1. Write a class Array which will store integer elements in dynamically allocated space.

class Array

{

int size;

int \*arr;

}

Provide below functions in Array class -

1. Default constructor - Take 5 as default size. Initialize the array elements to 0

2. Parameterized constructor - Takes size as parameter. Initialize the array elements to 0.

3. Copy constructor - Performs deep copy of Array object.

4. Destructor - Deletes memory which is allocated dynamically.

1. Modify class Array in assignment – 41 to overload below operators

1. Overloaded [] operator - used to get/set array element e.g. arr[4] gives 4th index element

2. Overloaded ! operator - reverses array elements.

3. Overload >> - Accepts array size from user. Also accepts elements of array from user.

4. Overload << - Displays array.

1. Write a class Matrix which will store elements of integer matrix (two dimensional array) in dynamically allocated space. Provide below functions in matrix class -

1. Default constructor - Take 3 as default row and column size. Initialize the matrix elements to 0

2. Parameterized constructor - Takes row and column size as parameters. Initialize matrix to 0.

3. Copy constructor - Performs deep copy of matrix object.

4. Destructor - Deletes memory which is allocated dynamically.

5. Overloaded + operator - Performs addition of two matrices.

6. Overloaded ! operator - Calculates transpose of a user entered square matrix.

7. Overload >> - Accepts row & column size from user. Also accepts matrix elements from user.

8. Overload << - Displays matrix.

1. Create objects c1,c2,c3,c4 of class Complex implemented earlier. Overload necessary operators to execute following statements
   1. c3=c1+c2
   2. c3=c1-c2
   3. c2= -c1
   4. c4= ++c1
   5. c4=c1++
2. Define a class “Book” having members – bookId, title, price. Maintain count of books. Accept and display details of five books.
3. Write a class Employee for an application which will have data members for employee id, employee name and salary. Provide the following functionalities in Employee class.

1. Initialing objects using default and parameterized constructors.

2. Accepting and displaying the information of employee from console

class Employee

{

int emp\_id;

char emp\_name[20];

double salary;

public:

Employee();

Employee(int, char\*, double);

void Accept();

void Display();

};

In "main" function declare an array of Employee objects. Write a menu driven program to -

a. Insert record into an array.

b. Update information of specific employee on the basis of emp\_id accepted from user

c. Display all records.