Example : 1

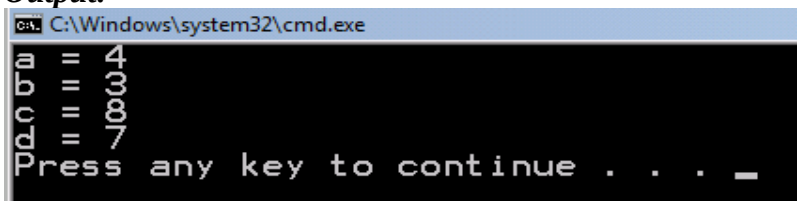
A.java

```
public class A{
   int a;
   int b;
   int c;

   A(int p, int q, int r){
     a=p;
     b=q;
     c=r;
   }

}
```

B,java

```
class B extends A{
     int d;
     B(int l, int m, int n, int o){
       super(l,m,n);
       d=o;
       }
     void Show(){
       System.out.println("a = " + a);
       System.out.println("b = " + b);
       System.out.println("c = " + c);
       System.out.println("d = " + d);
     }

     public static void main(String args[]){
       B b = new B(4,3,8,7);
       b.Show();
     }
}
```

Output:



Example : 2

Circle.java

```
import java.util.*;

public class Circle {

//declaring the instance variable
protected double radius;

public Circle(double radius)
{this.radius = radius;}


public double getArea()
{return Math.PI*radius*radius;}

}
```

Cylinder.java

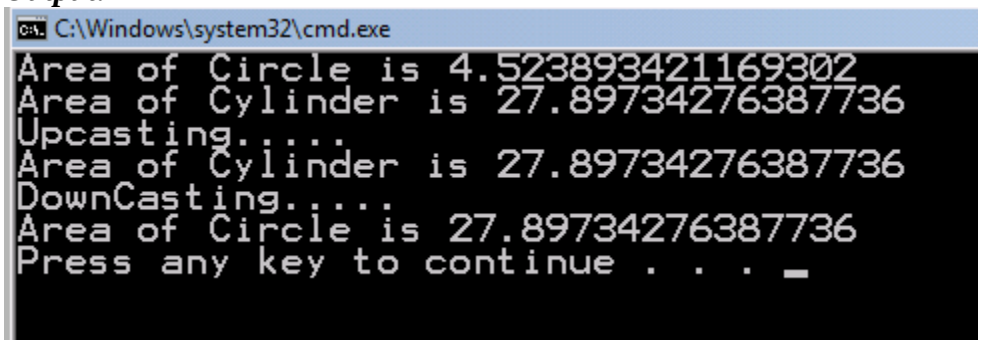
```
import java.util.*;

public class Cylinder extends Circle {


protected double length;


public Cylinder(double radius, double length)
    {
    super(radius);
    this.length = length;
    }


 // method overriden here
public double getArea()
{ return 2*super.getArea()+2*Math.PI*radius*length; }


}
```

Code Example of Inheritance and Polymarphism

Test.java

```
public class Test{
public static void main(String[] args)
{

//Circle Class
Circle myCircle = new Circle(1.20);
System.out.println("Area of Circle is "+ myCircle.getArea());

//Cylinder Class
Cylinder myCylinder = new Cylinder(1.20,2.50);
System.out.println("Area of Cylinder is "+ myCylinder.getArea());

//UpCasting
Circle myCircle1 = new Cylinder(1.20,2.50);
System.out.println("Upcasting.....");
System.out.println("Area of Cylinder is "+ myCircle1.getArea());


//DownCasting
Circle myCircle2 = new Cylinder(1.20,2.50);
Cylinder myCylinder2;
myCylinder2 = (Cylinder) myCircle2;
System.out.println("DownCasting.....");
System.out.println("Area of Circle is "+ myCylinder2.getArea());

}

}
```

Output:

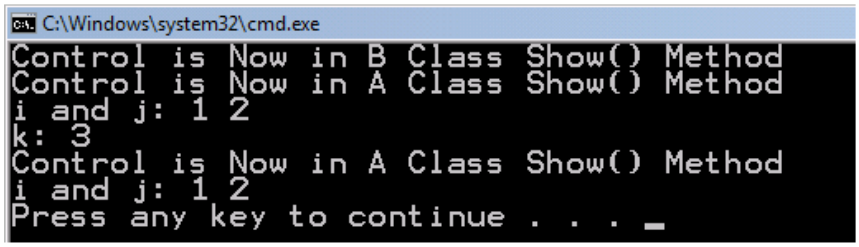


Example : 3

Override.java

```
class A {
int i, j;
A(int a, int b) {
i = a;
j = b;
}

void show() {
    System.out.println("Control is Now in A Class Show() Method");
    System.out.println("i and j: " + i + " " + j);}
}


class B extends A {
int k;
B(int a, int b, int c) {
super(a, b);
k = c;
}

void show() {
    System.out.println("Control is Now in B Class Show() Method");
    super.show(); // this calls show() in Super class (A)
    System.out.println("k: " + k);}
    }


class Override {
public static void main(String args[]) {
B subOb = new B(1, 2, 3);
A supOb = new A(1, 2);
subOb.show(); // this calls show() in B class
supOb.show(); // this calls show() in A class
}

}
```

Output:



Example : 4
Overloading.java

```
import javax.swing.*;

public class Overloading {


    public static void main(String[] args) {

        double n1   = getDouble();
        double n2   = getDouble("Enter the second number.");
        double n3   = getDouble("Enter last number.", 0.0, 100.0);

        double average = (n1 + n2 + n3) / 3.0;

        displayString("Average is " + average);
    }


    private static double getDouble()
    { return getDouble("Enter a number");}


    private static double getDouble(String prompt)
   {
        String tempStr;
        tempStr = JOptionPane.showInputDialog(null, prompt);
        return Double.parseDouble(tempStr);
    }


   private static double getDouble(String prompt, double low, double high) {
       double result;
       String rangePrompt = prompt + "  Value must be in range "
                                   + low + " to " + high;

       do {
           result = getDouble(rangePrompt);
       } while (result < low || result > high);

       return result;
   }


   private static void displayString(String output)
    {JOptionPane.showMessageDialog(null, output);}
```