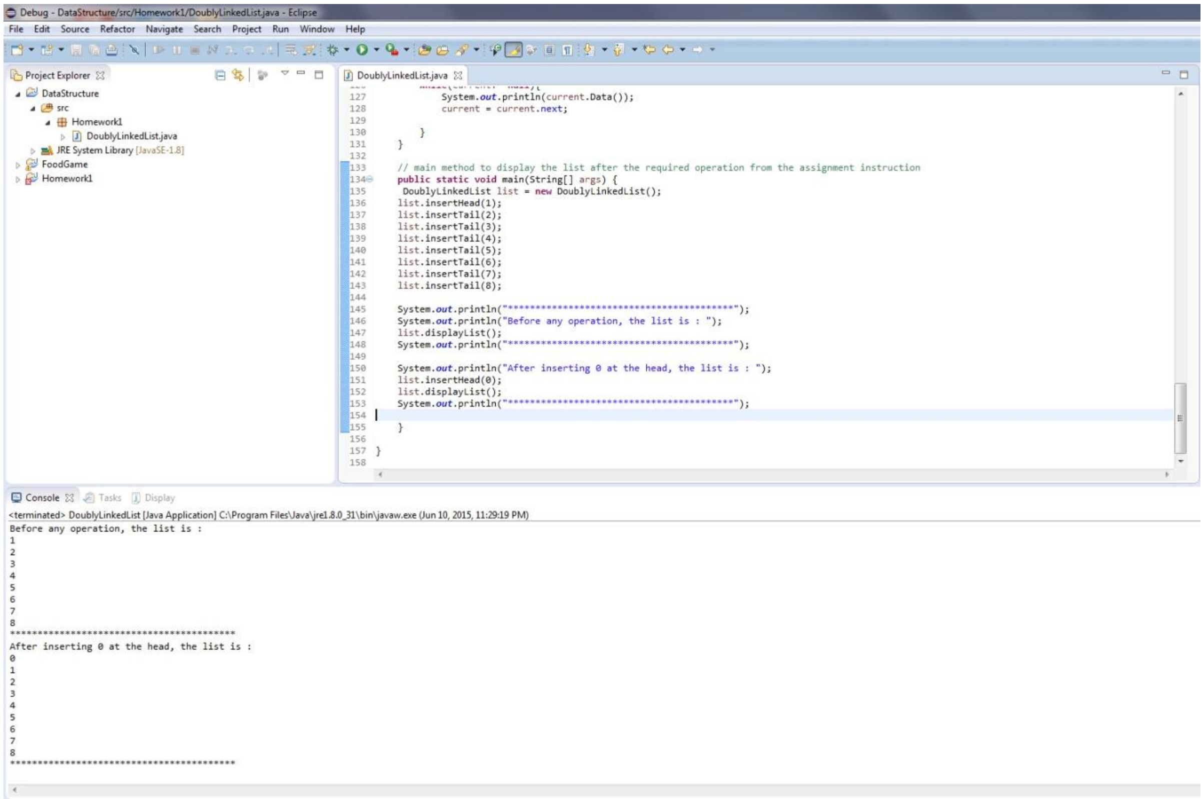
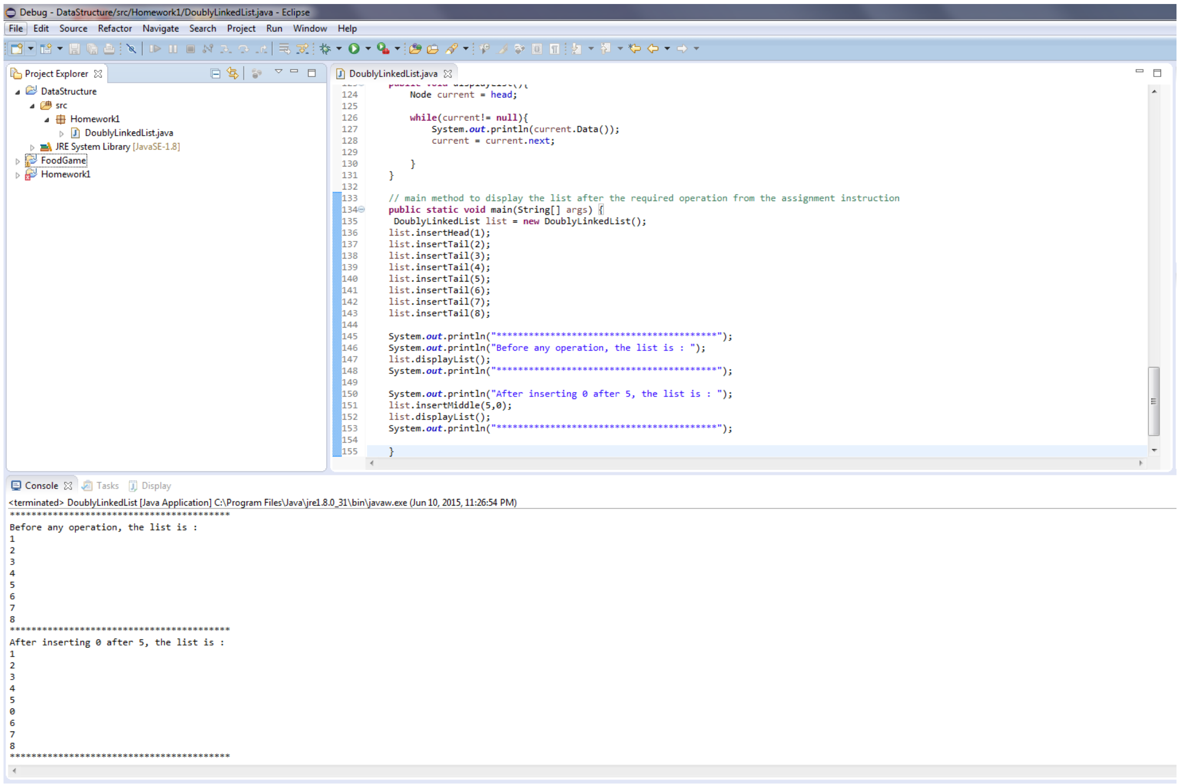
```
/*****************************************************************
Make double linked list - next and a previous pointer.
a. Write insert function.  Run the function 3 times.
Show an insert at the head, tail and the middle of the list.
b. Write delete function.  Run it 3 times. Show a delete at
head of list,tail of list and from middle of list.
First you will create a list with at least 8 items.
Run the programs and capture the screen shot.
Submit each screen shot and the code.
*****************************************************************/
package Homework1;

public class DoublyLinkedList {
       // Construct the Node class for the nodes and their internal link in the
double lined list
       public class Node {
              public Node pre;
              private int data;
              public Node next;

              //First constructor
              public Node(int data) {
                     pre = null;
                     this.data = data;
                     next = null;
              }

              //Second constructor
              public Node(Node pre, int data, Node next){
                     this.pre = pre;
                     this.data = data;
                     this.next = next;
              }

              // Get data method to return the value stored in the node
              public int Data(){
                     return data;
              }
       }

       private Node head;

       // insertHead method to insert an element in the head of the list
       public void insertHead(int data){
              if (head == null){
                     head = new Node(null, data, null);
              }
           else{
              Node newNode = new Node(null, data, head);
              head.pre = newNode;
               head = newNode;
           }
       }

       // insertMiddle method to insert an element in the middle of the list
```

```
    public void insertMiddle(int data1, int data2){
     Node current = head;

        while(current.Data()!= data1){
            current = current.next;
        }
        Node newNode = new Node(current,data2,current.next);
        newNode.next = current.next;
        current.next.pre = newNode;
        current.next = newNode;
        newNode.pre = current;
        }

    // insertTail method to insert an element in the tail of the list
    public void insertTail(int data){
        if (head == null){
            head = new Node(null, data, null);
        }
        else{
            Node current = head;
            while(current.next != null){
                current = current.next;
            }
            Node newNode = new Node(current, data, null);
            current.next = newNode;
        }
    }

    // deleteHead method to delete the head from the list
    public void deleteHead(){
        if (head == null){
        System.out.println("The list is empty!");
        }
      else{
        head.next.pre = null;
         head = head.next;
      }
    }

    // deleteMiddle method to delete an element in the middle of the list
    public void deleteMiddle(int data){
        if (head == null){
        System.out.println("The list is empty!");
        }
      else{
        Node current = head;
            while(current.Data()!= data){
                current = current.next;
            }
            current.pre.next = current.next;
      }
}

    // deleteTail method to delete the tail from the list
    public void deleteTail(){
        if (head == null){
            System.out.println("The list is empty!");
        }
        else{
            Node current = head;
            while(current.next != null){
                current = current.next;
            }
            current.pre.next = null;
        }
    }

    // displayList method to display every element in the list
    public void displayList(){
        Node current = head;

        while(current!= null){
            System.out.println(current.Data());
            current = current.next;

        }
    }

    // main method to display the list after the required operation from the
assignment instruction
    public static void main(String[] args) {
     DoublyLinkedList list = new DoublyLinkedList();
    list.insertHead(1);
    list.insertTail(2);
    list.insertTail(3);
    list.insertTail(4);
    list.insertTail(5);
    list.insertTail(6);
    list.insertTail(7);
    list.insertTail(8);

    System.out.println("*****************************************");
    System.out.println("Before any operation, the list is : ");
    list.displayList();
    System.out.println("*****************************************");

    System.out.println("After inserting 0 at the head, the list is : ");
    list.insertHead(0);
    list.displayList();
    System.out.println("*****************************************");

/*****************************************************************************
  The other functions were put in the comment section so that
  they will not affect the result!

    System.out.println("After inserting 0 after 5, the list is : ");
    list.insertMiddle(5,0);
    list.displayList();
    System.out.println("*****************************************");
  System.out.println("After inserting 0 at the tail, the list is : ");
    list.insertTail(0);
    list.displayList();
    System.out.println("*****************************************");


  System.out.println("After deleting the head, the list is : ");
    list.deleteHead();
    list.displayList();
    System.out.println("*****************************************");

  System.out.println("After deleting the middle 5, the list is : ");
    list.deleteMiddle(5);
    list.displayList();
    System.out.println("*****************************************");

  System.out.println("After deleting the tail, the list is : ");
    list.deleteTail();
    list.displayList();
    System.out.println("*****************************************");

*****************************************************************************/

    }

}
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

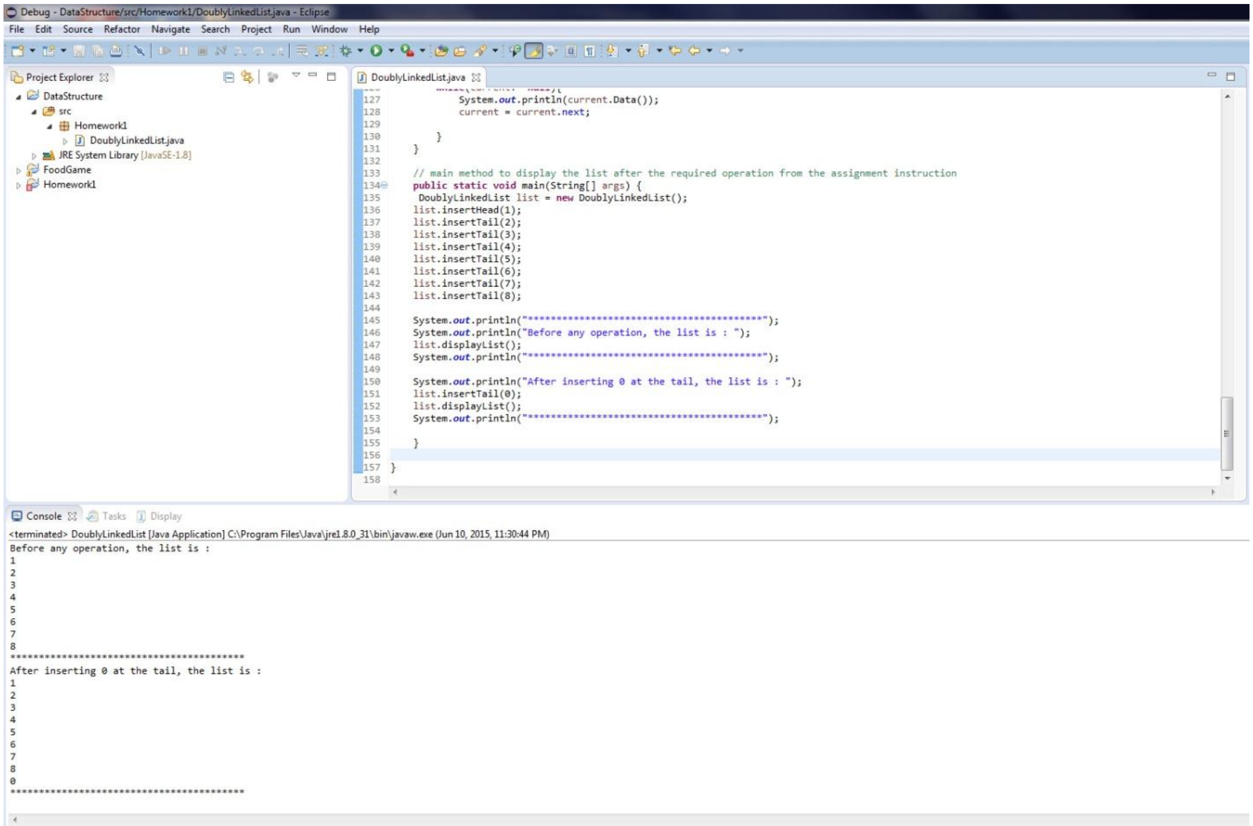
```java
            System.out.println(current.Data());
            current = current.next;

        }
    }

    // main method to display the list after the required operation from the assignment instruction
    public static void main(String[] args) {
     DoublyLinkedList list = new DoublyLinkedList();
    list.insertHead(1);
    list.insertTail(2);
    list.insertTail(3);
    list.insertTail(4);
    list.insertTail(5);
    list.insertTail(6);
    list.insertTail(7);
    list.insertTail(8);

    System.out.println("****************************************");
    System.out.println("Before any operation, the list is : ");
    list.displayList();
    System.out.println("****************************************");

    System.out.println("After inserting 0 at the head, the list is : ");
    list.insertHead(0);
    list.displayList();
    System.out.println("****************************************");

    }

}
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:29:19 PM)
Before any operation, the list is :
1
2
3
4
5
6
7
8
****************************************
After inserting 0 at the head, the list is :
0
1
2
3
4
5
6
7
8
****************************************
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

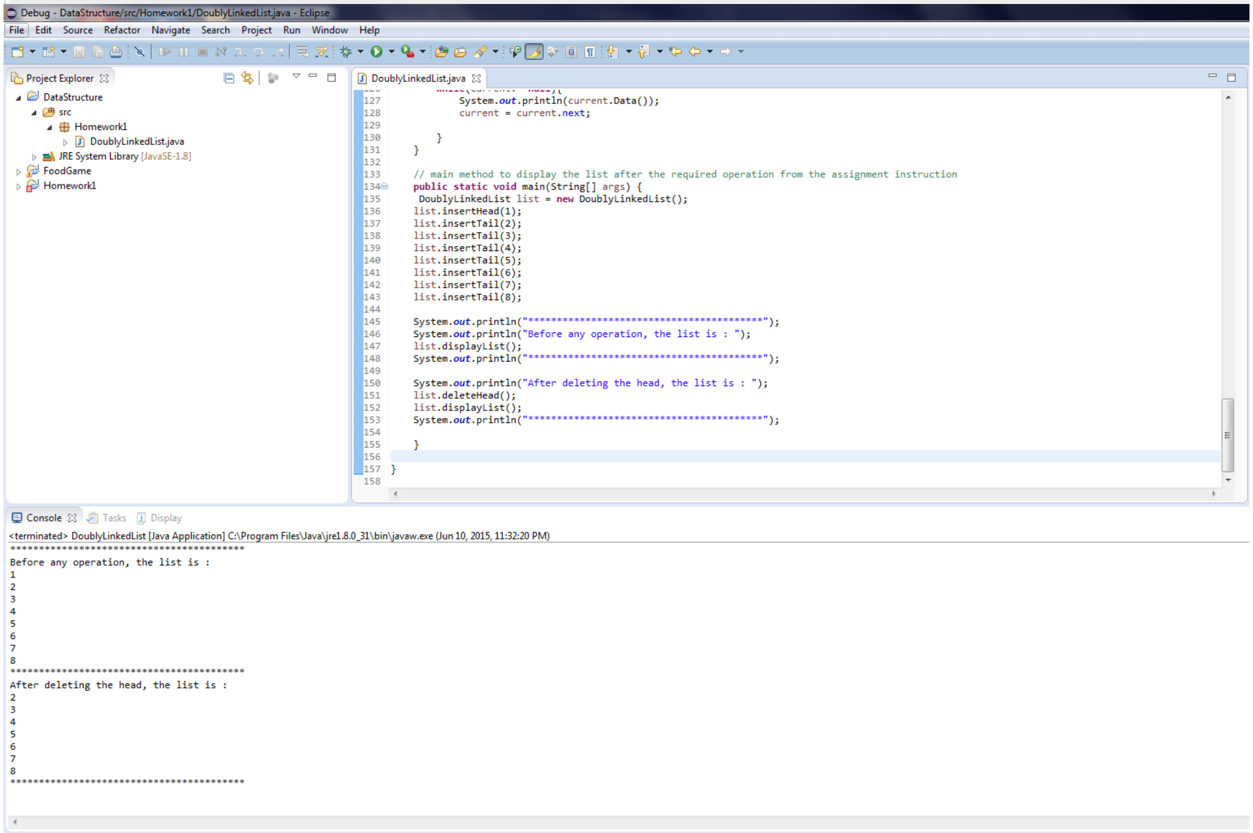
```java
        Node current = head;

        while(current!= null){
            System.out.println(current.Data());
            current = current.next;

        }
    }

    // main method to display the list after the required operation from the assignment instruction
    public static void main(String[] args) {
     DoublyLinkedList list = new DoublyLinkedList();
    list.insertHead(1);
    list.insertTail(2);
    list.insertTail(3);
    list.insertTail(4);
    list.insertTail(5);
    list.insertTail(6);
    list.insertTail(7);
    list.insertTail(8);

    System.out.println("****************************************");
    System.out.println("Before any operation, the list is : ");
    list.displayList();
    System.out.println("****************************************");

    System.out.println("After inserting 0 after 5, the list is : ");
    list.insertMiddle(5,0);
    list.displayList();
    System.out.println("****************************************");

    }
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:26:54 PM)
****************************************
Before any operation, the list is :
1
2
3
4
5
6
7
8
****************************************
After inserting 0 after 5, the list is :
1
2
3
4
5
0
6
7
8
****************************************
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

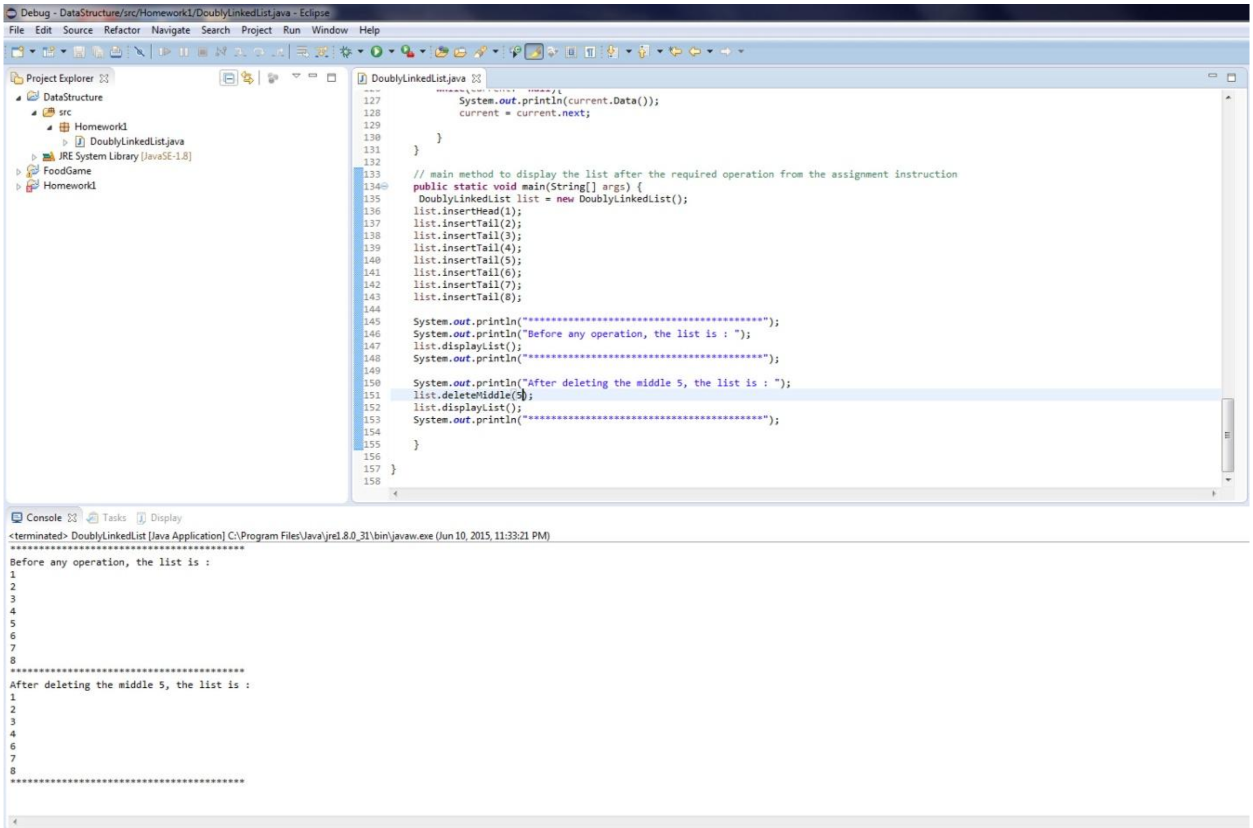
```
                System.out.println(current.Data());
                current = current.next;

            }
        }

        // main method to display the list after the required operation from the assignment instruction
        public static void main(String[] args) {
         DoublyLinkedList list = new DoublyLinkedList();
        list.insertHead(1);
        list.insertTail(2);
        list.insertTail(3);
        list.insertTail(4);
        list.insertTail(5);
        list.insertTail(6);
        list.insertTail(7);
        list.insertTail(8);

        System.out.println("*****************************************");
        System.out.println("Before any operation, the list is : ");
        list.displayList();
        System.out.println("*****************************************");

        System.out.println("After inserting 0 at the tail, the list is : ");
        list.insertTail(0);
        list.displayList();
        System.out.println("*****************************************");

        }

}
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:30:44 PM)
Before any operation, the list is :
1
2
3
4
5
6
7
8
*****************************************
After inserting 0 at the tail, the list is :
1
2
3
4
5
6
7
8
0
*****************************************
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

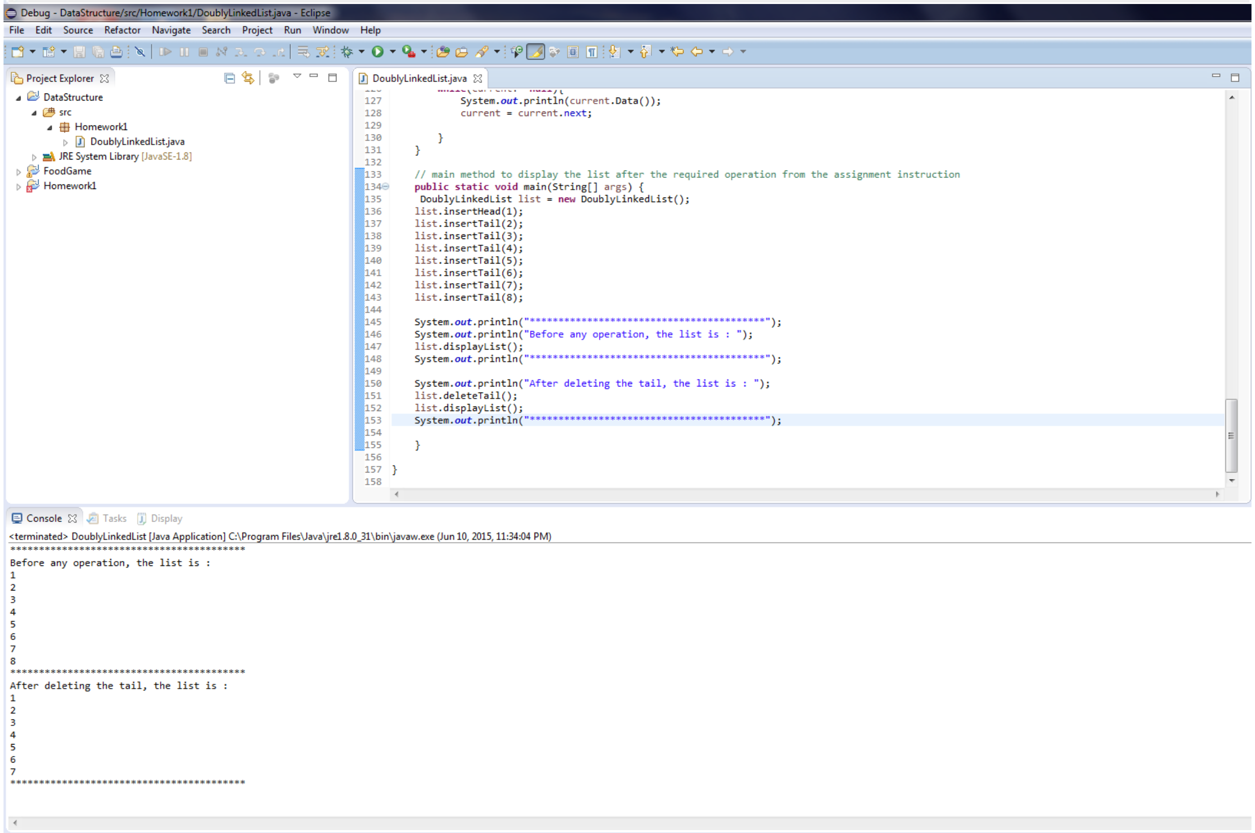
```
                System.out.println(current.Data());
                current = current.next;

            }
        }

        // main method to display the list after the required operation from the assignment instruction
        public static void main(String[] args) {
         DoublyLinkedList list = new DoublyLinkedList();
        list.insertHead(1);
        list.insertTail(2);
        list.insertTail(3);
        list.insertTail(4);
        list.insertTail(5);
        list.insertTail(6);
        list.insertTail(7);
        list.insertTail(8);

        System.out.println("*****************************************");
        System.out.println("Before any operation, the list is : ");
        list.displayList();
        System.out.println("*****************************************");

        System.out.println("After deleting the head, the list is : ");
        list.deleteHead();
        list.displayList();
        System.out.println("*****************************************");

        }

}
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:32:20 PM)
*****************************************
Before any operation, the list is :
1
2
3
4
5
6
7
8
*****************************************
After deleting the head, the list is :
2
3
4
5
6
7
8
*****************************************
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

```java
            System.out.println(current.Data());
            current = current.next;

        }
    }

    // main method to display the list after the required operation from the assignment instruction
    public static void main(String[] args) {
     DoublyLinkedList list = new DoublyLinkedList();
    list.insertHead(1);
    list.insertTail(2);
    list.insertTail(3);
    list.insertTail(4);
    list.insertTail(5);
    list.insertTail(6);
    list.insertTail(7);
    list.insertTail(8);

    System.out.println("******************************************");
    System.out.println("Before any operation, the list is : ");
    list.displayList();
    System.out.println("******************************************");

    System.out.println("After deleting the middle 5, the list is : ");
    list.deleteMiddle(5);
    list.displayList();
    System.out.println("******************************************");

    }

}
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:33:21 PM)
******************************************
Before any operation, the list is :
1
2
3
4
5
6
7
8
******************************************
After deleting the middle 5, the list is :
1
2
3
4
6
7
8
******************************************
```


Debug - DataStructure/src/Homework1/DoublyLinkedList.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DataStructure
src
Homework1
DoublyLinkedList.java
JRE System Library [JavaSE-1.8]
FoodGame
Homework1
DoublyLinkedList.java
Console
Tasks
Display

```java
            System.out.println(current.Data());
            current = current.next;

        }
    }

    // main method to display the list after the required operation from the assignment instruction
    public static void main(String[] args) {
     DoublyLinkedList list = new DoublyLinkedList();
    list.insertHead(1);
    list.insertTail(2);
    list.insertTail(3);
    list.insertTail(4);
    list.insertTail(5);
    list.insertTail(6);
    list.insertTail(7);
    list.insertTail(8);

    System.out.println("******************************************");
    System.out.println("Before any operation, the list is : ");
    list.displayList();
    System.out.println("******************************************");

    System.out.println("After deleting the tail, the list is : ");
    list.deleteTail();
    list.displayList();
    System.out.println("******************************************");

    }

}
```

```
<terminated> DoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (Jun 10, 2015, 11:34:04 PM)
******************************************
Before any operation, the list is :
1
2
3
4
5
6
7
8
******************************************
After deleting the tail, the list is :
1
2
3
4
5
6
7
******************************************
```