KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE

A

Lab Report On

Object Oriented Programming (COMP116)

Lab Work No: 6

Submitted by:

Ashraya Kadel

CE I/II

Rollno: 25

Submitted to:

Rajani Chulyadyo

Department of Computer Science and Engineering

SUBMISSION DATE: 17 / 12 / 2023

(Q) We need to implement FIFO called Queue with four functions:
insert (adding element), remove (removing first inserted), front (peeking the first element), rear (peeking last element)

Create an interface class IQueue with the functionalities. Create a class ArrayQueue that inherits IQueue and stores data elements in array.

Ans:

(*) Source Code:

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

class IQueue
{
    public:
        virtual ~IQueue () {}
        virtual bool insert (int element) = 0;
        virtual bool remove (int &element) = 0;
        virtual bool front (int & element) = 0;
        virtual bool rear (int &element) = 0; };

class ArrayQueue : public IQueue
{
    private:
        int topindex;
        int size;
        int *data;
```

```
public:
ArrayQueue (int size): topindex(-1), size(size),
                       data(new int[size]) {}
bool insert (int element)
{  if  (topindex < size-1)
   {  topindex++;
      data[topindex] = element;
      return ~~tt~~ true;
   }
   else
   {  throw  runtime_error ("The queue is full, Please
remove before any insertion");

bool remove (int &element)
{ ~~int~~ if (topindex >=0)
 { int i;
   for (i=0; i<topindex; i++)
   { int temp;
      temp = data [i+1];
      data[i] = data [i+1];
   }
   topindex --;
   return true;
 }
  else {
  throw runtime-error ("No element in queue");
   return false;
}
```

```
bool front (int &element)
{ if (topindex>=0)
  { element = data[0];
     return true; }
 else
   throw runtime-error ("No element in Queue!");
   return ~~type~~ false;
 }

bool rear (int &element)
{ if  (topindex >=0)
  { element = data[topindex];
     return true; }
   else {
     throw runtime_error ("No element in Queue");
     return false;}
 };

int main()
{
  IQueue *I = new ArrayQueue (10);
  I->insert(5); I->insert(6); int element;
  I->rear(element);
  cout<< "Last element is" << element<< endl;
   I-> front (element);
  cout<< "Front element is" << element <<endl;
}
```