**PROGRAMMING FUNDAMENTALS**

**LAB TASK 2**

**Question 1) Design a flowchart, pseudocode, algorithm for processing a customer order at a restaurant, including handling special requests (Like add on).**

**Flowchart**

Cancel Order

Serve Food

Print Receipt

Yes

No

Confirm Order

Display Order Summary

Calculate Bill

Process special requests

Capture special requests

Take order

Display menu

**Algorithm**

1. Start
2. Display menu to the customer
3. Take order
4. Capture special requests
5. Process special requests
6. Calculate bill
7. Display order summary
8. Confirm order
9. Display receipt
10. Serve food
11. End

**Pseudocode**

START

PRINT “ Display Menu”

// Input

INPUT Take order

// Input

INPUT “Capture special requests”

// Process Steps

PRINT “Calculate bill”  
PRINT “Order summary”

PRINT “Confirm order”

// Conditional Statements

IF Confirmation=Yes THEN

PRINT “Receipt”

PRINT “Serve food”

ELSE

PRINT “Your order has been cancelled”

END

**Question 2)** **Design a pseudocode, algorithm for handling a customer’s deposit transaction at a bank, including checks for account validity and deposit amount conditions.**

**Algorithm**

1. Start
2. Read the customer ID from the user
3. Read the deposit amount from the user
4. If deposit amount is less than and equals to 0 print error
5. Check account validity
6. Check if account is active, if not print an error message and terminate the process
7. Add the deposit amount to the current balance
8. Inform the user that the deposit was successful
9. Display the new balance.
10. End

**Pseudocode**

START

// Input

INPUT CustomerID

// Input

INPUT deposit amount

// Conditional Statements

IF deposit amount <= 0 THEN

PRINT “Error”

// Process steps

PRINT “Check account validity through customer ID”

// Conditional Statements

IF Account status is ‘NOT ACTIVE’ THEN

PRINT “Error”

ELSE

PRINT “Add deposit to the current balance”

PRINT “Deposit successful”

PRINT “Updated balance”

END

**Question 3)** **Design a pseudocode, algorithm to determine which of three provided numbers is the greatest.**

**Algorithm**

1. Start
2. Input three numbers:num1,num2,num3
3. Set the value of greatest number as num1
4. If num2 is greater than current greatest number then greatest number is updated to num2
5. If num3 is greater than current greatest number then greatest number is updated to num3
6. Output the greatest number
7. End

**Pseudocode**

START

// Input

INPUT num1

INPUT num2

INPUT num3

// Initialize

SET greatest number= num1

// Conditional Statements

IF num2 > greatest number THEN

SET greatest number = num2

IF num3 > greatest number THEN

SET greatest number = num3

// Output

PRINT “The greatest number”

END

**Question 4) Implement an algorithm where the user enters a number, and an appropriate month is displayed.**

**Algorithm**

1. Start
2. Input a number (1-12) from the user
3. IF the number is 1, display “January”
4. ELSE IF the number is 2, display “February”
5. ELSE IF the number is 3, display “March”
6. ELSE IF the number is 4, display “April”
7. ELSE IF the number is 5, display “May”
8. ELSE IF the number is 6, display “June”
9. ELSE IF the number is 7, display “July”
10. ELSE IF the number is 8, display “August”
11. ELSE IF the number is 9, display “September”
12. ELSE IF the number is 10, display “October”
13. ELSE IF the number is 11, display “November”
14. ELSE IF the number is 12, display “December”
15. ELSE the number is not between 1 and 12 , display an error message.
16. Display the month
17. End

**Question 7) Implement an algorithm for making a simple calculator with all the operators (+,-,\*,/,%)**

**Algorithm**

1. Start
2. Input the first number, num1
3. Input the operator (+,-,\*,/,%)
4. Input the second number, num2
5. IF operator is ‘+’ , result = num1 + num2
6. ELSE IF the operator is ‘-‘ , result = num1- num2
7. ELSE IF the operator is ‘\*’ , result = num1\*num2
8. ELSE IF the operator is ‘/’ , result = num1/num2
9. ELSE IF the operator is ‘%’ , result = num1 % num2
10. If the operator is not one of the valid options, display an error message: "Invalid operator. Please use +, -, \*, /, or %."
11. Display the result
12. End

**Question 5) Create pseudocode a small calculator which only does ‘+’ or ‘-‘ Operations. (Hint: take three variable inputs with one being used for the operator)**

**Pseudocode**

START

// Input

INPUT first number, num1

// Input

INPUT Operator(+ or -)

// Input

INPUT second number, num2

// Process

IF operator is ‘+’ THEN

SET num1 + num2

ELSE IF operator is ‘-‘ THEN

SET num1 – num2

ELSE

PRINT “Invalid Operator. Use + or - only”

// Output

PRINT “Result”

END

**Question 9) Why we use gitignore?**

It helps prevent unnecessary files from being committed to the repository, keeping the repository clean and focused only on source code and important files.

**Protect Sensitive Information** by ignoring files that contain sensitive information. It ensures that irrelevant or sensitive files do not clutter the version control system, thus improving project management and collaboration.

**Question 10) Difference between algorithm and pseudocode?**

**Algorithm**

1. An algorithm is a step-by-step procedure or set of rules designed to perform a task or solve a problem. It outlines a clear sequence of actions.
2. Algorithms can be expressed in any language or form, including natural language, flowcharts, or pseudocode.

**Pseudocode**

1. Pseudocode is a way of expressing an algorithm in a structured way that resembles programming languages. It uses a combination of natural language and programming language to describe the steps of the algorithm.
2. It is designed to be easily understood by humans. It is less formal and closer to natural language than actual code.