package homework;

super class and parent class

public class Human {  
      
    private String s1;  
    private String s2;  
    {  
            String s1, s2;  
              
                  s1 ="Super class";  
                  s2 ="Parent class";  
            }  
            public Human(String str)  
            {  
                   s1= str;  
                   s2= str;  
            }  
    }  
    class Boy extends Human  
    {  
            private static String str;  
            private String s2;  
            private String s1;  
            public Boy() {  
            super(str);  
            // TODO Auto-generated constructor stub  
        }  
              
                  
  
            public Class Boy()  
            {  
                  s2 ="Child class";  
                return null;  
            }  
            public void disp()  
            {  
                   System.out.println("String 1 is: "+s1);  
                   System.out.println("String 2 is: "+s2);  
            }  
            public static void main(String args[])  
            {  
                    Boy obj = new Boy();  
                    obj.disp();  
            }  
    }

overload:  
  
public class Overload {  
      
      
    void demo (int a)  
    {  
       System.out.println ("a: " + a);  
    }  
    void demo (int a, int b)  
    {  
       System.out.println ("a and b: " + a + "," + b);  
    }  
    double demo(double a) {  
       System.out.println("double a: " + a);  
       return a\*a;  
    }  
}  
class MethodOverloading  
{  
    public static void main (String args []) {  
    }   
    {  
        Overload Obj = new Overload();  
        double result;  
        Obj .demo(10);  
        Obj .demo(10, 20);  
        result = Obj .demo(5.5);  
        System.out.println("O/P : " + result);  
    }  
}

Resever:  
package homework;  
  
public class resever {    
    
 private Node head;    
    
 private static class Node {    
  private int value;    
  private Node next;    
    
  Node(int value) {    
   this.value = value;    
    
  }    
 }    
    
 public void addToTheLast(Node node) {    
    
  if (head == null) {    
   head = node;    
  } else {    
   Node temp = head;    
   while (temp.next != null)    
    temp = temp.next;    
    
   temp.next = node;    
  }    
 }    
    
    
 public void printList(Node head) {    
  Node temp = head;    
  while (temp != null) {    
   System.out.format("%d ", temp.value);    
   temp = temp.next;    
  }    
  System.out.println();    
 }    
    
 // Reverse list using this function     
public static Node reverseLinkedList(Node currentNode)    
 {    
// For first node, previousNode will be null    
Node previousNode=null;    
  Node nextNode;    
  while(currentNode!=null)    
  {    
   nextNode=currentNode.next;    
  // reversing the link    
   currentNode.next=previousNode;    
  // moving currentNode and previousNode by 1 node    
   previousNode=currentNode;    
   currentNode=nextNode;    
  }    
  return previousNode;    
 }    
    
 public static void main(String[] args) {    
  resever list = new resever();    
  // Creating a linked list    
  Node head=new Node(5);    
  list.addToTheLast(head);    
  list.addToTheLast(new Node(6));    
  list.addToTheLast(new Node(7));    
  list.addToTheLast(new Node(1));    
  list.addToTheLast(new Node(2));    
     
  list.printList(head);    
  //Reversing LinkedList    
  Node reverseHead=reverseLinkedList(head);    
  System.out.println("After reversing");    
  list.printList(reverseHead);    
     
 }    
    
}

linked list:  
public class linkedlist {  
    class LinkedList {  
  
           static class Node {  
  
               int data;  
               Node next;  
  
               Node(int d) {  
                   data = d;  
                   next = null;  
               }  
           }  
  
           // A simple and tail recursive function to reverse  
           // a linked list.  prev is passed as NULL initially.  
           Node reverseUtil(Node curr, Node prev) {  
  
               /\* If last node mark it head\*/  
               if (curr.next == null) {  
                   head = curr;  
  
                   /\* Update next to prev node \*/  
                   curr.next = prev;  
                   return null;  
               }  
  
               /\* Save curr->next node for recursive call \*/  
               Node next1 = curr.next;  
  
               /\* and update next ..\*/  
               curr.next = prev;  
  
               reverseUtil(next1, curr);  
               return head;  
           }  
  
           // prints content of double linked list  
           void printList(Node node) {  
               while (node != null) {  
                   System.out.print(node.data + " ");  
                   node = node.next;  
               }  
           }  
  
           public static void main(String[] args) {  
               LinkedList list = new LinkedList();  
               list.head = new Node(1);  
               list.head.next = new Node(2);  
               list.head.next.next = new Node(2);  
               list.head.next.next.next = new Node(4);  
               list.head.next.next.next.next = new Node(5);  
               list.head.next.next.next.next.next = new Node(6);  
               list.head.next.next.next.next.next.next = new Node(7);  
               list.head.next.next.next.next.next.next.next = new Node(8);  
  
               System.out.println("Original Linked list is :");  
               list.printList(head);  
               Node res = list.reverseUtil(head, null);  
               System.out.println("");  
               System.out.println("");  
               System.out.println("Reversed linked list : ");  
               list.printList(res);  
           }  
        }  
  
}

facebook:  
package homework;  
import [org.openqa.selenium.By](http://org.openqa.selenium.by/);  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
  
  
public class facebook {  
  
    public static void main(String[] args)throws InterruptedException   
     {  
        // TODO Auto-generated method stub  
        WebDriver d=new FirefoxDriver();  
        // open URL  
          
        d.get("[http://www.facebook.com](http://www.facebook.com/)");  
          
      
        // send emaild to  textbox  
        Thread.sleep(3000);  
        d.findElement(By.id("facebook")).sendKeys("[susmithareddy4@gmail.com](mailto:tweety.usha11@gmail.com)");  
          
              
        // click on next  
        Thread.sleep(3000);  
        d.findElement(By.id("next")).click();  
          
        // send password to textbox  
        Thread.sleep(3000);  
        d.findElement(By.id("Passwd")).sendKeys("ushakalavati");  
          
        // click on sign button  
        Thread.sleep(3000);  
        d.findElement(By.id("signIn")).click();  
          
        // click on sign down  
        Thread.sleep(3000);  
        d.findElement(By.id("signout")).click();  
          
          
          
    }  
}