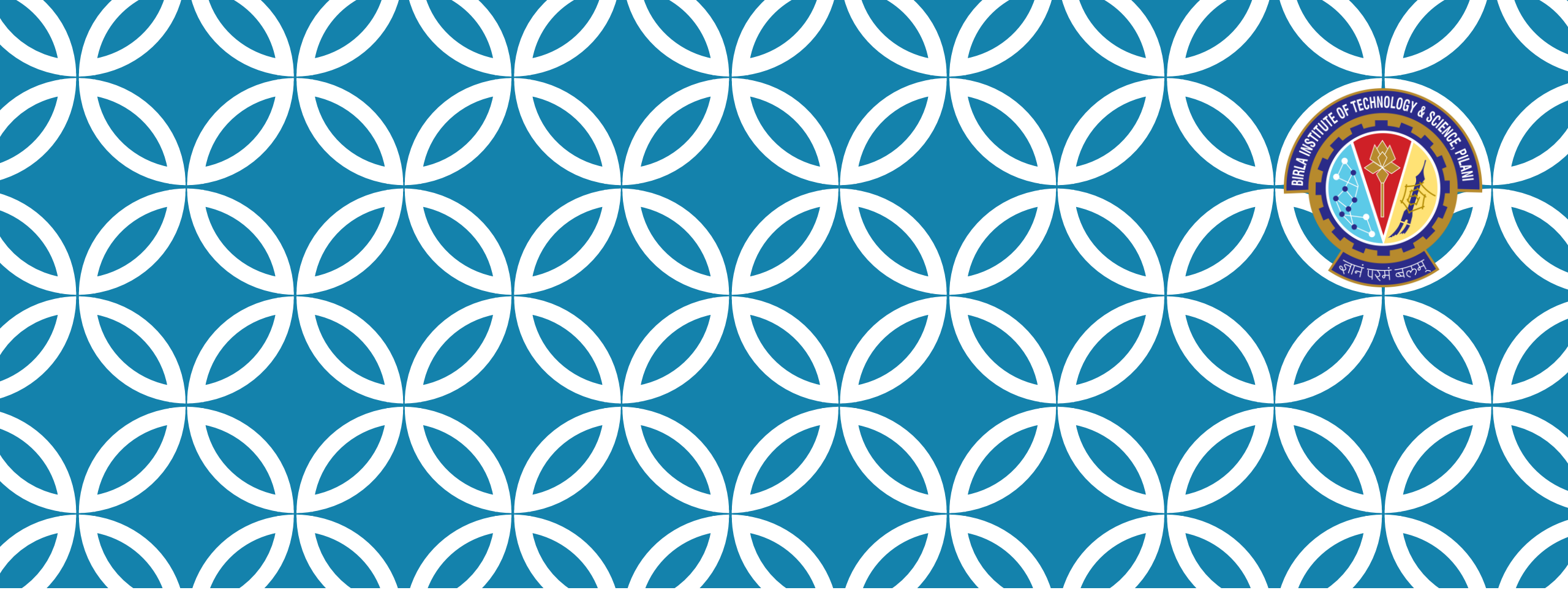

BITS F232: FOUNDATIONS OF DATA STRUCTURES & ALGORITHMS (1ST SEMESTER 2023-24) C++ CONTINUED…

Chittaranjan Hota, PhD
Sr. Professor of Computer Sc.
BITS-Pilani Hyderabad Campus
hota[AT]hyderabad.bits-pilani.ac.in

DESIGN PATTERNS: TEMPLATES IN C++

```cpp
#include <iostream>
#include <string>
using namespace std;
template <typename T>
inline T const& Maximum (T const& a, T const& b) {
   return a < b ? b:a;
}

int main () {
   int i = 46;
   int j = 357;
   cout << "Maximum integer:" << Maximum(i, j) << '\t';

   float p = 345.8;
   float q = 577.5;
   cout << "Maximum float:" << Maximum(p, q) << endl;

   string r = "BITS";
   string s = "Pilani";
   cout << "Larger string:" << Maximum(r, s) << endl;

   return 0;
}
```

```
Maximum integer:357 Maximum float:577.5
Larger string:Pilani
```

```cpp
#include <iostream>
using namespace std;
template <typename T>
class Array {
   private: T *ptr; int size;
   public:
      Array(T arr[], int s);
      void print();
};
template <typename T>
  Array <T>::Array (T arr[], int s) {
    ptr = new T[s];
    size = s;
    for(int i = 0; i < size; i++) ptr[i] = arr[i];
}
template <typename T>
  void Array<T>::print() {
    for (int i = 0; i < size; i++) cout<<" "<<*(ptr + i);
    cout<<endl;
}
int main() {
  int arr1 [5] = {10, 20, 30, 40, 50};
  double arr2 [5] = {3.5, 6.5, 7.2, 7.3, 7.9};
  Array <int> a (arr1, 5);
  Array <double> b (arr2, 5);
  a.print();
  b.print();
  return 0;
}
```

```
10 20 30 40 50
3.5 6.5 7.2 7.3 7.9
```

WHAT TYPE?

```cpp
#include <iostream>
#include <vector>
#include <cstdlib>
#include <string>
#include <stdexcept>
using namespace std;

template <class T>
class Stack {
   private:
      vector<T> elems;
   public:
      void push(T const&);
      void pop();
      T top() const;

      bool empty() const {
         return elems.empty();
      }
};

template <class T>
void Stack<T>::push (T const& elem) {
   elems.push_back(elem);
}

template <class T>
void Stack<T>::pop () {
   if (elems.empty()) {
      throw out_of_range("Stack<>::pop(
   }
   elems.pop_back();
}
```

```cpp
template <class T>
T Stack<T>::top () const {
   if (elems.empty()) {
      throw out_of_range("Stack<>::top(): empty stack");
   }
   return elems.back();
}

int main() {
   try {
      Stack<int>         intStack;
      Stack<string> stringStack;
      intStack.push(345);
      cout << intStack.top() <<endl;

      stringStack.push("BITS F232");
      cout << stringStack.top() << std::endl;
      stringStack.pop();
      stringStack.pop();
   } catch (exception const& ex) {
      cerr << "Exception case: " << ex.what() <<endl;
      return -1;
   }
}
```

```
345
BITS F232
Exception case: Stack<>::pop(): empty stack
```

STANDARD TEMPLATE LIBRARY (STL) IN C++

- A library of container classes, algorithms, and iterators.

Can you name some?

```
Size of the vector: 1
Expanded size: 4
Vaue of vector0:56.5
Vaue of vector1:57.5
Vaue of vector2:58.5
Vaue of vector3:59.5
Value through iterator= 56.5
Value through iterator= 57.5
Value through iterator= 58.5
Value through iterator= 59.5
```

$$\text{InitialValue} + \sum_{i=0}^{n-1} a[i]$$

How to do this using STLs?

Can you name some of the STL functions in this code?

STL on strings:
insert, append, swap, size, resize, reverse etc.

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  int main() {
5     vector <double> v;
6     int i;
7     v.push_back (56.5);
8     cout << "Size of the vector: " << v.size() << endl;
9     for(i = 1; i < 4; i++) {
10       v.push_back(v[0] + i);
11    }
12    cout << "Expanded size: " << v.size() << endl;
13
14    for(i = 0; i < 4; i++) {
15       cout <<"Vaue of vector"<<i<< ":"<< v[i] << endl;
16    }
17
18    vector<double>::iterator t = v.begin();
19    while( t != v.end()) {
20       cout << "Value through iterator= " << *t<< endl;
21       t++;
22    }
23    return 0;
24 }
```

ELEMENTARY DATA STRUCTURES: ARRAYS AND LINKED LISTS

ARRAYS: LINEAR DATA STRUCTURES

- What are Arrays?
- Can you give some examples?
- Why are they called linear data structures?

Applications of arrays: Maths (vectors, matrices, polynomials,…), databases, compilers (control flow), dynamic memory allocations etc.

THANK YOU!

Next Class: Dynamic Arrays, and Linked List…