Assignment - 2

C++

Name: Kamithkar Vinod
Course: PG DAC AUGUST 2025
PRN: 250850320040
Form No: 250500480
Date: 08-10-2025

Problem 1: Basic Pointer

Task: Write a program in C++ to declare an integer variable, store its address in a pointer, and display both the value of the variable and its address using the pointer.

Code:

```cpp
#include <iostream>
using namespace std;

int main() {
    int num = 10;
    int *ptr = &num;

    cout << "Value of num: " << num << endl;
    cout << "Address of num using pointer: " << ptr << endl;
    cout << "Value accessed through pointer: " << *ptr << endl;
    return 0;
}
```

Output: 1—

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>1_basic_pointer
Value of num: 10
Address of num using pointer: 0xf28ebff794
Value accessed through pointer: 10
```

Problem 2: Pointer Arithmetic

Task: Write a program in C++ to create an array of 5 integers. Use a pointer to traverse the array and print all elements using pointer arithmetic.

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int arr[5] = {10, 20, 30, 40, 50};
    int *ptr = arr;

    cout << "Array elements using pointer arithmetic:\n";
    for (int i = 0; i < 5; i++) {
        cout << *(ptr + i) << " ";
    }
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>2_pointer_arithmetic
Array elements using pointer arithmetic:
10 20 30 40 50
```

Problem 3: Call by Reference (using pointer)

Task: Write a program in C++ to swap two numbers using pointers.

Code: —

```cpp
#include <iostream>
using namespace std;

void swapNumbers(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x = 5, y = 10;
    cout << "Before Swap: x = " << x << ", y = " << y << endl;

    swapNumbers(&x, &y);

    cout << "After Swap: x = " << x << ", y = " << y << endl;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>3_call_by_reference
Before Swap: x = 5, y = 10
After Swap: x = 10, y = 5
```

Problem 4: Dynamic Memory Allocation

Task: Write a program in C++ to dynamically allocate memory for an array of 5 integers using a pointer, take input from the user, and display the array elements.

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int *arr = new int[5];

    cout << "Enter 5 integers: ";
    for (int i = 0; i < 5; i++) {
        cin >> arr[i];
    }

    cout << "Array elements: ";
    for (int i = 0; i < 5; i++) {
        cout << arr[i] << " ";
    }

    delete[] arr;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>4_dynamic_memory_allocation
Enter 5 integers: 2 5 2 4 9
Array elements: 2 5 2 4 9
```

Problem 5: Function with Pointer

Task: Write a program in C++ to pass a pointer to a function that updates the value of a variable.

Code: —

```cpp
#include <iostream>
using namespace std;

void updateValue(int *p) {
    *p = *p + 10;
}

int main() {
    int num = 5;
    cout << "Before update: " << num << endl;

    updateValue(&num);
```

```
13
14      cout << "After update: " << num << endl;
15      return 0;
16  }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>5
Before update: 5
After update: 15
```

Problem 6: String with Pointers

Task: Write a program in C++ to count the length of a string using a character pointer (without using built-in functions like strlen).

Code: —

```
1   #include <iostream>
2   using namespace std;
3
4   int main() {
5       char str[100];
6       cout << "Enter a string: ";
7       cin.getline(str, 100);
8
9       char *ptr = str;
10      int length = 0;
11
12      while (*ptr != '\0') {
13          length++;
14          ptr++;
15      }
16
17      cout << "Length of string: " << length << endl;
18      return 0;
19  }
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>
Enter a string: today is Indian Air Force Day
Length of string: 29
```

Problem 7: Dynamic Integer

Task: Write a program in C++ to dynamically allocate memory for a single integer using new, assign a value, display it, and then free the memory using delete.

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int *ptr = new int;
    *ptr = 25;

    cout << "Value: " << *ptr << endl;

    delete ptr;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>
Value: 25
```

Problem 8: Dynamic Array

Task: Write a program in C++ to create an array of 5 integers using new, take input from the user, display the array elements, and then release the memory using delete[].

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int *arr = new int[5];

    cout << "Enter 5 integers: ";
    for (int i = 0; i < 5; i++) {
        cin >> arr[i];
    }

    cout << "Array elements: ";
    for (int i = 0; i < 5; i++) {
        cout << arr[i] << " ";
    }

    delete[] arr;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>8
Enter 5 integers: 2 4 5 2 9
Array elements: 2 4 5 2 9
```

Problem 9: Dynamic 2D Array

Task: Write a program in C++ to create a 2D array (matrix) dynamically using new. Take input for rows and columns from the user, fill the matrix, display it, and free memory using delete[].

Code:

```cpp
#include <iostream>
using namespace std;

int main() {
    int rows, cols;
    cout << "Enter rows and columns: ";
    cin >> rows >> cols;

    int **matrix = new int*[rows];
    for (int i = 0; i < rows; i++)
        matrix[i] = new int[cols];

    cout << "Enter matrix elements:\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            cin >> matrix[i][j];

    cout << "Matrix:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            cout << matrix[i][j] << " ";
        cout << endl;
    }

    for (int i = 0; i < rows; i++)
        delete[] matrix[i];
    delete[] matrix;

    return 0;
}
```

Output:

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>
Enter rows and columns: 2 4
Enter matrix elements:
4 5 9 7
1 2 3 4
Matrix:
4 5 9 7
1 2 3 4
```

Problem 10: Dynamic String

Task: Write a program in C++ to dynamically allocate memory for a string using a character pointer and new. Take user input for the string, display it, and then free the memory.

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "Enter the size of string: ";
    cin >> size;

    char *str = new char[size];
    cout << "Enter string: ";
    cin.ignore();
    cin.getline(str, size);

    cout << "You entered: " << str << endl;

    delete[] str;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>
Enter the size of string: 9
Enter string: Indian
You entered: Indian
```

Problem 11: Function Returning Dynamic Memory

Task: Write a program in C++ with a function that returns a pointer to a dynamically allocated array. In main(), call the function, display the array, and free the memory.

Code: —

```cpp
#include <iostream>
using namespace std;

int* createArray(int n) {
    int *arr = new int[n];
    cout << "Enter " << n << " integers: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];
    return arr;
}

int main() {
    int n = 5;
    int *ptr = createArray(n);

    cout << "Array elements: ";
    for (int i = 0; i < n; i++)
        cout << ptr[i] << " ";

    delete[] ptr;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>
Enter 5 integers: 2 42 45 78 9
Array elements: 2 42 45 78 9
```

Problem 12: Pointer Re-allocation (basic simulation)

Task: Write a program in C++ to dynamically allocate an array of integers using new, fill it with values, then allocate a bigger array, copy the old values into it, add more elements, and release both old and new arrays properly.

Code: —

```cpp
#include <iostream>
using namespace std;

int main() {
    int n = 3;
    int *arr = new int[n];

    cout << "Enter 3 integers: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    int newSize = 5;
```

```
    int *newArr = new int[newSize];

    for (int i = 0; i < n; i++)
        newArr[i] = arr[i];

    cout << "Enter 2 more integers: ";
    for (int i = n; i < newSize; i++)
        cin >> newArr[i];

    cout << "New array: ";
    for (int i = 0; i < newSize; i++)
        cout << newArr[i] << " ";

    delete[] arr;
    delete[] newArr;
    return 0;
}
```

Output: —

```
V:\CDAC\CDAC_PG_DAC_PRACTICE\4_CPP\2_Assignments>1
Enter 3 integers: 24 25 99
Enter 2 more integers: 4 8
New array: 24 25 99 4 8
```