Introduction to Programing

Group 4

Main Thread

Exercise 9 & 10

2-D ARRAY

```cpp
#include <iostream>

int main()
{
    const int n = 2;
    const int m = 2;

    // 2 Dimensional Array of type INT
    int arr[n][m] = { {1, 2}, {3, 4} };

    // The first element is in position 0, 0
    std::cout << arr[0][0] << "\n";

    return 0;
}
```

ELEMENTS

```cpp
#include <iostream>

int main()
{
    const int n = 2;
    const int m = 2;

    int arr[n][m] = { {1, 2}, {3, 4} };

    // To display the elements of 2-D array
    // You need to use two cycles
    // The first cycle is going through the rows
    // The second cycle is going through the columns

    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < m; ++j)
        {
            std::cout << arr[i][j] << " ";
        }
        std::cout << "\n";
    }

    return 0;
}
```

INPUT ELEMENTS

```cpp
#include <iostream>

int main()
{
    const int n = 2;
    const int m = 2;

    int arr[n][m];

    // To display the elements of 2-D array
    // You need to use two cycles
    // The first cycle is going through the rows
    // The second cycle is going through the columns

    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < m; ++j)
        {
            std::cout << "Input element in position " << i << " , " << j;
            std::cin >> arr[i][j];
        }
    }

    return 0;
}
```

SEARCH ELEMENT

```cpp
#include <iostream>

int main()
{
    int searchNumber;
    bool isNumberFound = false;

    const int n = 2;
    const int m = 2;

    int arr[n][m] = { {1, 2}, {3, 4} };

    std::cout << "Input the number you want to search for: ";
    std::cin >> searchNumber;

    // Search element
    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < m; ++j)
        {
            if (arr[i][j] == searchNumber)
            {
                isNumberFound = true;
                break;
            }
        }
    }

    if (isNumberFound)
    {
        std::cout << "The number " << searchNumber << " is contained in the array\n";
    }
    else
    {
        std::cout << "The number " << searchNumber << " is not contained in the array\n";
    }

    return 0;
}
```

2-D VECTOR

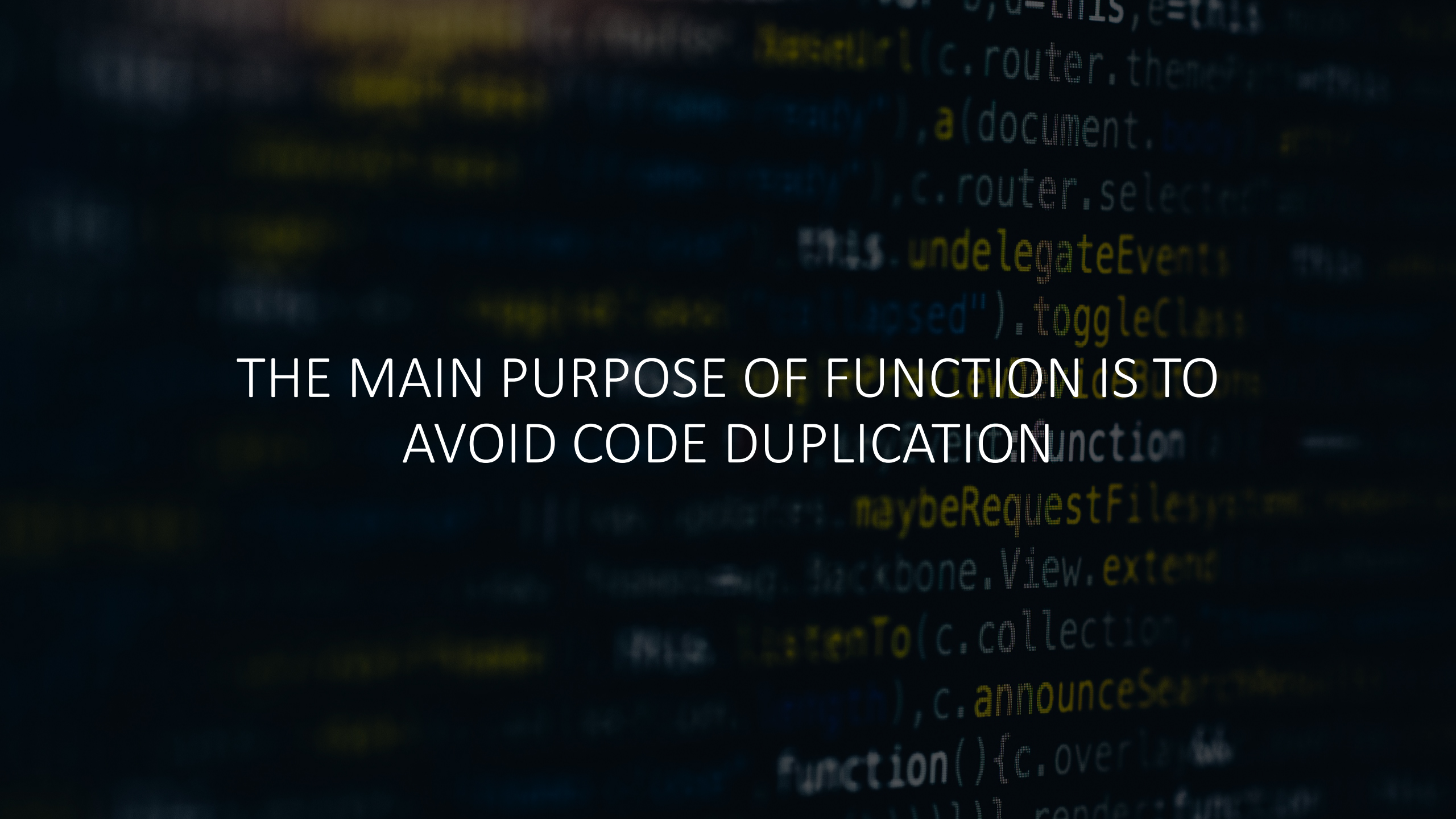
CLASS

Dynamic Array

INPUT ELEMENTS

```cpp
#include <iostream>
#include <vector>

int main()
{
    const int n = 2;

    std::vector<std::vector<int>> arr;

    for (int i = 0; i < n; ++i)
    {
        arr.push_back(std::vector<int>());
        for (int j = 0; j < n; ++j)
        {
            arr[i].push_back(i + j);
        }
    }

    return 0;
}
```


FUNCTIONS


THE MAIN PURPOSE OF FUNCTION IS TO
AVOID CODE DUPLICATION

```cpp
#include <iostream>

int Multiply(int, int);
void MultiplyAndLog(int, int);

int main()
{
    // Aim - decomposition
    // The optimized variant of the code
    MultiplyAndLog(8, 3);
    MultiplyAndLog(2, 5);
    MultiplyAndLog(345, 242);

    // The not optimized variant of the code
    int firstResult = Multiply(8, 3);
    std::cout << firstResult << "\n";

    int secondResult = Multiply(2, 5);
    std::cout << secondResult << "\n";

    int thirdResult = Multiply(345, 242);
    std::cout << thirdResult << "\n";

    return 0;
}

int Multiply(int a, int b)
{
    return a * b;
}

void MultiplyAndLog(int a, int b)
{
    std::cout << Multiply(a, b) << "\n";
}
```


Thank you for watching