

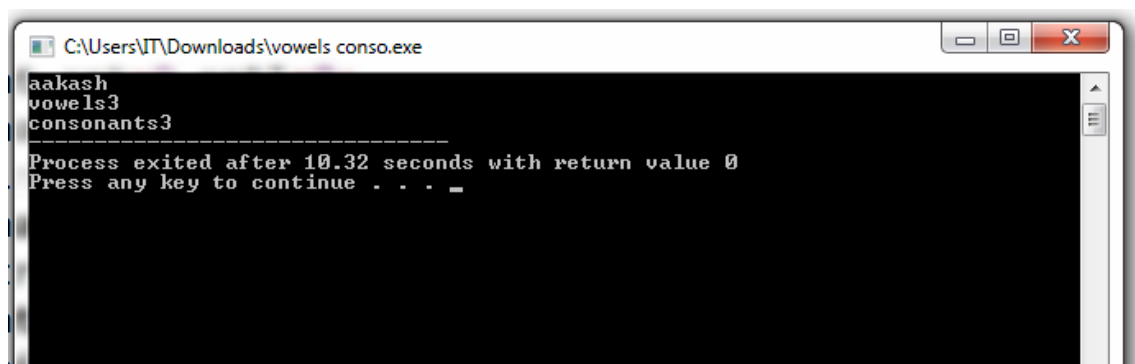
## EXPERIMENT - 7

**AIM:** WAP to count vowels and consonants in a string using pointer.

**Source Code:**

```
#include<iostream>
using namespace std;
int main()
{
    int cnt=0,cnt1=0;
    char a[1000];
    cin>>a;
    char*ptr;
    ptr=a;
    int i=0;
    while(*(ptr+i)!='\0')
    {
        if(*(ptr+i)=='a' || *(ptr+i)=='e' || *(ptr+i)=='i' || *(ptr+i)=='o' || *(ptr+i)=='u')
            cnt++;
        else cnt1++;
        i++;
    }
    cout<<"vowels"<<cnt<<endl;
    cout<<"consonants"<<cnt1;
}
```

**OUTPUT:**



```
C:\Users\IT\Downloads\vowels conso.exe
aakash
vowels3
consonants3
-----
Process exited after 10.32 seconds with return value 0
Press any key to continue . . . _
```

## EXPERIMENT - 8

**AIM:** WAP to find the sum if array elements using dynamic memory location.

**Source Code:**

```
package array.java;

import java.util.Scanner;

class SumDemo{

    public static void main(String args[]){

        Scanner scanner = new Scanner(System.in);

        int[] array = new int[10];

        int sum = 0;

        System.out.println("Enter the elements:");

        for (inti=0; i<10; i++)

            {

                array[i] = scanner.nextInt();

            }

        for(intnum : array) {

            sum = sum+num;

        }System.out.println("Sum of array elements is:"+sum);

    }

}
```

**OUTPUT:**



```
ADITI@C071A
Enter the elements:
1
2
3
4
5
6
7
8
9
10
Sum of array elements is:55
BUILD SUCCESSFUL (total time: 13 seconds)
```


## EXPERIMENT-9

**AIM:** WAP to implement concepts of encapsulation in JAVA/C#.

**Source Code:**

```
// c++ program to explain Encapsulation
#include<iostream>
using namespace std;
class Test
{
    private:    // data hidden from outside world
    int x;
    public:
        // function to set value of
        // variable x
    void set(int a)
    {
        x =a;
    }
        // function to return value of
        // variable x
    int get()
    { return x; }
};
int main()
{
    Test obj;
    obj.set(5);
    cout<<"value setted : "<<obj.get();
    return 0;
}
```

**OUTPUT:**

 C:\Users\Asus Prime\OneDrive\Documents\c++ programs\encapsulation.exe

```
value setted : 5
-----
Process exited after 0.05677 seconds with return value 0
Press any key to continue . . .
```

## EXPERIMENT - 10

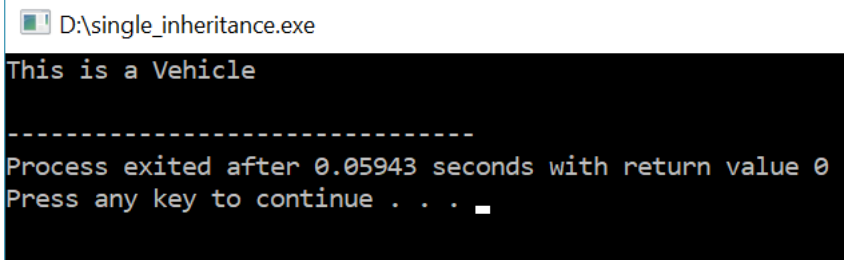
**AIM:** WAP to implement concepts of inheritance in JAVA/C#.

**SOURCE CODE:**

**IMPLEMENTATION OF SINGLE INHERITANCE**

```
#include <iostream>
using namespace std;
// base class
class Vehicle {
public:
    Vehicle()
    {
        cout<< "This is a Vehicle" <<endl;
    }
};
// sub class derived from two base classes
class Car: public Vehicle{ };
// main function
int main()
{
    // creating object of sub class will
    // invoke the constructor of base classes
    Car obj;
    return 0;
}
```

**OUTPUT:-**



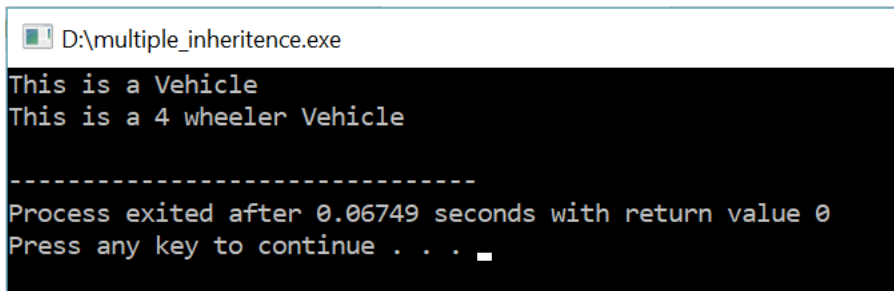
```
D:\single_inheritance.exe
This is a Vehicle
-----
Process exited after 0.05943 seconds with return value 0
Press any key to continue . . .
```

**IMPLEMENTATION OF MULTIPLE INHERITANCE**

```
// C++ program to explain multiple inheritance
#include <iostream>
using namespace std;
class Vehicle {
public:
    Vehicle()
    {
        cout<< "This is a Vehicle" <<endl;
    }
}
```

```
    }  
};  
class FourWheeler  
{  
public:  
    FourWheeler()  
    {  
        cout<< "This is a 4 wheeler Vehicle" <<endl;  
    }  
};  
class Car: public Vehicle, public FourWheeler{};  
int main()  
{  
    // creating object of sub class will  
    // invoke the constructor of base classes  
    Car obj;  
    return 0;  
}
```

### **OUTPUT:-**

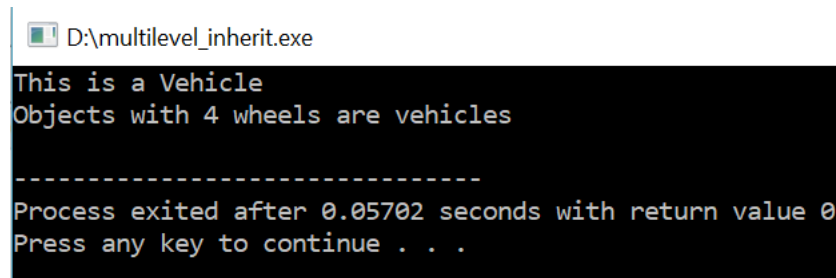


### **IMPLEMENTATION OF MULTILEVEL INHERITANCE**

```
// C++ program to implement Multilevel Inheritance  
#include <iostream>  
using namespace std;  
class Vehicle  
{  
public:  
    Vehicle()  
    {  
        cout<< "This is a Vehicle" <<endl;  
    }  
};  
class fourWheeler: public Vehicle  
{ public:  
    fourWheeler()  
    {  
        cout<<"Objects with 4 wheels are vehicles"<<endl;  
    }  
};  
// sub class derived from two base classes
```

```
class Car: public fourWheeler
{
public:
    car()
    {
        cout<<"Car has 4 Wheels"<<endl;
    }
};
int main()
{
    Car obj;
    return 0;
}
```

### **OUTPUT:-**

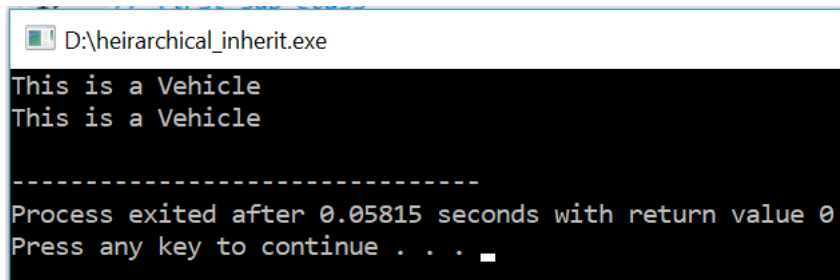


```
D:\multilevel_inherit.exe
This is a Vehicle
Objects with 4 wheels are vehicles
-----
Process exited after 0.05702 seconds with return value 0
Press any key to continue . . .
```

### **IMPLEMENTATION OF HIERARCHICAL INHERITANCE**

```
// C++ program to implement Hierarchical Inheritance
#include <iostream>
using namespace std;
class Vehicle
{
public:
    Vehicle()
    {
        cout<< "This is a Vehicle"<<endl;
    }
};
class Car: public Vehicle {};
class Bus: public Vehicle {};
int main()
{
    Car obj1;
    Bus obj2;
    return 0;
}
```

### **OUTPUT:-**



```
D:\heirarchical_inherit.exe
This is a Vehicle
This is a Vehicle

-----
Process exited after 0.05815 seconds with return value 0
Press any key to continue . . .
```

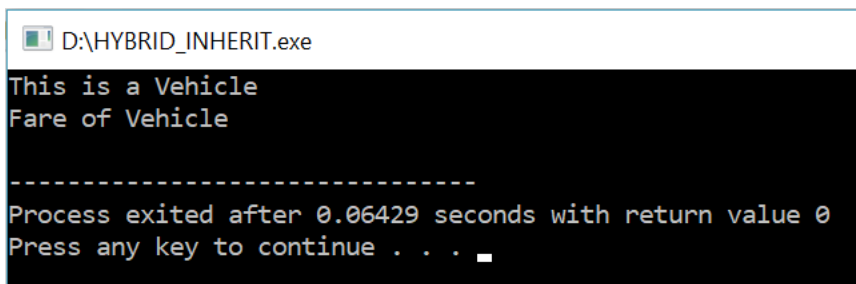
### **IMPLEMENTATION OF HYBRID INHERITANCE**

```
// C++ program for Hybrid Inheritance
#include <iostream>
using namespace std;
class Vehicle
{
public:
Vehicle()

{   cout<< "This is a Vehicle" <<endl; }

};
class Fare
{   public:
Fare()
{
cout<<"Fare of Vehicle\n";
}
};
class Car: public Vehicle {};
class Bus: public Vehicle, public Fare {};
int main()
{
    Bus obj2;
return 0;
}
```

### **OUTPUT:-**



```
D:\HYBRID_INHERIT.exe
This is a Vehicle
Fare of Vehicle

-----
Process exited after 0.06429 seconds with return value 0
Press any key to continue . . .
```

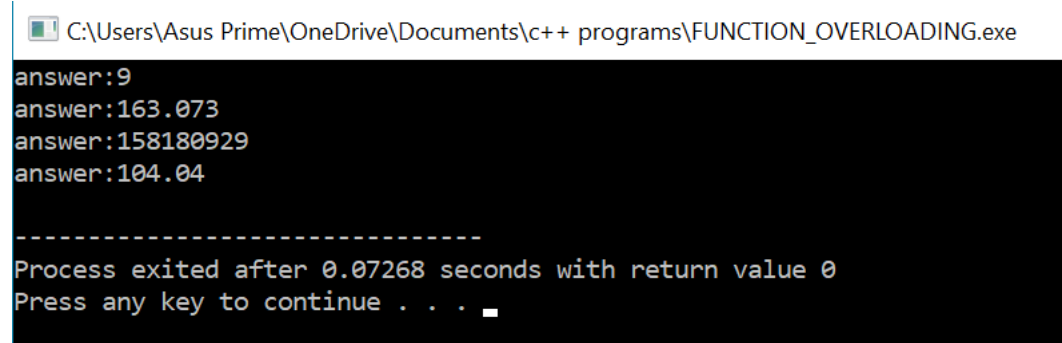
## EXPERIMENT - 11

**AIM:** WAP to implement concepts of polymorphism in JAVA/C#.

### IMPLEMENTATION OF FUNCTION OVERLOADING

#### SOURCE CODE:-

```
#include<iostream>
using namespace std;
class Test
{
public:
int square(int a) {cout<<"answer:"<<a*a<<endl;}
double square(double a) {cout<<"answer:"<<a*a<<endl;}
long square(long a) { cout<<"answer:"<<a*a<<endl;}
float square(float a) {cout<<"answer:"<<a*a<<endl;}
};
int main()
{
    Test obj;
    obj.square(3);
    obj.square(12.77);
    obj.square(12577);
    obj.square(10.2);
    return 0;
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\Users\Asus Prime\OneDrive\Documents\c++ programs\FUNCTION\_OVERLOADING.exe". The output of the program is as follows:

```
answer:9
answer:163.073
answer:158180929
answer:104.04


-----
Process exited after 0.07268 seconds with return value 0
Press any key to continue . . .
```

### IMPLEMENTATION OF OPERATOR OVERLOADING (PRE AND POST ++ )OPERATOR )

```
#include <iostream>
using namespace std;
class Test
{
private:
intnum;
public:
Test(): num(8){ }
void operator ++()
```



```
{
num = num+2;
}
void operator ++(int)
{
num=num+3;
}
void Print() {
cout<<"The Count is: "<<num;
}
};
int main()
{
    Test tt;
    ++tt; // calling of a function "void operator ++()"
    tt.Print();
    cout<<endl;
    tt++;
    tt.Print();
    return 0;
}
```

 C:\Users\Asus Prime\OneDrive\Documents\c++ programs\++OPERATOR\_OVERLOADING.exe


```
The Count is: 10
The Count is: 13
-----
Process exited after 0.08805 seconds with return value 0
Press any key to continue . . .
```

### IMPLEMENTATION OF OPERATOR OVERLOADING (+ OPERATOR )

```
#include<iostream>
using namespace std;
class Complex
{
private:
int real, imag;
public:
Complex(int r = 0, inti =0)
{
    real = r;
    imag = i;
}
// This is automatically called when '+' is used with between two Complex objects
Complex operator + (Complex const&obj)
{
    Complex r;
    r.real = real + obj.real;
    r.imag = imag + obj.imag;
    return r;
}
void print()
```

```
        {
            cout<< real << " + i" <<imag<<endl;
        }
    };
int main()
{
    Complex c1(10, 5), c2(2, 4);
    Complex c3 = c1 + c2; // call to "operator+"
    c3.print();
}
```

#### **OUTPUT:-**


 C:\Users\Asus Prime\OneDrive\Documents\c++ programs\+operator\_overloading.exe

```
12 + i9
-----
Process exited after 0.06778 seconds with return value 0
Press any key to continue . . .
```

#### **IMPLEMENTATION OF OPERATOR OVERLOADING (--OPERATOR )**

```
#include <iostream>
using namespace std;
class Test
{
private:
    intnum;
public:
    Test(): num(8){ }
    void operator --() { num = num-2; }
    void Print() { cout<<"The Count is: "<<num; }
};
int main()
{
    Test tt;
    --tt; // calling of a function "void operator --()"
    tt.Print();
    return 0;
}
```

#### **OUTPUT:-**


 C:\Users\Asus Prime\OneDrive\Documents\c++ programs\--operator\_overlading.exe

```
The Count is: 6
-----
Process exited after 0.06722 seconds with return value 0
Press any key to continue . . .
```

### IMPLEMENTATION OF OVERRIDING

```
#include<iostream>
using namespace std;
class Base
{ public:
inta,b;
void input()
{   cout<<" base class display,enter the values:\t";
    cin>>a>>b;
}
void sum()
{ int s=a+b;
cout<<"\nBase class\t";
cout<<"sum="<<s;
}
};
classDerived:public Base
{ public:
inta,b,c;
void input()
{   cout<<"\nDerived class display,enter values:\t";
    cin>>a>>b>>c; }
void sum()
{cout<<"\nDerived Class\n";
int s=a+b+c;
cout<<"SUM="<<s;}
};
int main()
{ Base b;    //Base class object
  Derived d;
b.input();  //Derived class object
b.sum();
d.input();
d.sum();
}
```

### OUTPUT:-

 C:\Users\Asus Prime\OneDrive\Documents\c++ programs\overriding.exe

```
base class display
enter the values:12 13
Base class
sum=25
Derived class display
enter values:12 13 14
Derived Class
SUM=39
-----
Process exited after 54 seconds with return value 0
Press any key to continue . . .
```

## EXPERIMENT - 12

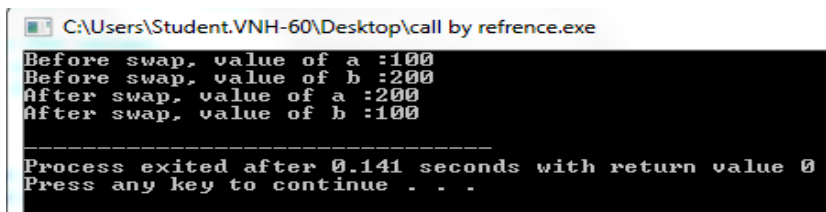
**AIM:** WAP in C++ to implement different parameter passing methods.

### Implementation of call by address

```
#include <iostream>
using namespace std;
void swap(int *x, int *y);
int main () {
    // local variable declaration:
    int a = 100;
    int b = 200;

    cout<< "Before swap, value of a : " << a <<endl;
    cout<< "Before swap, value of b : " << b <<endl;
    swap(&a, &b);
    cout<< "After swap, value of a : " << a <<endl;
    cout<< "After swap, value of b : " << b <<endl;
    return 0;
}
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    return;
}
```

### OUTPUT:-



### Implementation of call by reference

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

// function declaration
void swap(int&x, int&y);

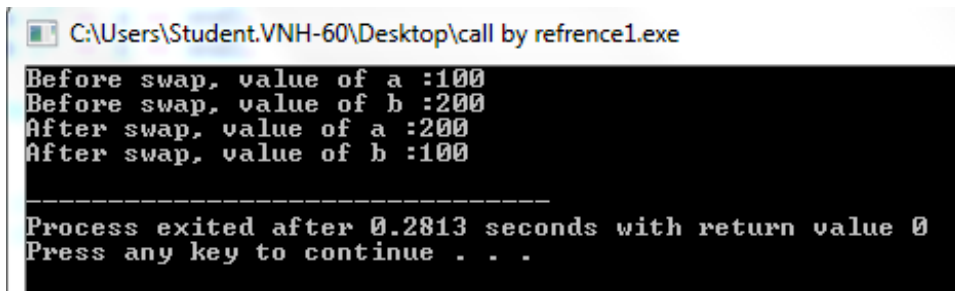
int main () {
    // local variable declaration:
```

```
int a = 100;
int b = 200;

cout<< "Before swap, value of a :" << a <<endl;
cout<< "Before swap, value of b :" << b <<endl;
swap(a, b);
cout<< "After swap, value of a :" << a <<endl;
cout<< "After swap, value of b :" << b <<endl;

return 0;
}
void swap(int&x, int&y) {
int temp;
temp = x;
  x = y;
  y = temp;
return;
}
```

### **OUTPUT:-**



```
C:\Users\Student.VNH-60\Desktop\call by refrence1.exe
Before swap, value of a :100
Before swap, value of b :200
After swap, value of a :200
After swap, value of b :100
-----
Process exited after 0.2813 seconds with return value 0
Press any key to continue . . .
```

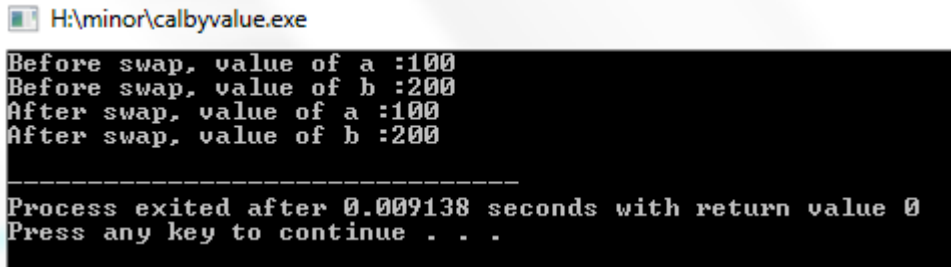
### **Implementation of call by value**

```
#include <iostream>
using namespace std;
void swap(int x, int y)
{
int temp;
temp = x;
  x = y;
  y = temp;
return;
}

int main () {
  // local variable declaration:
int a = 100;
int b = 200;
cout<< "Before swap, value of a :" << a <<endl;
cout<< "Before swap, value of b :" << b <<endl;
  // calling a function to swap the values.
swap(a, b);
cout<< "After swap, value of a :" << a <<endl;
```

```
cout<< "After swap, value of b : " << b <<endl;  
return 0;  
}
```

### OUTPUT:-

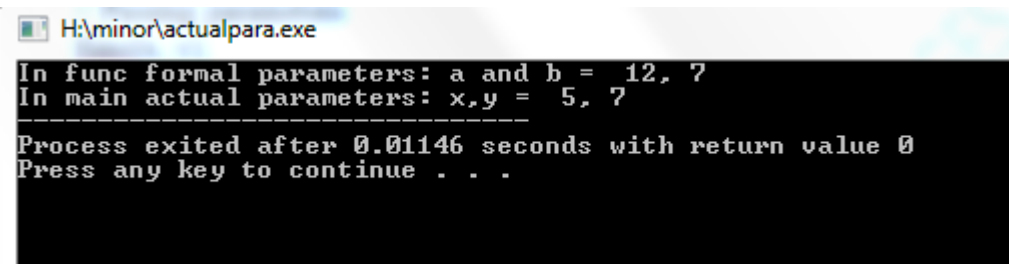


```
H:\minor\calbyvalue.exe  
Before swap, value of a :100  
Before swap, value of b :200  
After swap, value of a :100  
After swap, value of b :200  
-----  
Process exited after 0.009138 seconds with return value 0  
Press any key to continue . . .
```

### Implementation of actual and formal parameters

```
#include<iostream>  
using namespace std;  
  
voidfunc(int a, int b)  
{  
    a += b;  
    cout<<"In func formal parameters: a and b = " << a << ", " << b;  
}  
int main()  
{  
    int x = 5, y = 7;  
    func(x, y);  
    cout<<"\nIn main actual parameters: x,y = " << x << ", " << y;  
    return 0;  
}
```

### OUTPUT:-



```
H:\minor\actualpara.exe  
In func formal parameters: a and b = 12, 7  
In main actual parameters: x,y = 5, 7  
-----  
Process exited after 0.01146 seconds with return value 0  
Press any key to continue . . .
```

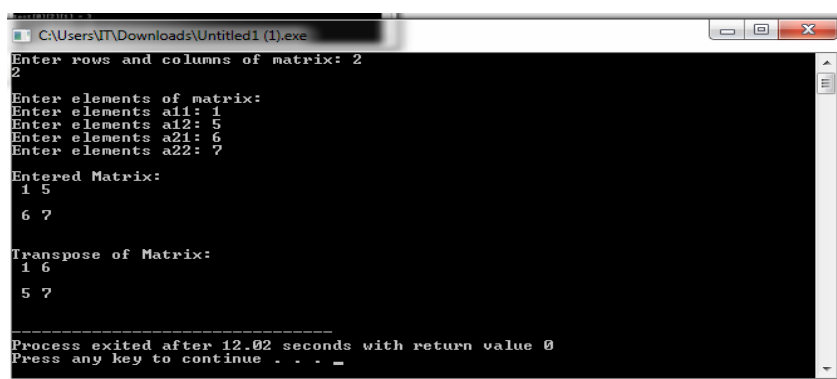
## EXPERIMENT - 13

**AIM:** WAP to find transpose of a matrix.

**Source Code:**

```
#include <iostream>
using namespace std;
int main()
{
    int a[10][10], trans[10][10], r, c, i, j;
    cout << "Enter rows and columns of matrix: ";
    cin >> r >> c;
    cout << endl << "Enter elements of matrix: " << endl;
    for(i = 0; i < r; ++i)
    for(j = 0; j < c; ++j)
    {
        cout << "Enter elements a" << i + 1 << j + 1 << ": ";
        cin >> a[i][j];
    } cout << endl << "Entered Matrix: " << endl;
    for(i = 0; i < r; ++i){
        for(j = 0; j < c; ++j)
        {cout << " " << a[i][j];
            if(j == c - 1)
                cout << endl << endl } }
    for(i = 0; i < r; ++i){
        for(j = 0; j < c; ++j)
            {trans[j][i]=a[i][j]:}}
    cout << endl << "Transpose of Matrix: " << endl;
    for(i = 0; i < c; ++i)
    { for(j = 0; j < r; ++j)
        {
            cout << " " << trans[i][j];
            if(j == r - 1)
                cout << endl << endl;
        } }return 0;}
```

Output:



```
C:\Users\VT\Downloads\Untitled1 (1).exe
Enter rows and columns of matrix: 2
2
Enter elements of matrix:
Enter elements a11: 1
Enter elements a12: 5
Enter elements a21: 6
Enter elements a22: 7
Entered Matrix:
1 5
6 7
Transpose of Matrix:
1 6
5 7
-----
Process exited after 12.02 seconds with return value 0
Press any key to continue . . .
```

## EXPERIMENT - 14

**AIM:** WAP in java to implement concurrent execution of a job using threads.

**Source Code:**

```
class NewThread extends Thread
{
    public void run()
    {
        for(int i=0;i<=3;i++)
            System.out.println(Thread.currentThread().getName());
    }
}

public class NewClass1
{
    public static void main(String [] args)
    {
        NewThread t1=new NewThread();
        t1.setName("t1");
        NewThread t2=new NewThread();
        t2.setName("t2");
        NewThread t3=new NewThread();
        t3.setName("t3");
        t1.start();
        t2.start();
        t3.start();
    }
}
```

**Output:**

```
t1
t2
t1
t3
t2
t3
t3
t1
t2
```



## EXPERIMENT - 15

**AIM:** WAP that demonstrates the use of THIS pointer.

### Source Code:

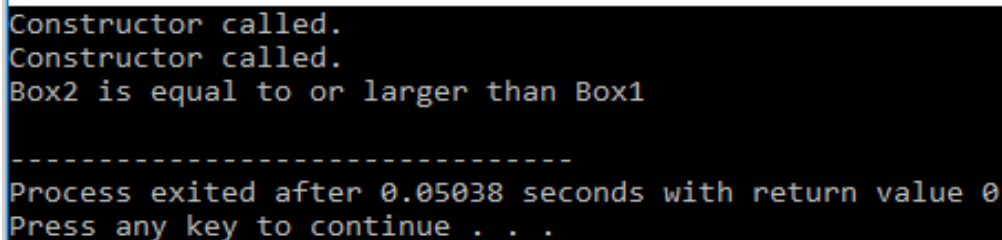
```
#include <iostream>
using namespace std;
class Box {
public:
    // Constructor definition
    Box(double l = 2.0, double b = 2.0, double h = 2.0) {
        cout << "Constructor called." << endl;
        length = l;
        breadth = b;
        height = h;
    }
    double Volume() {
        return length * breadth * height;
    }
    int compare(Box box) {
        return this->Volume() > box.Volume();
    }

private:
    double length;    // Length of a box
    double breadth;   // Breadth of a box
    double height;    // Height of a box
};

int main(void) {
    Box Box1(3.3, 1.2, 1.5);    // Declare box1
    Box Box2(8.5, 6.0, 2.0);    // Declare box2

    if(Box1.compare(Box2)) {
        cout << "Box2 is smaller than Box1" << endl;
    } else {
        cout << "Box2 is equal to or larger than Box1" << endl;
    }

    return 0;
}
```

A screenshot of a terminal window showing the output of the C++ program. The output consists of three lines: "Constructor called.", "Constructor called.", and "Box2 is equal to or larger than Box1". Below these lines is a dashed line separator, followed by the text "Process exited after 0.05038 seconds with return value 0" and "Press any key to continue . . .".

```
Constructor called.
Constructor called.
Box2 is equal to or larger than Box1

-----
Process exited after 0.05038 seconds with return value 0
Press any key to continue . . .
```