**National University of Computer and Emerging Sciences**



**Lab Manual 06**

**Object Oriented Programming**

|  |  |
| --- | --- |
| Course Instructor | Ms. Arooj Khalil |
| Lab Instructor (s) | Ms. Fariha Maqbool  Ms. Amara Nasir |
| Section | BSE-2A |
| Semester | Spring 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to:

* Use \*this pointer
* Use of dynamic arrays in objects
* Copy Constructors
* Operator Overloading

**TASK-1:**

Create a class which has following data members:

1. array ptr (pointer to a dynamic array)
2. array size

and implement the following:

1. Write a default constructor and initialize big\_int\_ to nullptr.

* ArrayInt();

1. Write an overloaded constructor and perform deep copy.

* ArrayInt (const int \* arr, int size);

1. Write a copy constructor and perform deep copy. Print “Copy Constructor Called” and observer the scenarios where the copy constructor is called.

* ArrayInt (const ArrayInt &obj);

1. Destructors
2. Getters and setters
3. Display Function (To display the elements of array)

Moreover, you have to overload the following overloaded operators:

1. []

This operator returns an element in the array only if the index is not out of bounds.

1. Pre and Post increment

Increase value of all elements of array by an integer **n**.

1. Pre and Post decrement

Decrease value of all elements in the array by an integer **n**.

Main function will be as following:

int main()  
{  
   ClassName obj1, obj2;  
 // Set the elements in array of obj1 after taking input from user

    cout<<"Enter value of n : ";  
    cin>>n;  
    obj2=obj1 + n; //Use overloaded operator to increment the values of elements

    cout<<"After incrementing all element in array by "<<n<<" : ";  
    for(int i=0;i<num;i++)  
        cout<<obj2[i]<<" "; //Get the element using overloaded operator []  
    return 0;  
}

**TASK-2:**

Implement a class called **Complex**. The Complex class will have two data members:

* int real; // The real part of complex number
* int imaginary; // Imaginary part of the complex number.

You have to implement default constructor, overloaded constructor, copy constructor, destructor and overload the operators **+, - , << , >>, ==, !=,=**

**Sample Run:**

|  |  |
| --- | --- |
| **Driver.cpp** | **Output** |
| int main()  {  Complex C1;  Complex C2(5,6);  Complex C3;  cout<<"Input a complex number"<<endl;  cin>>C3;  cout<<C1<<C2<<C3;  if(C1==C2)  cout<<"C1 == C2"<<endl;  else  cout<<"C1 != C2"<<endl;  if(C1!=C3)  cout<<"C1 != C3"<<endl;  else  cout<<"C1==C3"<<endl;  Complex C4= C2 - C3;  cout<<C4;  system("pause");  return 0;  } |  |