

OBJECT ORIENTED PROGRAMMING PROJECT

SEMESTER-III

**PRESENTED BY:**

**NAKSH ARORA(14103001)**

**ADITI BHARDWAJ (14103108)**

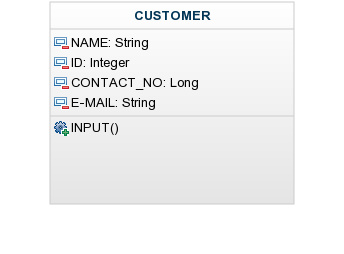
**SIMRAN CHHABRA(14103020)**

**NIKITA JAIN (14103010)**

**PRATEEK AGARWAL (14103028)**

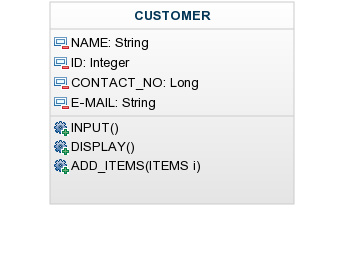
## INCREMENT 1

## SINGLE APPLICATION, SINGLE CLASS, SINGLE METHOD



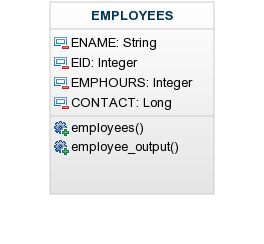
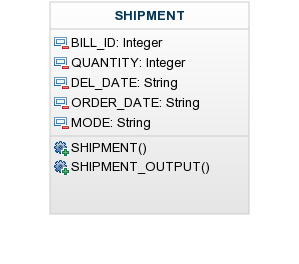
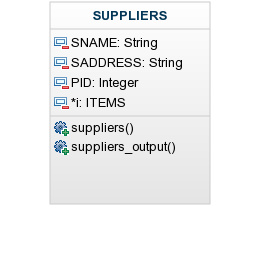
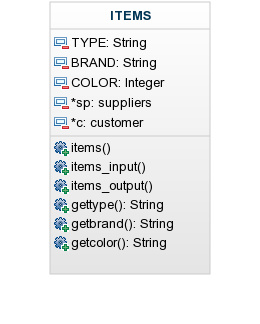
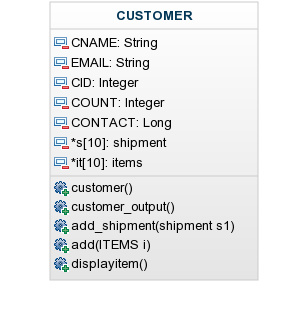
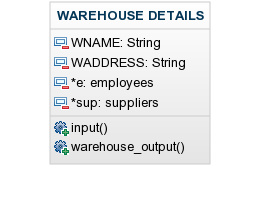
## INCREMENT 2

## SINGLE APPLICATION, SINGLE CLASS, MULTIPLE METHOD



# INCREMENT 3

# SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, SIMPLE RELATIONSHIP

## INCREMENT 4

## SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, COMPLEX RELATIONSHIP

## INCREMENT 5

## SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, AGGREGIATION/COMPOSITE ASSOCIATION

## INCREMENT 6

## Single Application, Multiple Class, Multiple Method, Inheritance (including Friend Functions)

## INCREMENT 7

## Single Application, Multiple Class, Multiple Method, Polymorphism (including Overloading and Overriding)

## INCREMENT 8

**TEMPLATES:**

//bubble sort to sort suppliers according to their pids

template<class T>

void suppsort(T a[], int n)

{

int i, j;

for(i=0;i<n-1;i++)

{

for(j=i+1;j<n;j++)

{

if(a[i]>a[j])

{

T element;

element = a[i];

a[i] = a[j];

a[j] = element;

}

}

}

for(int i=0;i<n;i++)

cout<<a[i]<<" ";

cout<<endl;

}

## INCREMENT 9

1. //FUNCTION TO CHECK THE VALIDITY OF THE TYPE OF THE ITEM

void checkname(string type)

{

string f=type;

int flag=0;

try

{

int l=f.length();

for(int i=0;i<l;i++)

if (((f[i] >= 'a') && (f[i] <= 'z')) || ((f[i] >= 'A') && (f[i] <= 'Z')))//CHECKING WHETHER THE TYPE OF ITEM CONATINS A STRING OF LOWER CASE LETTERS OR A STRING OF UPPER CASE

flag=1;

else

throw "enter a valid name with characters only";//IF THE TYPE OF ITEM CONTAINS ONLY STRINGS WITH LOWER CASE LETTERS OR A STRING OF UPPER CASE THEN THIS MESSAGE IS PRINTED

if(flag)

cout<<"you entered a valid name"<<endl;

}

catch(const char \*msg)

{

cerr<<msg<<endl;

}

}