OOP Lab

Name: **Soham Das**

Section: **A1**

Roll No: **002311001004**

Assignment - 1

IT-UG2

2. Write a program that accepts two integers from keyboard, adds them and prints their values. Use cin and cout.

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

int main()
{
    int num1, num2, sum;

    cout << "Enter the first integer: ";
    cin >> num1;

    cout << "Enter the second integer: ";
    cin >> num2;

    sum = num1 + num2;

    cout << "The sum of " << num1 << " and " << num2 << " is: " << sum
<< endl;

    return 0;
}
```

3. Create a factorial table using cout as follows:

1! = 1

2! = 2

3! = 6

...

6! = 720

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

int main()
{
    int factorial = 1;

    for (int i = 1; i <= 6; i++)
    {
```

```
        factorial *= i;
        cout << i << "! = " << factorial << endl;
    }

    return 0;
}
```

4. Write a program to print 1 to 10 using a for loop. Declare the loop variable inside the for loop. Check the scope of this variable.

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

int main() {
    for (int i = 1; i <= 10; i++) {
        cout << i << " ";
    }
    cout << endl;

    // Trying to access the loop variable 'i' outside the loop will
result in an error.
    // Uncommenting the line below will cause a compilation error.
    // cout << i << endl;

    return 0;
}
```

5. Write a program to display Celsius to Fahrenheit conversion table using a for loop. Consider only 0˚ to 100˚ Celsius. Declare variables when they are used for the first time.

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

int main()
{
    cout << "Celsius\tFahrenheit" << endl;

    for (int celsius = 0; celsius <= 100; celsius++)
    {
        double fahrenheit = (celsius * 9.0 / 5.0) + 32.0;
        cout << celsius << "\t" << fahrenheit << endl;
    }
```

```
    return 0;
}
```

6. Write a program that defines a constant PI and takes radius of a circle from keyboard and prints area of that circle.

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

int main()
{
    const double PI = 3.14159;
    double radius;

    cout << "Enter the radius of the circle: ";
    cin >> radius;

    double area = PI * radius * radius;

    cout << "The area of the circle is: " << area << endl;

    return 0;
}
```

7. Write a function that takes an integer and returns the factorial of that number. Declare function parameter as const. Call the function with some argument from main function, store the result and print it.

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

int factorial(const int n) {
    int result = 1;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
```

```
    int number;
    cout << "Enter a number to calculate its factorial: ";
    cin >> number;

    int fact = factorial(number);
    cout << "The factorial of " << number << " is: " << fact << endl;

    return 0;
}
```

8. Write a function swap() that takes two integer arguments and interchanges the values of those arguments using reference. Now in the main function, instantiate two integer variables with some values. Print their values. Call the swap function with these variables. Finally print the values of those variables. Check the result.

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

void swap(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}

int main()
{
    int x = 5, y = 10;

    cout << "Before swapping:" << endl;
    cout << "x = " << x << ", y = " << y << endl;

    swap(x, y);

    cout << "After swapping:" << endl;
    cout << "x = " << x << ", y = " << y << endl;

    return 0;
}
```

9. Now write another function swap() that takes two strings (character array) and interchanges them without reference parameters. Test this function using some

arguments. Rewrite the function using reference parameters. Again test this function with some arguments .

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

void swapWithoutReference(char a[], char b[])
{
    char temp[100];
    strcpy(temp, a);
    strcpy(a, b);
    strcpy(b, temp);
}

int main()
{
    char str1[] = "Hello";
    char str2[] = "World";

    cout << "Before swapping:" << endl;
    cout << "str1 = " << str1 << ", str2 = " << str2 << endl;

    swapWithoutReference(str1, str2);

    cout << "After swapping (without reference):" << endl;
    cout << "str1 = " << str1 << ", str2 = " << str2 << endl;

    return 0;
}
```

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

void swapWithReference(char &a[], char &b[]) {
    char temp[100];
    strcpy(temp, a);
    strcpy(a, b);
    strcpy(b, temp);
}

int main() {
    char str1[] = "Hello";
    char str2[] = "World";
```

```
    cout << "Before swapping:" << endl;
    cout << "str1 = " << str1 << ", str2 = " << str2 << endl;

    swapWithReference(str1, str2);

    cout << "After swapping (with reference):" << endl;
    cout << "str1 = " << str1 << ", str2 = " << str2 << endl;

    return 0;
}
```

10. Write a function that takes an integer and returns the factorial of that number. Declare function parameter as read only reference. Call the function with some argument from main function, store the result and print it.

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

int factorial(const int &n)
{
    return (n <= 1) ? 1 : n * factorial(n - 1);
}

int main()
{
    int number = 5;
    int result = factorial(number);
    cout << result << endl;
    return 0;
}
```

11. Write a function Strcpy to copy one string to another with suitable formal parameters declarations. Following points must be considered.

a) Source string must not get modified

b) Target string is allowed to get modified

c) The Pointers must be constant pointers.

Use it to copy some strings.

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

void Strcpy(char *target, const char *source)
{
    while (*source)
        *target++ = *source++;
    *target = '\0';
}

int main()
{
    const char *source = "Hello, World!";
    char target[50];
    Strcpy(target, source);
    cout << target << endl;
    return 0;
}
```

12. Write an inline function add() that takes three integer arguments and returns the sum of these arguments.

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

inline int add(int a, int b, int c)
{
    return a + b + c;
}

int main()
{
    cout << add(5, 10, 15) << endl;
    return 0;
}
```

13. Consider the following two scenarios:

a) We want to find out the maximum between three integers.

b) We also want to find out the maximum element of an array of integers.

Write two overloaded functions for these two scenarios

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

int maximum(int a, int b, int c)
{
    return max(a, max(b, c));
}

int maximum(int arr[], int size)
{
    int maxElem = arr[0];
    for (int i = 1; i < size; ++i)
        if (arr[i] > maxElem)
            maxElem = arr[i];
    return maxElem;
}

int main()
{
    int a = 10, b = 20, c = 15;
    cout << maximum(a, b, c) << endl;

    int arr[] = {1, 5, 3, 9, 7};
    int size = sizeof(arr) / sizeof(arr[0]);
    cout << maximum(arr, size) << endl;

    return 0;
}
```

14. Write two overloaded functions print() such that one prints the elements of a vector and the other prints elements of a matrix. Note that a vector and a matrix may be represented as a one-dimensional array and a two-dimensional array respectively.

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

void print(int arr[], int size)
{
    for (int i = 0; i < size; ++i)
        cout << arr[i] << " ";
    cout << endl;
}
```

```cpp
void print(int matrix[][3], int rows)
{
    for (int i = 0; i < rows; ++i)
    {
        for (int j = 0; j < 3; ++j)
            cout << matrix[i][j] << " ";
        cout << endl;
    }
}

int main()
{
    int vector[] = {1, 2, 3, 4, 5};
    int vectorSize = sizeof(vector) / sizeof(vector[0]);
    print(vector, vectorSize);

    int matrix[2][3] = {{1, 2, 3}, {4, 5, 6}};
    print(matrix, 2);

    return 0;
}
```

15. Consider function add() in 13. Specify the default values for second and third parameters to 0 (zero). Now call this function with three, two and one arguments and see the result.

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

int maximum(int a, int b = 0, int c = 0)
{
    return max(a, max(b, c));
}

int main()
{
    cout << maximum(10, 20, 15) << endl;
    cout << maximum(10, 20) << endl;
    cout << maximum(10) << endl;

    return 0;
}
```