select an appropriate non-linear data structure to solve problem.

Step 3: Ask trainees to perform a given task referring to what they learnt from step 2.

Step 4: Monitor the activity and provide assistance where is needed.

Step 5: Ask trainees to present their task's results to trainer.

Step 6: Provide feedback and repeat until perfect results achieved

Step 7: Ask trainees to read key readings 2.3.2 in their trainee manual.

Step 8: Ask trainees to perform application of learning 2.3

 **Points to Remember**

- **Depth-First Search (DFS)** is a fundamental algorithm used in graph theory to traverse or search through a graph or tree structure.
- **Graph Data Structure is** a collection of nodes connected by edges.
- Types of graphs in data structure we have: **Finite Graph Infinite Graph, Trivial Graph, Simple Graph, Multi Graph**

 **Application of learning 2.3.**

You are designing a market distribution network where each main warehouse supplies multiple regional warehouses, and each regional warehouse, in turn, supplies several local stores. However, there are also direct routes between some local stores for emergency supplies. How would you model this system using a combination of graph and tree structures?

Solution:

This scenario involves a hierarchical structure (tree) combined with additional connections (graph) to handle special cases like emergency routes.

- **Tree Structure:**

- Model the hierarchical distribution network using a tree.
- The main warehouse is the root node.
- Regional warehouses are child nodes of the main warehouse.
- Local stores are child nodes of the respective regional warehouses.
- **Graph Structure:**
- Add direct routes (edges) between some local stores that need emergency supplies, which do not follow the hierarchical structure.
- These edges create a graph structure, allowing for flexible routes outside the strict tree hierarchy.

Learning outcome 2 end assessment

Theoretical assessment

I. Read the Read the following statement related to data structure and algorithm fundamentals, and choose the letter corresponding to the correct answer

1. Which of the following is an example of a non-linear data structure?
   a) Linked List
   b) Queue
   c) Stack
   d) Tree

   Answer: d) Tree

2. Which sorting algorithm does NOT use recursion?
   a) Quick Sort
   b) Merge Sort
   c) Insertion Sort
   d) Heap Sort

   Answer: c) Insertion Sort

3. Which of the following operations is common to both linear and binary search?
   a) Sorting the data before searching
   b) Comparing elements
   c) Splitting the data into two halves
   d) Reversing the data
   Answer: b) Comparing elements
4. What is the best sorting algorithm to use when stability is a requirement?
   a) Quick Sort
   b) Merge Sort
   c) Heap Sort
   d) Shell Sort
   Answer: b) Merge Sort
5. Which classification of sorting algorithms involves comparing elements to one another?
   a) By Recursion
   b) By Memory Usage
   c) By Number of Comparisons
   d) By Stability
   Answer: c) By Number of Comparisons

II. Read the following statement related to data structure and algorithm fundamentals, and answer True if the statement is correct and False if the statement is wrong:

1) A binary search can be used on unsorted data

   Answer: False: Binary search requires the data to be sorted in order to function properly

2) Merge Sort is both a stable and recursive sorting algorithm

Answer: True: Merge Sort is stable (preserves the order of equal elements) and uses recursion

3) Arrays and linked lists are examples of non-linear data structures.

   Answer: False: Arrays and linked lists are linear data structures, not non-linear

4) The number of swaps is a classification criterion for sorting algorithms.

Answer: True: Sorting algorithms can be classified by the number of swaps they perform, affecting efficiency

5) Time complexity measures the amount of memory an algorithm uses during execution

Answer: False: Time complexity measures the amount of time an algorithm takes to run, while space complexity measures memory usage.

Practical assessment

Hasha Ltd is facing challenges in managing its warehouse data due to the use of hard copies for record-keeping. As a solution, you have been tasked with creating an efficient data management system that handles imports, exports, and transit processes for furniture. Use the following data structures: **Stack, Sorting, and Linked List** to develop the system.

For each question below, **write an algorithm** that addresses the respective problem:

1. The company receives shipments daily, and the manager needs to process them in the order they arrive. Sometimes the manager needs to undo the last action (removal of incorrect shipments). Write an algorithm using a **Stack** to manage this process. The stack should allow the manager to undo the last operation efficiently. Define the methods needed to push, pop, and check the top of the stack.

2. The warehouse manager needs to sort furniture items by categories such as Tables, Chairs, Beds, and Dressers to produce a daily report. Write an algorithm using a Sorting Algorithm to organize these categories. Explain which sorting algorithm (e.g., Quick Sort, Merge Sort, Bubble Sort, etc.) you would choose and why it is appropriate for this situation.

3. The number of shipments in the warehouse can vary daily, so a flexible data structure is needed. Write an algorithm using a Linked List to represent the furniture items in the warehouse, allowing dynamic addition and removal of items. Explain how the linked list supports these operations and how you would traverse the list to generate reports

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Linear data structures and their operations are well applied	1.1. Linked lists data structures are applied		
		1.2. Array data structures are applied		
		1.3. Queue data structures are applied		
		1.4. Stack data structures are applied		
2	Non-linear data structure and their operations are applied	2.1. Tree data structures are applied		
		2.2 Graph data structures are applied		
		2.3 Table data structures are applied		

END

Further information to the trainer

BOOKS

Algorithms, L. J. (2016). *Loiane Groner.* Birmingham, UK: Packt Publishing.

Bae, S. (2017). *Data Structures and Algorithms with JavaScrip.* UK: Packt Publishing.

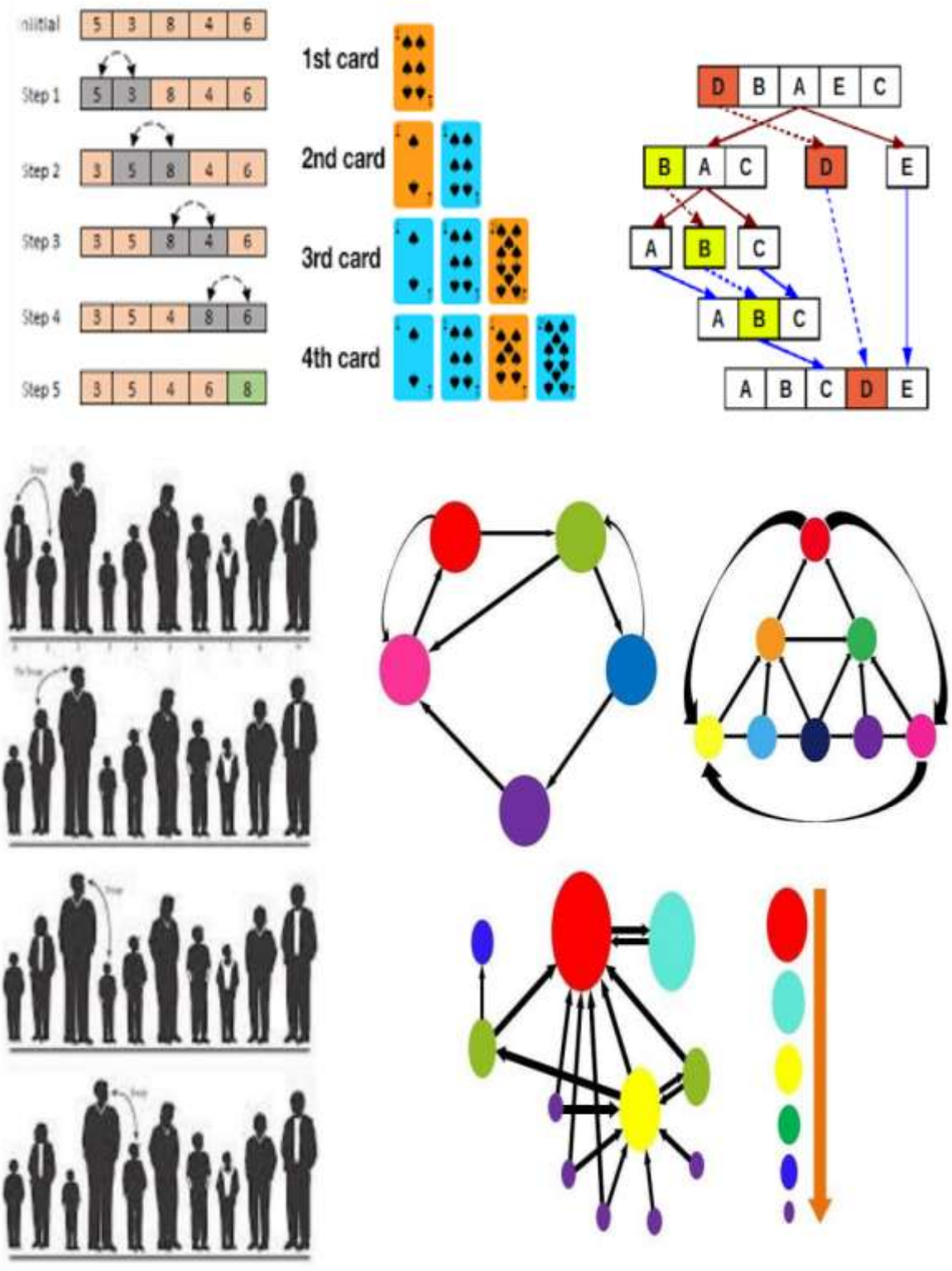
Rozentals, N. (2016). *Mastering JavaScript Data Structures and Algorithms.* Birmingham, UK: Packt Publishing.

WEB LINKS

programiz. (2024). */dsa/data-structure-types*. Retrieved from programiz: https://www.programiz.com/dsa/data-structure-types

simplilearn. (2024). */tutorials/data-structure-tutorial/what-is-data-structure#:~:text=Data%20structures%20are%20a%20specific,the%20data%20for%20easy%20use.* Retrieved from simplilearn: https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-data-structure#:~:text=Data%20structures%20are%20a%20specific,the%20data%20for%20easy%20use.

techtarget. (2024). */searchdatamanagement/definition/data-structure*. Retrieved from techtarget: https://www.techtarget.com/searchdatamanagement/definition/data-structure.


Initial
Step 1
Step 2
Step 3
Step 4
Step 5
1st card
2nd card
3rd card
4th card
D B A E C
B A C D E
A B C
A B C
A B C D E

Indicative contents
1.1 Development of JavaScript source code **1.2 Run JavaScript source codes** **1.3 Test Time and space complexity**

Key Competencies for Learning Outcome 3: Implement Algorithm using JavaScript

Knowledge	Skills	Attitudes
• Description of JavaScript running environment • Description of Key concepts of measuring time and space complexity • Description of Time and space measurement tools	• Preparing JavaScript running environment • Writing JavaScript source code • Appy sorting operations • Apply search operations • Using browser-embedded Tools • Using IDE Terminal • Documenting test findings	• Being critical thinker • Being innovative • Being attentive • Being creative • Being practical oriented • Being details Oriented • Having a teamwork spirit

Duration: 55 hrs

Learning outcome 3 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe correctly JavaScript running environment in line of its type
2. Prepare properly JavaScript running environment in accordance with computer operating system
3. Write properly JavaScript Source code based on Algorithm
4. Run successfully JavaScript source code in accordance with expected result
5. Describe properly Key concepts of measuring time and space complexity based on properties of developed program
6. Test successfully Time and space complexity based on data structure standards

 **Resources**

Equipment	Tools	Materials
● Computer	● Benchmarkify.js ● jsPerf ● WebStorm ● Visual Studio Code IDE built-in profiling tool ● Frame graphs ● Blackfire ● YSlow ● Chrome/Microsoft/Firefox DevTools	● Internet

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- N/A

Indicative content 3.1: Development of JavaScript Source Code

Duration: 20 hrs

Practical Activity 3.1.1: Preparing JavaScript running environment

Notes to the trainer

- Trainer may avail Chrome/Microsoft/Firefox DevTools for JavaScript execution
- Avail WebStorm and Visual Studio Code as the IDEs
- Avail flash discs to be used during installation

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:
To get started with JavaScript programming, especially for implementing data structures like Linked Lists, Arrays, Queues, Stacks, Trees, Graphs, and Tables, you need to set up a development environment. Go in computer LAB and prepare environment to run JavaScript programs.

Step 2: Demonstrate how to set up a JavaScript development environment step by step.

Step 3: Ask trainees to individually perform the task 3.1.1 referring to the demonstration

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.1 in trainee's manual

Points to Remember

- The **JavaScript running environment** refers to the platform or environment where JavaScript code is executed.
- There are two main types of environments where JavaScript can run: Browser Environment and Node.js Environment (Server-side).

Practical Activity 3.1.2: Implementing LinkedList Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1:	Introduce the activity and read the task described below: As software developer implement a function to add a new node at the beginning, middle, and end of a linked list.
Step 2:	Demonstrate the activities though the steps in order to perform given task
Step 3:	Ask trainees to individually perform the task 3.1.2 referring to the demonstration
Step 4:	Monitor the activity by giving the assistance where it is needed.
Step 5:	Ask the trainees to read the key readings 3.1.2 in trainee's manual

Points to Remember

- There are multiple methods to add data to a linked list depending on where the new node has to be inserted, and are as follows –
  - ✓ Adding Node to the Linked List
    - Adding node to the beginning of the linked list
    - Adding node at a specific position
    - Adding node to the end of the linked list
  - ✓ Removing Node
    - Removing a specific node
    - Removing a node at a Specific Position

Practical Activity 3.1.3: Implementing Array Data structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:
Given an array containing numbers 1 through n but with one missing number, find the missing number.

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.3 referring to the demonstration

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.3

Points to Remember

- The JavaScript **Array** object lets you store multiple values in a single variable. An array is used to store a sequential collection of multiple elements of same or different data types.

Practical Activity 3.1.4: Implementing Queue Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:

Create a queue class with enqueue (), dequeue (), and peek() methods using an array.

enqueue('A') adds an element, and dequeue ()

i. Removes the front element

ii. Write a function isEmpty() to check if the queue is empty or not, and demonstrate its use.

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.4 referring to the demonstration

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.4

Points to Remember

- To implement a queue data structure, we need the following methods:
  - **enqueue:** To add elements at the end of the queue.
  - **dequeue:** To remove an element from the front of the queue.
  - **peek:** To get the front element without removing it.
  - **isEmpty:** To check whether an element is present in the queue or not.
  - **printQueue:** To print the elements present in the queue.

Practical Activity 3.1.5: Implementing Stack Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:
Create a stack class with methods for push (), pop (), and peek().

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.5 referring to the demonstration

Step 4: Monitor the activity by Giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.5

Points to Remember

- Though Arrays in JavaScript provide all the functionality of a Stack, let us implement our own Stack class. Our class will have the following functions.
  - push(element) − Function to push elements on top of the stack.
  - pop() − Function that removes an element from the top and returns it.
  - peek() − Returns the element on top of the stack.
  - isFull() − Checks if we reached the element limit on the stack.
  - isEmpty() − checks if the stack is empty.
  - clear() − Remove all elements.
  - display() − display all contents of the array

Practical Activity 3.1.6: Implementing Tree Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:
Write a function to calculate the height of a binary

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.6 referring to the demonstration

Step 4: Monitor the activity by Giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.6

Points to Remember

- Tree data structure is a hierarchical structure that is used to represent and organize data in a way that is easy to navigate and search.

Practical Activity 3.1.7: Implementing Graph Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:

Implement a graph using an adjacency list and an adjacency matrix in JavaScript.

Write code to add vertices and edges and print both the adjacency list and matrix

Step 2: Demonstrate the activities though the steps in order to perform given task
Step 3: Ask trainees to individually perform the task 3.1.7 referring to the demonstration
Step 4: Monitor the activity by giving the assistance where it is needed.
Step 5: Ask the trainees to read the key readings 3.1.7

Points to Remember

- A graph consists of nodes (vertices) connected by edges. It can be represented in various ways, but a common one is using an Adjacency List.

Practical Activity 3.1.8: Implementing Table Data Structure

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:
Write a function that checks for duplicates in an array using a hash table
Test with different arrays to find duplicate.

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.8 referring to the demonstration

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.8

Points to Remember

- A hash table uses a hash function to map keys to values. In JavaScript, a native object can be used as a simple hash table.

Practical Activity 3.1.9: Performing sort operations

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:

A store manager wants to analyse the sales data of different products in their shop. The data is stored as an array of objects, where each object contains the name of a product and the total sales (in USD) for that product. Write a JavaScript function that takes this array as input and performs the following tasks:

a) Sort the products in descending order of sales (highest sales first).

b) Sort the products alphabetically by name if two or more products have the same sales. Return the sorted array.

Step 2: Demonstrate the activities though the steps in order to perform given task

Step 3: Ask trainees to individually perform the task 3.1.9 referring to the demonstration

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.1.9

Points to Remember

- **Quick sort**: it is one of the sorting algorithms that works on the idea of divide and conquer
- **Bubble Sort:** it is an algorithm that sorts an array from the lowest value to the highest value

Practical Activity 3.1.10: Performing sort operations

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

 **Key steps:**

While delivering this activity, pass through the following steps:

Step 6: Introduce the activity and read the task described below:

A store manager wants to analyse the sales data of different products in their shop. The data is stored as an array of objects, where each object contains the name of a product and the total sales (in USD) for that product.

Write a JavaScript function that takes this array as input and performs the following tasks:

a) Write a function to search for a product by its name using binary search for faster lookup.
b) If the product is found, return its price. If the product is not found, return "Product not available.

Step 7: Demonstrate the activities though the steps in order to perform given task

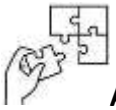
Step 8: Ask trainees to individually perform the task 3.1.10 referring to the demonstration

Step 9: Monitor the activity by giving the assistance where it is needed.

Step 10: Ask the trainees to read the key readings 3.1.10

 **Points to Remember**

- **Linear Search:** Checks each element one by one.
- **Binary Search:** Uses a divide-and-conquer approach to reduce the search space.

 **Application of learning 3.1.**

Hasha Ltd is facing challenges in managing its warehouse data due to the use of hard copies for record-keeping. As a solution, you have been tasked with creating an efficient data management system that handles imports, exports, and transit processes for furniture. Use the following data structures: **Stack, Sorting, and Linked List** to develop the system.

For each question below, **Write JavaScript code** that addresses the respective problem:

1. The company receives shipments daily, and the manager needs to process them in the order they arrive. Sometimes the manager needs to undo the last action (removal of incorrect shipments). Write JavaScript code using a **Stack** to manage this process. The stack should allow the manager to undo the last operation efficiently. Define the methods needed to push, pop, and check the top of the stack.

2. The warehouse manager needs to sort furniture items by categories such as Tables, Chairs, Beds, and Dressers to produce a daily report. Write JavaScript code using a Sorting Algorithm to organize these categories.

3. The number of shipments in the warehouse can vary daily, so a flexible data structure is needed. Write JavaScript code using a Linked List to represent the furniture items in the warehouse, allowing dynamic addition and removal of items.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Linear data structures and their operations are well applied	1.1. Linked lists data structures are applied		
		1.2. Stack data structures are applied		
		1.3. Stack operations are defined		
2	Warehouse data are well managed	2.1 Manager is able to undo the last operations		
		2.2 Manager is able to sort items		
		2.3 item is well added		
		2.4 item is well removed		

Indicative content 3.2: Run JavaScript Source Codes

Duration: 20 hrs

Theoretical Activity 3.2.1: Description of browser embedded Tools and IDE Terminal

Notes to the trainer:

- Avail computer LAB with JavaScript IDE installed

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Discuss browser-embedded tools, particularly focusing on the rendering engine and web development tools.

ii. Define an IDE and List top 5 JavaScript IDEs

Step 2: Ask trainees to present their findings in front of Class

Step 3: Provide expert view to presented content and address any question or concern

Step 4: Ask trainees to read key readings 3.2.1 in trainee's Manual

Points to Remember

- **Browser embedded tools** are features or functions integrated directly into a web browser, providing additional capabilities beyond the basic browsing experience.
- The main component of browser embedded Tools include Rendering engine and Web dev tools

Application of learning 3.2.

As software developer install a browser and JavaScript IDE terminal into your computer and write JavaScript codes that display Hello Word, using browser embedded Tools.

Solution:

1. **Install a Browser:** Download and install **Google Chrome** (or any modern browser like Firefox or Edge) from its official website.

2. **Open Developer Tools Console:** Launch the browser and press one of the following shortcuts to open Developer Tools:
    - F12
    - Ctrl + Shift + I (Windows/Linux)
    - Cmd + Option + I (Mac)
3. **Access the Console Tab**

- Click on the **Console** tab in Developer Tools.
- This is where you can type and run JavaScript code.

4. **Write and Run the JavaScript Code**

- Type the following code in the console:

```
<Script>
console.log ("Hello World");
</script>
```

- Press Enter.

5. **View the Output**

Indicative content 3.3: Test Time and Space Complexity

Duration: 15 hrs

Theoretical Activity 3.3.1: Description of Key concepts of measuring time and space complexity

Notes to the trainer:

- N/A

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Differentiate algorithm time complexity from space complexity?
ii. Describe the following algorithm measurement notation
    a. O(n log n)
    b. O(n)
iii. Describe time and space measurement tools

Ask trainees to present their findings in front of Class

Provide expert view to presented content and address any question or concern

Ask trainees to read key readings 3.2.1 in trainee's Manual

Points to Remember

- Time and space measurement tools include Profiling tools, Benchmark.js, Benchmark if y and jsPerf.

Practical Activity 3.3.2: Testing the program performance

Notes to the trainer

- Avail computer LAB with JavaScript IDE installed

 **Key steps:**

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and read the task described below:

Develop a simple web page and Test its Time and space complexity, and finally document test findings.

Step 2: Demonstrate the activities through the steps in order to perform given task

Step 3: Ask trainees to individually perform the provided task of measuring program time and space complexity.

Step 4: Monitor the activity by giving the assistance where it is needed.

Step 5: Ask the trainees to read the key readings 3.3.2

 **Points to Remember**

- Steps to measure program performance using Google Chrome DevTools:

1. Open DevTools
2. Use the Performance Panel
3. Use the Profiler Panel

 **Application of learning 3.3.**

DF TSS is a TVET school that has a new school management system. Before its use, the school need software engineer to test for them the developed system and document test findings.

Check list

SN	Criteria	Indicators	Observation Yes / No
1	**User Interface (UI) Testing**	1.1. The system's user interface is visually appealing and professional.	
		1.2. All buttons, links, and navigation work as intended.	
		1.3. Forms are properly validated (e.g., no empty fields allowed, correct formats enforced).	
2	**Functionality Testing**	2.1. All key modules (e.g., student records, attendance, grades) perform their intended functions.	
		2.2. User roles (e.g., admin, teacher, and student) function correctly with appropriate access controls.	
		2.3. Error messages appear for invalid inputs and guide users effectively.	
3	**Performance Testing**	3.1. The system loads within acceptable time limits (e.g., under 5 seconds for dashboard).	
		3.2. The system handles simultaneous users (e.g., 50+ users) without crashing.	
		3.3. Large data uploads/downloads (e.g., CSV files) are processed efficiently.	
4	**Security Testing**	4.1. User passwords are encrypted and stored securely.	
		4.2. The system prevents unauthorized access to sensitive data.	
		4.3. Protection against common vulnerabilities (e.g., SQL injection, XSS) is implemented.	

Learning outcome 3 end assessment

Theoretical assessment

I. Read the following statement related to data structure and algorithm fundamentals, and answer True if the statement is correct and False if the statement is wrong:

1. The bubble sort algorithm is generally faster than quick sort for large datasets
   False. Quick sort is generally faster for large datasets compared to bubble sort.
2. Arrays and linked lists are both linear data structures
   True. Both arrays and linked lists are linear data structures, meaning their elements are arranged in a sequential manner.
3. Binary search can be performed on both sorted and unsorted arrays
   False: Binary search only works on sorted arrays.
4. Benchmark.js and jsPerf are tools used for testing and measuring the performance of JavaScript code.
   True. *Both Benchmark.js and jsPerf are popular tools used to evaluate the performance of JavaScript code.*
5. Profiling and benchmarking tools like Benchmark.js and jsPerf help evaluate the performance of JavaScript code.

   True: These tools are used to measure time and space performance for optimizing JavaScript code.

II. Read the Read the following statement related to data structure and algorithm fundamentals and choose the letter corresponding to the correct answer

1. Which of the following environments can be used to run JavaScript source code?

   a) Browser Developer Tools
   b) IDE Terminal
   c) Text Editor
   d) Both a and b
   Answer: d) Both a and b

2. Which of the following is a linear data structure?

a) Graph
b) Tree
c) Linked List
d) Hash Table
Answer: c) Linked List

3. Which sorting algorithm generally has better average performance on large datasets?

a) Bubble Sort
b) Quick Sort
c) Insertion Sort
d) Selection Sort
Answer: b) Quick Sort

4. When can binary search be used?
a) On unsorted data
b) On sorted data
c) On linked lists only
d) On any data structure
Answer: b) On sorted data

5. Which of the following tools is commonly used to profile and measure JavaScript performance?
a) Chrome DevTools
b) jsPerf
c) Benchmark.js
d) All of the above
Answer: d) All of the above

Practical assessment

Hasha Ltd is facing challenges in managing its warehouse data due to the use of hard copies for record-keeping. As a solution, you have been tasked with creating an efficient data management system that handles imports, exports, and transit processes for furniture. Use the following data structures: **Stack, Sorting, and Linked List** to develop the system.

For each question below, **Write JavaScript code** that addresses the respective problem:

6. The company receives shipments daily, and the manager needs to process them in the order they arrive. Sometimes the manager needs to undo the last action (removal of incorrect shipments). Write JavaScript code using a **Stack** to manage this process. The stack should allow the manager to undo the last operation efficiently. Define the methods needed to push, pop, and check the top of the stack.

7. The warehouse manager needs to sort furniture items by categories such as Tables, Chairs, Beds, and Dressers to produce a daily report. Write JavaScript code using a Sorting Algorithm to organize these categories.

8. The number of shipments in the warehouse can vary daily, so a flexible data structure is needed. Write JavaScript code using a Linked List to represent the furniture items in the warehouse, allowing dynamic addition and removal of items.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1.	JavaScript source code is properly development	1.1 JavaScript running environment prepared		
		1.2 Sorting operations applied		
		1.3 Searching operations applied		
2.	Time and space complexity are well Tested	2.1 Time complexity measured		
		2.2 Space complexity Measured		
		2.3 Test findings are documented		

END

Further information to the trainer

BOOKS

Algorithms, L. J. (2016). *Loiane Groner.* Birmingham, UK: Packt Publishing.

Bae, S. (2017). *Data Structures and Algorithms with JavaScrip.* UK: Packt Publishing.

Karim, M. R. (2019). *JavaScript Algorithms and Data Structures.*

Rozentals, N. (2016). *Mastering JavaScript Data Structures and Algorithms.* Birmingham, UK: Packt Publishing.

WEB LINKS

educative. (2024). */blog/data-strucutres-hash-table-javascript* . Retrieved from educative: https://www.educative.io/blog/data-strucutres-hash-table-javascript

geeksforgeeks. (2024). */rendering-engines-used-by-different-web-browsers/* . Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/rendering-engines-used-by-different-web-browsers/

javatpoint. (2024). */javascript-queue*. Retrieved from javatpoint: https://www.javatpoint.com/javascript-queue

tutorialspoint. (2024). */javascript/javascript_arrays_object.htm*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/javascript/javascript_arrays_object.htm

tutorialspoint. (n.d.). */Creating-a-Stack-in-Javascript* . Retrieved from tutorialspoint: https://www.tutorialspoint.com/Creating-a-Stack-in-Javascript

RTB | RWANDA TVET BOARD

October 2024