

Heaven's Light is Our Guide  
Rajshahi University of Engineering & Technology  
Department of Computer Science & Engineering

## **Lab Manual**

Course Code: **CSE 1204 (Sec A)**  
Course Title: Sessional based on CSE 1203  
Instructor: Md. Shahid Uz Zaman  
Dept of CSE, RUET

## Module 5 [Advanced Topics]: (for Week 7)

### Topic 1[Exception Handling]

*Problem Statement:* In the following input *i* is the index of array *ax[]*. The program prints *ax[i]*. Then write catch block if *i* is out of range of *ax[]*. Write three catch blocks to fulfill the purpose

- i) a catch block receives the value of *i*
- ii) a catch block receives string "Out of Range Error"
- iii) a default catch() if above two catch block doesn't match

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

int main()
{
    int i;
    int ax[5]={10,20,60,40,30};
    cout<<"enter index:";
    cin>>i;
    cout<<"ax["<<i<<"]="<<ax[i]<<endl;
}
```

### Topic 2 [class Template]:

*Problem Statement:* In the following class data members *x* and *y* are integers and the method *Sum()* adds *x* and *y*. However, we need to perform the sum of *int+int*, *int+double*, *double+int* and *double+double*. To achieve it, change the definition of *x,y,setData()* and *Sum()* accordingly.

```
class A{
private:
    int x;
    int y;
public:
    void setData(int x,int y){
        this->x=x;
        this->y=y;
    }
    int Sum(){
        int s;
        s=x+y;
        return s;
    }
};

int main(){
    //write required statements to call SetData() & Sum()
}
```

### Topic 3 [STL:Array class]

Problem statement: Declare a STL array object ax with 6 elements and do the following:

- i) Assign 10,60,30,70,20 to ax using single statement
- ii) Print third element of ax using at() function
- iii) Print first element of ax using front() function
- iv) Print last element of ax using back() function
- v) Fill the elements of ax using fill() function
- vi) Test whether ax is empty or not using empty() function
- vii) Print size of ax
- viii) Print maximum size of ax using max\_size() function
- ix) Print address of first element of ax using begin() function
- x) Print address of last element of ax using end() function

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    array<int,6>ax;
    //write statements
}
```

### Topic 4[STL: pair class]

Problem statement: Define a pair class object px with int and string elements. Write statements to do the following

- i) Assign 10 to int and "Rajshahi" to px using make\_pair() function
- ii) Print int data member by first
- iii) Print string data member by second
- iv) Modify first data member to 20 using get<>() function
- v) Declare another pair bx and assign values to bx and swap it with ax

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    pair<int,string>px;
    //write statements
}
```

### Topic 5 [STL: tuple class]

Problem statement: Define a tuple class object tx with int.string and double elements. Write statements to do the following

- i) Assign <100,"Kamal",3.5> to tx using make\_tuple() function
- ii) Print int data member by get() function

- iii) Print string data member by get() function
- iv) Print double data member by get() function
- v) Modify third data member to 3.7 using get<>() function
- vi) Declare another tuple bx and assign values to bx and swap it with ax

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    tuple<int,string,double>tx;
    //write statements
}
```

### Topic 6 [STL: stack class]

Problem statement: Write a program to create and manipulate Stack using stack class and Perform the following operations using the specified method.

- i) use push() method to push data
- ii) use pop() method to pop data
- iii) use top() method to display top element
- iv) use empty() method to check whether stack is empty or not

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    stack<int>st;
    //write statements
}
```

### Topic 7 [STL: queue class]

Problem statement: Write a program to create and manipulate Queue using queue class. Perform the following operations using the specified method.

- i) use push() method to push data
- ii) use pop() method to pop data
- iii) use front() method to display front element
- iv) use back() method to display rear element
- v) use empty() method to check whether queue is empty or not

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    queue<int>qu;
```

```
//write statements  
}
```

### Topic 8 [STL: list class]

Problem statement: Write a program to create and manipulate linked list using `list` class and its following methods to

- i) insert 8 integers using **`push_back()`** method
- ii) insert two elements using **`push_front()`** method
- iii) display all the elements of the list in forward direction with user-defined **`Display()`** method using **`begin()`** and **`end()`** methods and iterator
- iv) display all the elements of the list in reverse direction with a user-defined **`DisplayRev()`** method using **`rbegin()`** and **`rend()`** method and iterator
- v) display front element using **`front()`** method
- vi) display back element using **`back()`** method
- vii) delete front element using **`pop_back()`** method
- viii) delete front element using **`pop_front()`** method
- ix) search an element **`x`** using **`find()`** method
- x) insert a new element **`x`** before an existing element **`y`** using **`insert()`** method
- xi) insert a new element **`x`** after an existing element **`y`** using **`insert()`** method
- xii) count a particular element **`x`**
- xiii) count a elements with condition using predicate function
- xiv) delete a particular element **`x`** with **`erase()`** method
- xv) delete first 4 elements with **`erase()`** method
- xvi) delete a particular element **`x`** with **`remove()`** method
- xvii) delete a elements with condition using **`remove_if()`** method using predicate function
- xviii) assign elements from another list using **`assign()`** method
- xix) assign elements from an array using **`assign()`** method
- xx) sort the list using **`sort()`** method
- xxi) Delete consecutive similar elements using **`unique()`** method

```
#include <iostream>  
#include <bits/stdc++.h>  
#include <iterator>  
#include <algorithm>  
using namespace std;  
  
int main(){  
    list<int>li;  
    //write statements  
}
```