# Object Oriented Programming – Lab 1

**Instructor** – Abdullah Yaqoob

**Name:** Meher Ali **-- Class:** BCS-3A **-- Roll no**: 24K-0545

## Task#01

**Source Code:**

Task-1.cpp > ...

```cpp
//TASK#01 by Meher Ali - 24K-0545
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    system("cls");//------------------------------------------------------------to clear terminal
    int amount;
    cout << "Enter amount: ";//-------------------------------------------------takes input from the user
    cin >> amount;

    int denominations[] = {5000, 1000, 500, 100, 20, 10, 5, 2, 1};//----------initializes an array of denominations given
    int size = sizeof(denominations) / sizeof(denominations[0]);//------------calculates the size of the array

    cout << "Minimum notes required for Rs. " << amount << " are:" << endl;

    for (int i = 0; i < size; i++) {//-----------------------------------------this loop runs for the size of the array
        if (amount >= denominations[i]) {//-----------------------------------checks if the amount is greater than or
        equal to the denomination
            int count = amount / denominations[i];//--------------------------calculates the number of notes required for
            that denomination
            amount %= denominations[i];//-------------------------------------updates the amount to the remainder after
            using that denomination
            cout << denominations[i] << " x " << count << endl;//-------------prints the denomination and the number of
            notes required
        }
    }

    return 0;
}
```

**Output:**

```
Enter amount: 5768
Minimum notes required for Rs. 5768 are:
5000 x 1
500 x 1
100 x 2
20 x 3
5 x 1
2 x 1
1 x 1
PS C:\Users\Lenovo\Desktop\OOP Lab-1>
```

# Task#02

## Source Code:

Task-2.cpp > ...

```cpp
//TASK#02 by Meher Ali - 24K-0545
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    system("cls");//-----------------------------------------------to clear terminal
    int rows;//----------------------------------------------------initializes rows variable and takes input from the user
    cout << "Enter the Number of Rows: ";
    cin >> rows;

    char alphabets[26];//------------------------------------------creates an array of alphabets and generates every letter and stores it in the array
    for (int i = 0; i < 26; i++) {
        alphabets[i] = 'A' + i;
    }

    for (int i = 1; i <= rows; i++) {//----------------------------loop to print top half (including middle row)
        for (int space = 1; space <= rows - i; space++) {//--------print spaces
            cout << " ";
        }
        for (int j = 0; j < 2 * i - 1; j++) {//--------------------print alphabets until 2*i - 1
            cout << alphabets[j];
        }
        cout << "\n";//-------------------------------------------new line
    }

    for (int i = rows - 1; i >= 1; i--) {//------------------------loop to print bottom half (excluding middle row)
        for (int space = 1; space <= rows - i; space++) {//--------print spaces
            cout << " ";
        }
        for (int j = 0; j < 2 * i - 1; j++) {//--------------------print alphabets until 2*i - 1
            cout << alphabets[j];
        }
        cout << "\n";//-------------------------------------------znew line
    }

    return 0;
}
```

## Output:

```
Enter the Number of Rows: 5
    A
   ABC
  ABCDE
 ABCDEFG
ABCDEFGHI
 ABCDEFG
  ABCDE
   ABC
    A
PS C:\Users\Lenovo\Desktop\OOP Lab-1>
```

# TASK#03

## Source Code:

Task-3.cpp > ...

```cpp
//TASK#03 by Meher Ali - 24K0545
#include <iostream>
#include <math.h>
#include <cstdlib>
using namespace std;

bool checkIfPrime(int num) {//----------------------------------this function checks if the number is prime or not
    if (num <= 1)//-------------------------------------------if the number is 1 or negative, its neglected;
        return false;
    for (int i = 2; i <= sqrt(num); i++) {//--------------------checking the factors of the number
        if (num % i == 0)//-----------------------------------if the ramainder of the number and its factor is 0 then its not prime
            return false;
    }
    return true;//---------------------------------------------if neither of the conditions are false, then the number is prime
}

void FindPrimeInRange(int start,int end) {//----------------------function to check the primes in a range of two numbers
    cout << "The Prime Numbers between "//----------------------a prompt
         << start << " and " << end << "are: " << endl;
    for (int i = start; i <= end; i++) {//----------------------a loop which puts every number in the checkIfPrime() function to check if its prime or not
        if (checkIfPrime(i) == true) {//-------------------------if a number is prime, its printed through the prompt inside the if-block
            cout << i << " ";
        }
    }
}
int main(){
    system("cls");//---------------------------------------------system("cls") to clear terminal
    int start, end;//-------------------------------------------initialization of two integers
    cout << "Enter the start of Range: "; cin >> start;//--------input from user
    cout << "Enter the end of the range: "; cin >> end;//--------input from user
    FindPrimeInRange(start, end);//-----------------------------vlue passed through the function to check the prime numbers between 'start' and 'end'
    return 0;
}
```

## Ouptut:

```
Enter the start of Range: 10
Enter the end of the range: 57
The Prime Numbers between 10 and 57are:
11 13 17 19 23 29 31 37 41 43 47 53
PS C:\Users\Lenovo\Desktop\OOP Lab-1>
```

# TASK#04

Task-4.cpp > swapArray(int [], int [], int)

Click to add a breakpoint

```cpp
//er Ali 24K-0545
#include <iostream>
#include <cstdlib>
using namespace std;

void swapArray(int arr[], int arr2[], int size) {//----------------a function to swap two arrays by using a temporary array
    for (int i = 0; i < size; i++) {//-----------------------------a loop to check and transfer every value from one arr to another
        int temp = arr[i];//---------------------------------------arr[i]-->temp which makes arr[i] = 0
        arr[i] = arr2[i];//----------------------------------------arr2[i]-->arr[i] which makes arr2[0] and the value at index i of arr2 is now in i'th index of arr[i]
        arr2[i] = temp;//------------------------------------------temp--->arr2[i] which empties the temp[] and transfers the value of arr[i] into arr[2]
    }
}


int checkforLargest(int arr[], int size){//------------------------a function to check for Largest element in an array
    int largest = arr[0];//-----------------------------------------a variable 'Largest' is initialized with the first value of the array
    for (int i = 0; i < size ; i++) {//----------------------------this loop runs = number of elements in the array
        if (arr[i] > largest){//-----------------------------------condition checks whether an element on i'th index is greater than largest or not
            largest = arr[i];//-----------------------------------if yes, then its value is transferred into largest... making it the largest
        }
    }
    return largest;//------------------------------------------------finally it returns, the largest value
}

int checkForSecondLargest(int arr[], int size){//------------------a function to check for the second largest element in an array
    int largest = checkforLargest(arr,size);//---------------------to get the largest value, we ran the previous checkforLargest() function
    int secondLargest = arr[0];//-----------------------------------initializing secondLargest variable with the first index of the array
    for (int i = 0 ; i < size ; i++) {//---------------------------this loop runs for the number of elements in the array i.e. size
        if (arr[i] > secondLargest){//-----------------------------checes if the i'th element of the array is greater than second largest
            secondLargest = arr[i];//-----------------------------if true, then it gets transferred in the variable,
            if(secondLargest < largest){//------------------------then it gets compared to the Largest variable from checkforLargest()
                return secondLargest;//---------------------------if its less...its returned as the second largest
            }
        }
    }

}

int findUnique(int arr[], int size) {//----------------------------this function checks whether an array has a unique number or not
    for (int i = 0; i < size; i++) {//-----------------------------checks every value in arr[]
        bool isUnique = true;//-----------------------------------bool isUnique is toggled to "true" initially
        for (int j = 0; j < size; j++) {//------------------------checks if the values of arr[] are same or not
            if (i != j && arr[i] == arr[j]) {//--------------------a condition which checks if numbers in an array are same or not
                isUnique = false;//-----------------------------if they are same, then isUnique is false
                break;//-----------------------------------------and then the nested loop is broken, just to be started again with respective increments or decrements
```

**Continue…**

```
45                  break;//-----------------------------------------and then the nested loop is broken, just to be started
                    again with respective increments or decrements
46              }
47          }
48          if (isUnique) {//-----------------------------------------a condition to check if the isUnique is true or not
49              return arr[i];///-----------------------------------if its true, then the unique value will be returned
50          }
51      }
52      return arr[0];//-----------------------------------------------or else the first value will be returned
53  }
54
55  int main(){
56      system("cls");//-----------------------------------------------system("cls") to clean the terminal before executing
        the rest of the program
57      int size;
58      cout << "Enter size of Array: ";//----------------------------prompt to take in the value of the size of both of the
        arrays
59      cin >> size;
60
61      int arr[size], arr2[size];//---------------------------------both arrays are initialized with their sizes
62
63      cout << "1)Values in the first Array: ";//--------------------taking values input for first array using loop
64      for(int i = 0 ; i < size ; i++) {
65          cin >> arr[i];
66      }
67      cout << "  Values in the second Array: ";//-------------------taking value intput for second array using loop
68      for(int i = 0 ; i < size ; i++) {
69          cin >> arr2[i];
70      }
71      cout << "2)Arrays before swapping: " << endl;//----------------prints both the array before swapping using a loop
72      cout << "  Array-1: [";//---------------------------------- prints first array in [n,n,n,n] format
73      for ( int i = 0 ; i < size ; i++) {
74          cout << arr[i];
75          if(i != size-1) {
76              cout << ",";
77          }
78      }
79      cout << "]";
80      cout << "\n  Array-2: [";//---------------------------------prints second array in [n,n,n,n] format
81          for ( int i = 0 ; i < size ; i++) {
82              cout << arr2[i];
83              if(i != size-1) {
84                  cout << ",";
85              }
86          }
87          cout << "]" << endl;
88
89      swapArray(arr , arr2 , size);//-----------------------------function to swap elemennts in both arrays
90
91      cout << "  After Swapping: " << endl;//----------------------prompt to start printing both arrays after swapping
92      cout << "  Array-1: [";//------------------------------------prompt before printing first array
93      for ( int i = 0 ; i < size ; i++) {//-------------------------prints array in [n,n,n,n] format
94          cout << arr[i];
95          if(i != size-1) {
96              cout << ",";
```

```
97          }
98      }
99      cout << "]";
100     cout << "\n  Array-2: [";//-----------------------------------prompt before printing second array
101     for ( int i = 0 ; i < size ; i++) {//-------------------------loop prints array in [n,n,n,n] format
102         cout << arr2[i];
103         if(i != size-1) {
104             cout << ",";
105         }
106     }
107     cout << "]";
108
109     int largest = checkforLargest(arr,size);//--------------------initializes variable with the largest variable from
        checkForLargest() function
110     cout << "\n3)The largest value in [";//----------------------prompt before printing first array
111     for ( int i = 0 ; i < size ; i++) {//-------------------------loop prints first array in [n,n,n,n] format
112         cout << arr[i];
113         if(i != size-1) {
114             cout << ",";
115         }
116     }
117     cout << "] is " << largest << endl;
118
119     int secondLargest = checkForSecondLargest(arr2,size);//--------initializes variable with second largest value
        returned from checkforsecondlargest() function
120     cout << "4)The Second largest value in [";//------------------prompt before printing second array
121     for ( int i = 0 ; i < size ; i++) {//-------------------------loop prints second array in [n,n,n,n] format
122         cout << arr2[i];
123         if(i != size-1) {
124             cout << ",";
125         }
126     }
127     cout << "] is " << secondLargest << endl;
128
129     int unique = findUnique(arr,size);//--------------------------initializes variable with a unique value returned from
        findUnique() function
130     cout << "5)The Unique number in the first array is " //--------prompt to show the unique value in firsta rray after
        swapping
131          << unique << endl;
132     return 0;
133 }
```

.

**Output:**

```
Enter size of Array: 4
1)Values in the first Array: 1 2 3 4
  Values in the second Array: 5 6 7 8
2)Arrays before swapping:
  Array-1: [1,2,3,4]
  Array-2: [5,6,7,8]
  After Swapping:
  Array-1: [5,6,7,8]
  Array-2: [1,2,3,4]
3)The largest value in [5,6,7,8] is 8
4)The Second largest value in [1,2,3,4] is 2
5)The Unique number in the first array is 5
PS C:\Users\Lenovo\Desktop\OOP Lab-1>
```

# TASK#05

## Source Code:

Task-5.cpp > ...

```cpp
//TASK # 05 by Meher Ali 24K-0545
#include <iostream>
#include <cstdlib>
using namespace std;

void multiplyMatrices(int* A, int* B, int* C, int m, int n, int p) {//a function to multiply two matrices
    for(int i = 0; i < m; i++) { //-----------------------------------loop for rows of first matrix
        for(int j = 0; j < p; j++) { //-------------------------------loop for columns of second matrix
            *(C + i*p + j) = 0; //----------------------------------initialize result cell with 0
            for(int k = 0; k < n; k++) { //-------------------------loop to perform multiplication & summation
                *(C + i*p + j) += (*(A + i*n + k)) * (*(B + k*p + j));// (row element of A * column element of B) added in
            }
        }
    }
}

int main() {
    system("cls");//--------------------------------------------------to clear terminal before running the program
    int m, n, n2, p; //-----------------------------------------------variables for dimensions of matrices

    cout << "Enter rows and columns of the first matrix: ";
    cin >> m >> n; //-------------------------------------------------taking rows & cols input for first matrix

    cout << "Enter rows and columns of the second matrix: ";
    cin >> n2 >> p; //------------------------------------------------taking rows & cols input for second matrix

    if(n != n2) { //--------------------------------------------------validation: columns of first == rows of second
        cout << "Matrix multiplication not possible. Columns of first must equal rows of second.";
        return 0; //--------------------------------------------------exit program if rule not satisfied
    }

    int* A = new int[m*n]; //-----------------------------------------dynamic allocation for first matrix
    int* B = new int[n2*p]; //----------------------------------------dynamic allocation for second matrix
    int* C = new int[m*p]; //-----------------------------------------dynamic allocation for result matrix

    cout << "Enter elements of the first matrix:\n";
    for(int i = 0; i < m; i++) { //-----------------------------------loop for filling first matrix
        for(int j = 0; j < n; j++) {
            cin >> *(A + i*n + j);
        }
    }

    cout << "Enter elements of the second matrix:\n";
    for(int i = 0; i < n2; i++) { //----------------------------------loop for filling second matrix
        for(int j = 0; j < p; j++) {
            cin >> *(B + i*p + j);
        }
    }

    multiplyMatrices(A, B, C, m, n, p); //----------------------------function call to multiply matrices

    cout << "Resultant Matrix after Multiplication:\n";
    for(int i = 0; i < m; i++) { //-----------------------------------loop for printing result matrix
        for(int j = 0; j < p; j++) {
            cout << *(C + i*p + j) << " "; //-------------------------prints each element in matrix format
        }
        cout << endl;
    }
```

**continued….**

```
60      delete[] A; //-----------------------------------------------------free memory for first matrix
61      delete[] B; //-----------------------------------------------------free memory for second matrix
62      delete[] C; //-----------------------------------------------------free memory for result matrix
63
64      return 0;
65  }
66
```

**Output:**