class Book {

    String author, title, publisher;

    public Book(String a, String t, String p) {

        author = a;

        title = t;

        publisher = p;

    }

    public void show()  {

        System.out.println("\nAuthor : " + author);

        System.out.println("Title : " + title);

        System.out.println("Publisher: " + publisher);

    }

}

class BookInfo extends Book {

    float price;

    int stock;

    public BookInfo(String a, String t, String p, float pr, int st) {

        super(a, t, p);

        price = pr;

        stock = st;

    }

    public void show() {

        super.show();

        System.out.println("Price: " + price);

        System.out.println("Stock: " + stock);

    }

}

public class MyBookInfo {

    public static void main(String[] args) {

        BookInfo b1 = new BookInfo("Galvin", "OS Concepts", "Unknown", 450, 57);

        b1.show();

    }

}

**C:\Users\3atha\code\study\oosw\expt-5\single**

**λ** javac MyBookInfo.java

**C:\Users\3atha\code\study\oosw\expt-5\single**

**λ** java MyBookInfo

Author : Galvin

Title : OS Concepts

Publisher: Unknown

Price: 450.0

Stock: 57

class Living {

public void display() {

System.out.println("I am alive!");

}

}

class Animal extends Living {

public void display() {

super.display();

System.out.println("I can walk!");

}

}

class Bird extends Animal {

public void display() {

super.display();

System.out.println("I can fly!");

}

}

public class MultilevelDemo {

public static void main(String[] args) {

Bird parrot = new Bird();

parrot.display();

}

}

**C:\Users\3atha\code\study\oosw\expt-5\multilevel**

**λ** javac MultilevelDemo.java

**C:\Users\3atha\code\study\oosw\expt-5\multilevel**

**λ** java MultilevelDemo

I am alive!

I can walk!

I can fly!

class Living {

public void display() {

System.out.println("I am alive!");

}

}

class Animal extends Living {

public void display() {

super.display();

System.out.println("I can move!");

}

}

class Plant extends Living {

public void display() {

super.display();

System.out.println("I have leaves!");

}

}

public class HierarchicalDemo {

public static void main(String[] args) {

Animal dog = new Animal();

dog.display();

Plant oak = new Plant();

oak.display();

}

}

**C:\Users\3atha\code\study\oosw\expt-5\hierarchical**

**λ** javac HierarchicalDemo.java

**C:\Users\3atha\code\study\oosw\expt-5\hierarchical**

**λ** java HierarchicalDemo

I am alive!

I can move!

I am alive!

I have leaves!

#include <iostream>

using namespace std;

class Terrestial

{

public:

void display()

{

cout << "We live on land!" << endl;

}

};

class Aquatic

{

public:

void display()

{

cout << "We live in water!" << endl;

}

};

class Amphibian: public Terrestial, public Aquatic

{

public:

void display()

{

Terrestial::display();

Aquatic::display();

cout << "We are amphibians!" << endl;

}

};

int main()

{

Amphibian frog;

frog.display();

return 0;

}

**C:\Users\3atha\code\study\oosw\expt-5\multiple**

**λ** g++ MultipleDemo.cpp -o MultipleDemo.exe

**C:\Users\3atha\code\study\oosw\expt-5\multiple**

**λ** MultipleDemo.exe

We live on land!

We live in water!

We are amphibians!

#include <iostream>

using namespace std;

class Animal

{

public:

void display()

{

cout << "We can move!" << endl;

}

};

class Mammal: virtual public Animal

{

public:

void display()

{

cout << "We have mammary glands!" << endl;

}

};

class Bird: virtual public Animal

{

public:

void display()

{

cout << "We can fly!" << endl;

}

};

class Bat: public Mammal, public Bird

{

public:

void display()

{

Animal::display();

Mammal::display();

Bird::display();

cout << "We are bats!" << endl;

}

};

int main()

{

Bat b;

b.display();

return 0;

}

**C:\Users\3atha\code\study\oosw\expt-5\hybrid**

**λ** g++ HybridDemo.cpp -o HybridDemo.exe

**C:\Users\3atha\code\study\oosw\expt-5\hybrid**

**λ** HybridDemo.exe

We can move!

We have mammary glands!

We can fly!

We are bats!