**Assignment - 35** **A Job Ready Bootcamp in C++, DSA and IOT**

## Templates

1. Write a C++ program to demonstrate the addition of multiple types of data using generic functions or templates.

```
#include <iostream>
using namespace std;
template <class I, class F = double, class D = double>
D sum(I a, F b)
{
    return (a + b);
}

int main()
{
    double s1 = sum<float, float, float>(15.25, 26.54);
    double s2 = sum<double, float>(15.25, 26.5);
    double s3 = sum<int, float, float>(15.25, 26.5);
    double s4 = sum<int, int, int>(15, 26);
    double s5 = sum<double, int, int>(15.25, 26);
    cout << "Sum of s1 " << s1 << endl;
    cout << "Sum of s2 " << s2 << endl;
    cout << "Sum of s3 " << s3 << endl;
    cout << "Sum of s4 " << s4 << endl;
    cout << "Sum of s5 " << s5 << endl;
    return 0;
}
==============================================================================
Output:
Sum of s1 41.79
Sum of s2 41.75
Sum of s3 41.5
Sum of s4 41
Sum of s5 41
```

2. Write a C++ Program to find Largest among two numbers using function template.

```
#include <iostream>
using namespace std;
template <class A, class B = int, class C = int>
C BigNum(A x, B y)
{
    if (x > y)
        return x;
    else
        return y;
}
int main()
{
    int large1 = BigNum<int, int, int>(15, 25);
    double large2 = BigNum<double, int, double>(5.25, 8);
    float large3 = BigNum<float, double, float>(10.4, 9.24);
    cout << "Larger is " << large1 << endl;
    cout << "Larger is " << large2 << endl;
    cout << "Larger is " << large3 << endl;
    return 0;
}
==============================================================================
Output:
Larger is 25
Larger is 8
Larger is 10.4
```

3. Write a C++ program to find the largest of three elements using a template.

```
#include <iostream>
using namespace std;
template <class A, class B = int, class C = int, class D = double>
D BigNum(A x, B y, C z)
{
    if (x >= y)
    {
        if (x >= z)
            return x;
        else
            return z;
    }
    else
    {
        if (y >= z)
            return y;
        else
            return z;
    }
}
int main()
{
    int large1 = BigNum<int, int, int, int>(15, 25, 10);
    double large2 = BigNum<double, int, double>(5.25, 8, 22.24);
    float large3 = BigNum<float, double, float, float>(10.4, 9.24, 8.6);
    cout << "Larger is " << large1 << endl;
    cout << "Larger is " << large2 << endl;
    cout << "Larger is " << large3 << endl;
    return 0;
}
=============================================================================
Output:
Larger is 25
Larger is 22.24
Larger is 10.4
```

4. Write a C++ Program to Swap data using function template.

```
#include <iostream>
using namespace std;
template <class A>
void Swap(A &a, A &b)
{
    A c;
    c = a;
    a = b;
    b = c;
}

int main()
{
    int x = 15;
    int y = 25;
    cout << "Before Swapping" << endl;
    cout << "x = " << x << "\ny = " << y << endl;
    Swap<int>(x, y);
    cout << "After Swapping" << endl;
    cout << "x = " << x << "\ny = " << y << endl;
    float a = 45.4;
    float b = 35.2;
    cout << "Before Swapping" << endl;
    cout << "a = " << a << "\nb = " << b << endl;
    Swap<float>(a, b);
    cout << "After Swapping" << endl;
    cout << "a = " << a << "\nb = " << b << endl;
    return 0;
}
==============================================================================
Output:
Before Swapping
x = 15
y = 25
After Swapping
x = 25
y = 15
Before Swapping
a = 45.4
b = 35.2
After Swapping
a = 35.2
b = 45.4
```

5. Write a C++ Program to Add two numbers using function template.

```
#include <iostream>
using namespace std;
template <class A, class B = int, class C = double>
C sum(A a, B b)
{
    return (a + b);
}
int main()
{
    cout << "Sum is " << sum<int, int, int>(5, 10) << endl;
    cout << "Sum is " << sum<float, float, float>(5.2, 10.2) << endl;
    cout << "Sum is " << sum<double, double, double>(10.254, 45.364) << endl;
    cout << "Sum is " << sum<float, float>(10.2, 45.3) << endl;
    return 0;
}
==============================================================================
Output:
Sum is 15
Sum is 15.4
Sum is 55.618
Sum is 55.5
```

6. Write a C++ Program to find Sum of Array using function template.

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

template <class T>
T sumOfArray(T a[], int size)
{
    T sum = 0;
    for (int i = 0; i < size; i++)
        sum += a[i];

    return sum;
}

int main()
{
    // for integer value
    int a[100], size;
    cout << "Enter size of array: ";
    cin >> size;
    cout << "Enter array elements" << endl;
    for (int i = 0; i < size; i++)
        cin >> a[i];
    cout << "Sum of array elements: " << sumOfArray(a, size) << endl;
    // for float value
    float b[100];
    int s;
    cout << "Enter size of array: ";
    cin >> s;
    cout << "Enter array elements" << endl;
    for (int i = 0; i < s; i++)
        cin >> b[i];
    cout << "Sum of array elements: " << sumOfArray(b, s) << endl;
    return 0;
}
==============================================================================
Output:
Enter size of array: 5
Enter array elements
10 20 30 40 50
Sum of array elements: 150
Enter size of array: 5
Enter array elements
1.2 2.1 3.2 6.5 5.4
Sum of array elements: 18.4
```

7. Write a C++ Program of Templated class derived from Non-templated class.

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

class Shape
{
public:
    void print()
    {
        cout << "Area of rectangle: ";
    }
};
template <class T1 = int, class T2 = int>
class Rectangle : public Shape
{
    T1 x;
    T2 y;

public:
    Rectangle(T1 a, T2 b)
    {
        x = a;
        y = b;
    }
    void displayArea()
    {
        cout << (x * y) << " Unit" << endl;
    }
};

int main()
{
    // for integer value
    Rectangle r(5, 8);
    r.print();
    r.displayArea();

    // for double value
    Rectangle r1(101.235, 128.356);
    r1.print();
    r1.displayArea();
    return 0;
}
==============================================================================
Output:
Area of rectangle: 40 Unit
Area of rectangle: 12994.1 Unit
```

8. Write a C++ Program to implement push and pop methods from stack using template.

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

template <class T>
class Stack
{
    private:
        int top;
        T st[100];
    public:
        Stack()
        {
            top = -1;
        }
        void push(T value)
        {
            st[++top] = value;
        }
        T pop()
        {
            return st[top--];
        }
};

int main()
{
    Stack <int> s1;
    Stack <string> s2;

    s1.push(10);
    s1.push(20);

    s2.push("Shekh");
    s2.push("Akhtar");

    cout << s1.pop() << " " << s1.pop() << endl;
    cout << s2.pop() << " " << s2.pop() << endl;

    return 0;
}
==============================================================================
Output:
20 10
Akhtar Shekh
```

9. Write a C++ Program to Perform Simple Addition Function Using Templates.

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

template <class T>
T add(T a, T b)
{
    return (a + b);
}
```

```
int main()
{
    int x = 2;
    int y = 5;
    cout << x << " + " << y << " = " << add(x, y) << endl;
    float p = 2.5;
    float q = 8.5;
    cout << p << " + " << q << " = " << add(p, q) << endl;
    double r = 125.535;
    double s = 8485.958;
    cout << r << " + " << s << " = " << add(r, s) << endl;

    return 0;
}
=============================================================================
Output:
2 + 5 = 7
2.5 + 8.5 = 11
125.535 + 8485.96 = 8611.49
```

10. Write a C++ program to implement Hash Table using Template Class.

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

template <class T>
T Hash(T value)
{
    return (value % 10);
}

int main()
{
    int a[10], number, search, temp, j;
    cout << "How much elements you want to enter : ";
    cin >> number;

    cout << "Enter " << number << " numbers : ";
    for(int i = 0; i < number; i++)
    {
        cin >> temp;
        j = Hash(temp);
        a[j] = temp;
    }

    cout << "Search number : ";
    cin >> search;
    int temp1 = Hash(search);

    if(search == a[temp1])
        cout << "Value found";
    else
        cout << "Value not found";

    return 0;
}
=============================================================================
Output:
How much elements you want to enter : 8
Enter 8 numbers : 7 8 9 3 1 4 6 2
Search number : 1
Value found
```