NATIONAL UNIVERSITY of Computer & Emerging Sciences

CS1004 - Object Oriented Programming

LAB # 1

REVISING PROGRAMMING FUNDAMENTALS

Lab Objectives :

- Review fundamental programming concepts
- Practice function implementation and usage
- Strengthen problem-solving skills with control structures
- Work with arrays and basic algorithms

LAB TASKS

Task 1: Simple Calculator Function

Write a program with a function calculator() that takes two numbers and an operator (+, -, *, /) as parameters and returns the result. In main, take input from user and display the result.
Sample Input:
Enter first number: 15
Enter operator: +
Enter second number: 25
Sample Output:
Result: 40

Task 2: Find Maximum of Three

Create a function findMax() that takes three integers as parameters and returns the maximum value. In main, take three numbers from user and display which is the largest.
Sample Input:
Enter three numbers: 45 67 23
Sample Output:
Maximum value: 67

Task 3: Count Vowels

Write a function countVowels() that takes a string as parameter and returns the count of vowels (a, e, i, o, u case insensitive). In main, take a string from user and display the vowel count.
Sample Input:
Enter a string: Programming is Fun
Sample Output:
Number of vowels: 5

Task 4: Factorial Calculator

Create a function factorial() that takes a positive integer n as parameter and returns its factorial using a loop. In main, take input from user and display the factorial.
Sample Input:
Enter a number: 6
Sample Output:
Factorial of 6 is: 720

Task 5: Array Maximum Finder

Write a function findArrayMax() that takes an integer array and its size as parameters and returns the maximum element. In main, create an array of 6 elements, take input, and display the maximum value.
Sample Input:
Enter 6 numbers: 23 67 12 89 45 34
Sample Output:
Maximum element in array: 89

Task 6: Palindrome Checker

Write a function isPalindrome() that takes an integer as parameter and returns 1 if it's a palindrome, otherwise 0. A palindrome number reads the same forwards and backwards (e.g., 121, 12321). In main, check 5 numbers entered by user.
Sample Input:
Enter 5 numbers: 121 234 545 1221 999
Sample Output:
121 is a Palindrome
234 is not a Palindrome
545 is a Palindrome
1221 is a Palindrome
999 is a Palindrome

Task 7: Bubble Sort Implementation

Create the following functions:

- bubbleSort(): Takes an integer array and its size, sorts it in ascending order
- displayArray(): Displays array elements

In main, take 8 numbers from user, display original array, sort it using bubble sort, and display sorted array.
Sample Input:
Enter 8 numbers: 64 34 25 12 22 11 90 88
Sample Output:
Original Array: 64 34 25 12 22 11 90 88
Sorted Array: 11 12 22 25 34 64 88 90

Task 8: String Manipulation

Write the following functions:

- stringLength(): Returns the length of a string without using strlen()
- stringCopy(): Copies one string to another without using strcpy()
- stringCompare(): Compares two strings and returns 0 if equal, -1 if first is smaller, 1 if first is larger

Test all functions in main with user input.
Sample Input:
Enter first string: Hello
Enter second string: World
Sample Output:
Length of first string: 5
Length of second string: 5
Copied string: Hello
Comparison result: -1 (First string is smaller)

Task 9: Array Search and Statistics

Create a comprehensive program with the following functions:

- linearSearch(): Searches for an element and returns its index (-1 if not found)
- countOccurrences(): Counts how many times an element appears in array
- findSecondLargest(): Returns the second largest element in array
- calculateMean(): Returns the average of array elements

In main, create an array of 12 elements and provide a menu to test all functions.
Sample Input:
Enter 12 numbers: 45 23 67 45 89 12 45 34 67 23 90 12
Menu:

1. Search element
2. Count occurrences
3. Find second largest
4. Calculate mean
   Enter choice: 2
   Enter element to count: 45

Sample Output:
45 appears 3 times in the array

Task 10: Bank Account System

Design a mini banking system with the following functions:

- createAccount(): Takes arrays for account numbers, names, and balances for n customers
- deposit(): Adds amount to a specific account
- withdraw(): Deducts amount from account (check sufficient balance)
- checkBalance(): Displays balance of specific account
- displayAll(): Shows all accounts

Manage 4 accounts with initial balances and perform at least 3 operations (deposit, withdraw, check balance).
Sample Input:
Initialize 4 accounts:
Account 1: 1001, Ali, 5000
Account 2: 1002, Sara, 7500
Account 3: 1003, Ahmed, 3000
Account 4: 1004, Fatima, 10000
Operations:

1. Deposit 2000 to account 1001
2. Withdraw 1500 from account 1003
3. Check balance of account 1002

Sample Output:
Initial Accounts:
1001 – Ali – 5000
1002 – Sara – 7500
1003 – Ahmed – 3000
1004 – Fatima – 10000
After deposit to 1001:
1001 – Ali – 7000
After withdrawal from 1003:
1003 – Ahmed – 1500
Balance of 1002: 7500
Final Account Status:
1001 – Ali – 7000
1002 – Sara – 7500

1003 – Ahmed – 1500
1004 – Fatima – 10000