Faculty of Computing and Information Technology

Programming Fundamentals Lab **Instructor: Hafiz Anzar Ahmad**

Pre Lab – 06 **Issue Date: 14 March 2025**

BSIT Spring 2025, Morning/Afternoon **Total Marks: 60 Marks**

Lab Objectives:

1. Implement basic control structures like loops and conditional statements.
2. Develop problem-solving skills using functions.
3. Perform number manipulations without using arrays or strings.
4. Apply programming concepts to real-world scenarios like parking management and employee bonuses.

TASK 1: 10 Marks

Find GCD (Greatest Common Divisor) of Two Numbers

Question Statement:
Write a function that calculates the GCD (Greatest Common Divisor) of two numbers without recursion.

Sample Run:

Find the GCD of 48 and 18
Factors of 48: 1, 2, 3, 4, 6, 8, 12, 16, 24, 48
Factors of 18: 1, 2, 3, 6, 9, 18
Common factors: 1, 2, 3, 6
GCD (48, 18) = 6 (largest common factor)

TASK 2: 10 Marks

Reverse Digits of a Number (Don't use any array or string)

Question Statement:

Write a function that reverses the digits of an integer without using strings or arrays.

Sample Output:

```
Input: 507
Output: Reversed number: 705
```

TASK 3: 10 Marks

Check if a Number is Prime and then find its Next Prime

Question Statement:

Write a program that checks whether a given number is prime, and if it is not, find the next prime number.

You are required to:

- Use a function **isPrime(num)** to check if a number is prime.
- Use another function **findNextPrime(num)** that calls **isPrime(num)** inside it to find the next prime number after num.
- Use **main()** to take user input and display results.
-

Sample Output:

```
Input:  10
Output: 10 is not prime. Next prime: 11
```

TASK 4: 10 Marks

Find the Sum of Digits of a Number (Don't use any array)

Question Statement:

Write a function that calculates the sum of the digits of a number.

Sample Output:

```
Input:
Enter a number: 345

Output:
Sum of digits: 12
```

TASK 5: 10 Marks

Employee Bonus Calculation

Scenario:

A company decides to reward employees based on their performance and experience. The final bonus amount depends on both the performance rating and the years of service.

Question Statement:

Write a function **calculatePerformanceBonus(rating)** returns a bonus based on performance:

- Excellent (5 stars) → 20% of salary
- Good (4 stars) → 15% of salary
- Average (3 stars) → 10% of salary
- Below Average (2 or fewer stars) → No bonus

- The function **calculateExperienceBonus(years)** gives an extra **5%** of salary if the employee has **5+** years of experience.
- The main function **calculateTotalBonus(salary, rating, years)** calls both functions inside it and calculates the final bonus amount.

Sample Output:

```
Input:
Enter employee's salary: 50000
Enter performance rating (1-5): 5
Enter years of experience: 6

Output:
Performance Bonus: 10000.00
Experience Bonus: 2500.00
Total Bonus: 12500.00
```

TASK 6: 10 Marks

Parking Lot Management System

Question Statement:
A parking lot has a limited number of parking spaces. You need to implement a Parking Management System that keeps track of:

- Available slots in the parking lot.
- Vehicle Entry – A vehicle can only enter if a parking slot is available. If no slot is available, display a message: "Parking Full! No slots available."
- Vehicle Exit – When a vehicle leaves, the slot becomes available for another vehicle.

Write a function that:

- Takes the current number of occupied slots and total parking capacity as input.
- Allows vehicles to enter or exit the parking lot based on availability.
- Returns the updated number of occupied slots

Sample Output:

```
Input: Total Slots = 5, Occupied Slots = 3, Action = "Enter"
Output: Vehicle Parked! Updated Occupied Slots = 4

Input: Total Slots = 5, Occupied Slots = 5, Action = "Enter"
Output: Parking Full! No slots available.

Input: Total Slots = 5, Occupied Slots = 2, Action = "Exit"
Output: Vehicle Exited! Updated Occupied Slots = 1
```

Coding is like magic—if you learn

the right spells, you can create anything!"