湖北工業大學

HUBEI UNIVERSITY OF TECHNOLOGY

Course exercise for Object oriented programming

Class	-	21LQ
Major	-	SOFTWARE ENGINEERING
Student ID	-	2111552104
Name	-	ASADUZZAMAN MD

1. Implementing a singly-Linked lists.
Given:

```
class SList{
        public SListNode head;
        public int size;
        public SList(){
                head = null;
                size = 0;         }
        public int length() {
                return size;       }
        public void insertFront(Object item){
                head = new SListNode(item, head);
                size++;   }
        public String toString() {
                int i;
                Object obj;
                String result = "[    ";
                SListNode cur = head;
                while (cur != null) {
                    obj = cur.item;
                    result = result + obj.toString() + "    ";
                    cur = cur.next;               }
                result = result + "]";
                return result;
        }
}
```

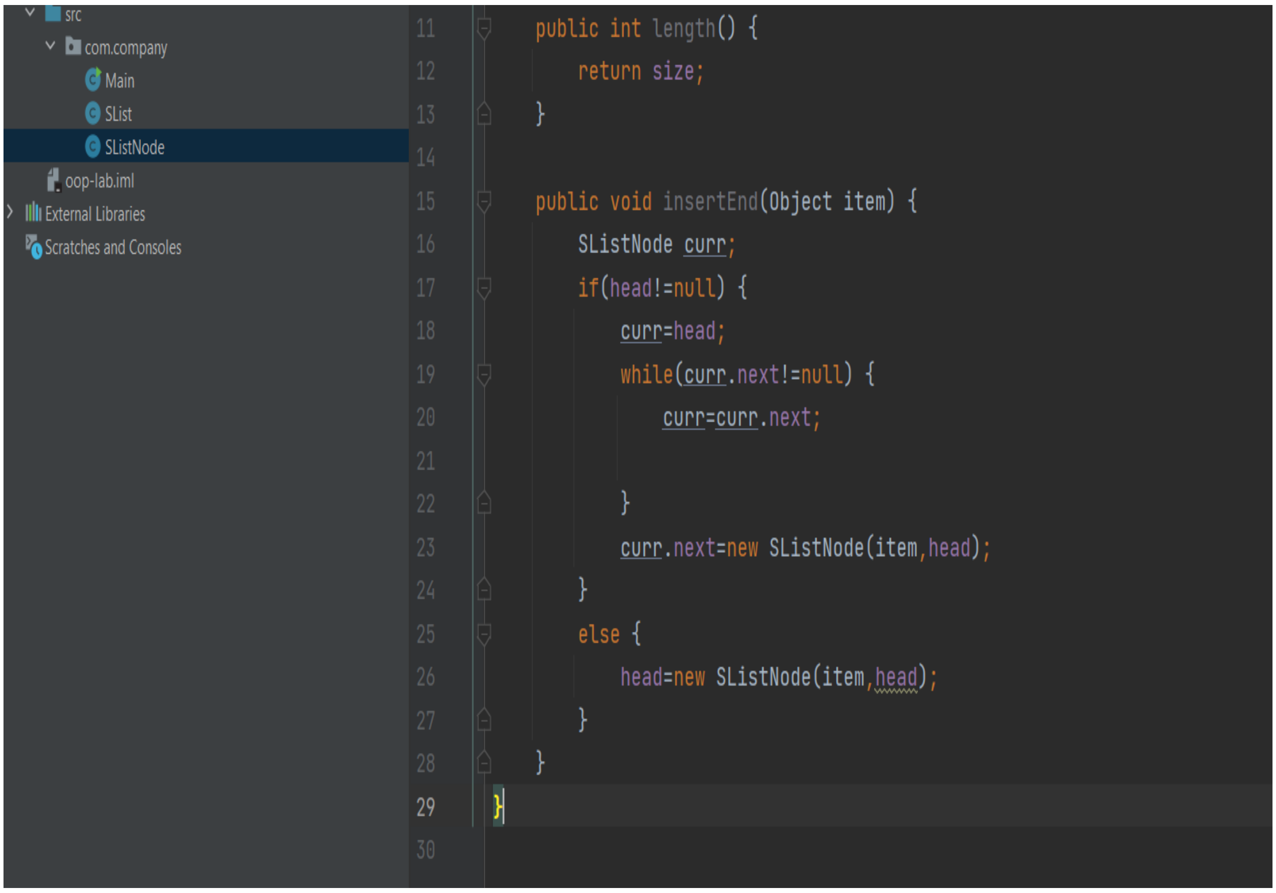
```
class SListNode{
  public Object item;
  public SListNode next;
        public SListNode(Object o, SListNode n){
          item = o;
          next = n;
  }
  public SListNode(Object o){
          item = o;
          next = null;
  }
}
```

1) Add an *insertEnd*(Object item) method for the SList class. This method inserts the parameter "item" at the end of this list.
   public void insertEnd(Object item) {

Required Code -

```
public void insertEnd(Object item) {
SListNode curr;
if(head!=null) {
        curr=head;
        while(curr.next!=null) {
                curr=curr.next;

        }
        curr.next=new SListNode(item,head);
}
else {
        head=new SListNode(item,head);
}
}
```

Code Screenshot in Intellij idea -



```
    public int length() {
        return size;
    }

    public void insertEnd(Object item) {
        SListNode curr;
        if(head!=null) {
            curr=head;
            while(curr.next!=null) {
                curr=curr.next;

            }
            curr.next=new SListNode(item,head);
        }
        else {
            head=new SListNode(item,head);
        }
    }
}
```

Explanation of my design:

We have a SList class and a SListNode class which were created before. Now I am adding an insertEnd(Object item) method in the SList class.

In this case, Iterate through the list till the end node and then connect new node to the last node.

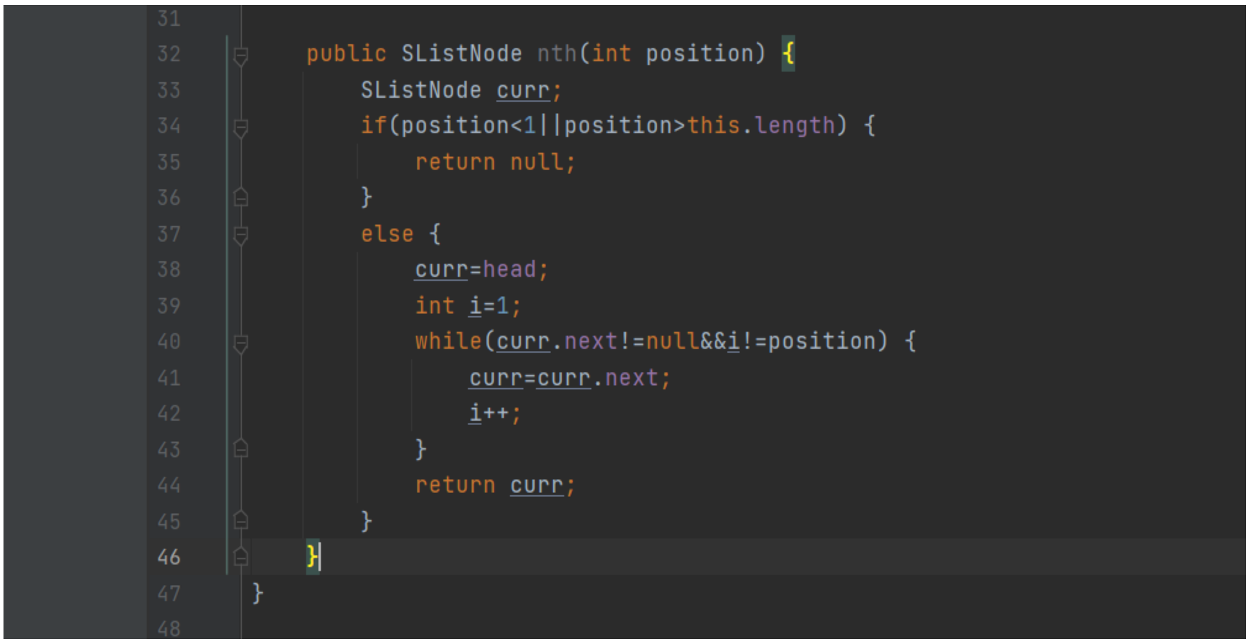
And, if list is empty then intialize head to new node.

2) Add a *nth*(int position) method for the SList class. This method returns the item at the specified position. If position < 1 or position > this.length(), null is returned. The range of the parameter position is from 1 to length().
   public int nth(int position){

Required Code -

```
    public SListNode nth(int position) {
    SListNode curr;
    if(position<1||position>this.length) {
        return null;
    }
    else {
        curr=head;
        int i=1;
        while(curr.next!=null&&i!=position) {
            curr=curr.next;
            i++;
        }
        return curr;
    }
}
```

Code Screenshot in Intellij idea -



```
    public SListNode nth(int position) {
        SListNode curr;
        if(position<1||position>this.length) {
            return null;
        }
        else {
            curr=head;
            int i=1;
            while(curr.next!=null&&i!=position) {
                curr=curr.next;
                i++;
            }
            return curr;
        }
    }
}
```

Explanation of my design:

Here, I am adding a *nth*(int position) method in our SList class. In this case, I am using if-else block where if block is for checking whether the given position is valid or not. In else block we travel through the list using head node till we get the required positon.

Here I am also using a while loop. The while loop is used to traverse the list till given position after that we return the item at that position.

The Output is:

If we enter the position which is valid then node present at that position is returned else null is returned.

3) Write a Test class to test your code. In your test code, you can use the following line of code to output ints in your int list.

All Required Code -

SListNode.java

```
    class SListNode{
     public Object item;
     public SListNode next;
public SListNode(Object o, SListNode n){
         item = o;
         next = n;
     }
     public SListNode(Object o){
         item = o;
         next = null;
     }
}
```

SList.java

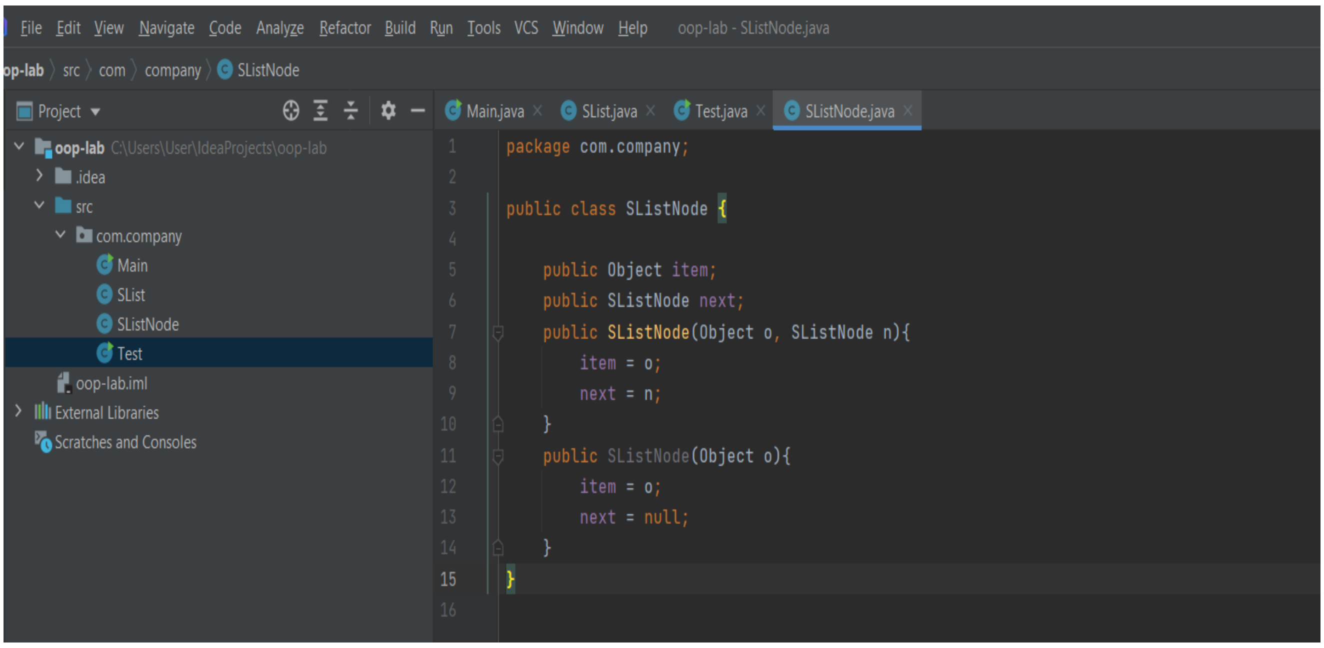
```
   public class SList {

    public SListNode head;
    public int size;
    private int length;

    public SList(){
        head = null;
        size = 0;
    }
    public int length() {
        return size;
    }

    public void insertFront(Object item){
        head = new SListNode(item, head);
        size++;
    }
    public String toString() {

        int i;

        Object obj;
        String result = "[    ";
        SListNode cur = head;

        while (cur != null) {
            obj = cur.item;
            result = result + obj.toString() + "    ";
            cur = cur.next;
        }
        result = result + "]";
        return result;
    }
}
```

Test.java

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

        String str;
        SList list = new SList();
        int[] num = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
        for (int n : num) {
            list.insertFront(n);
        }
        str = list.toString();
        System.out.println(str);
    }
}
```

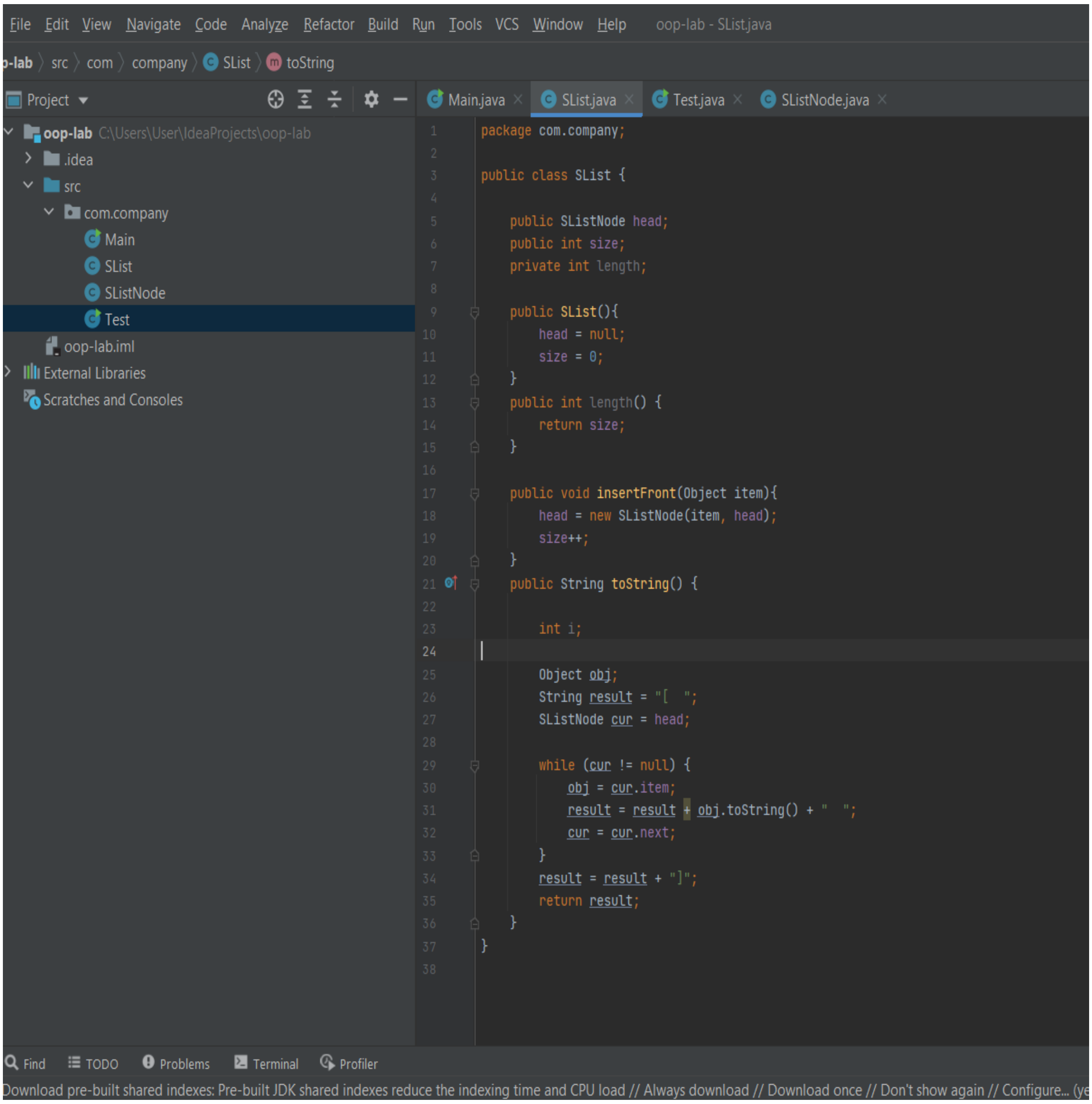
4) Run your program and take a screen shot and paste it here.

Code Screenshot of SListNode class (SListNode.java) -


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help oop-lab - SListNode.java
op-lab src com company SListNode
Project
oop-lab C:\Users\User\IdeaProjects\oop-lab
.idea
src
com.company
Main
SList
SListNode
Test
oop-lab.iml
External Libraries
Scratches and Consoles
Main.java
SList.java
Test.java
SListNode.java

```
package com.company;

public class SListNode {

    public Object item;
    public SListNode next;
    public SListNode(Object o, SListNode n){
        item = o;
        next = n;
    }
    public SListNode(Object o){
        item = o;
        next = null;
    }
}
```

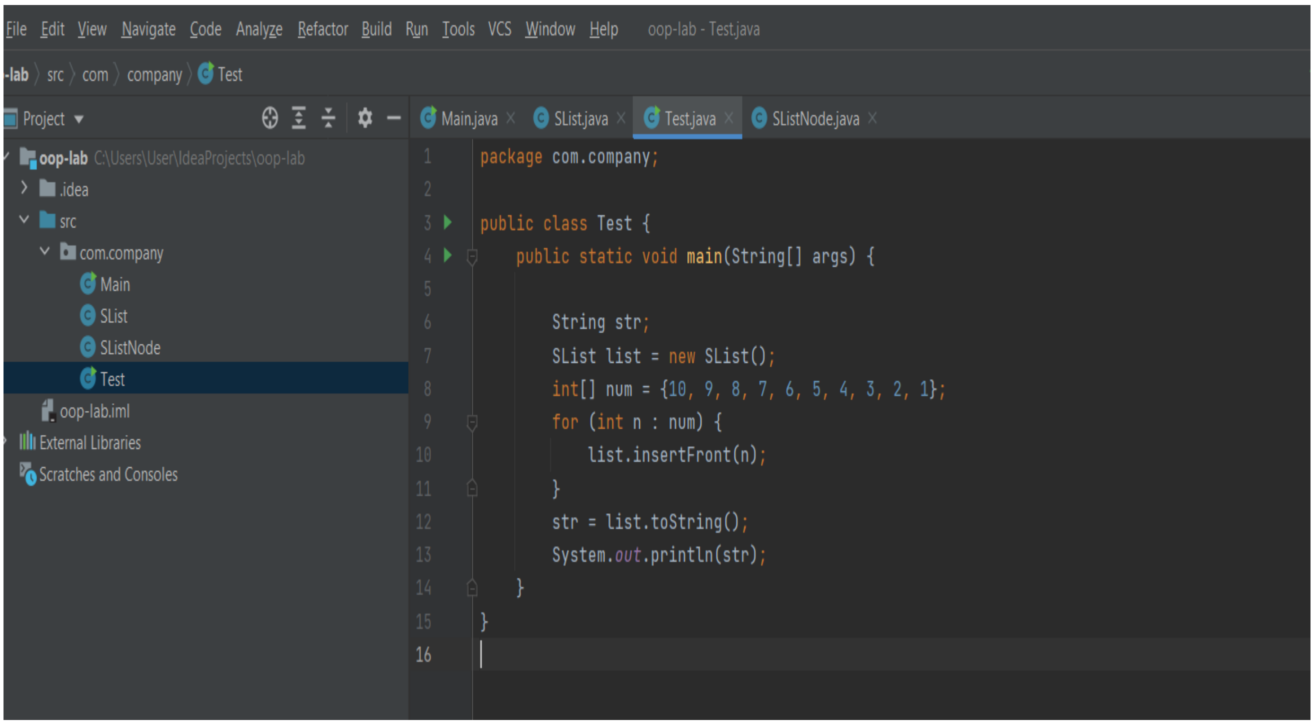
Code Screenshot of SList class (SList.java) -


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help oop-lab - SList.java
-lab src com company SList toString
Project
Main.java SList.java Test.java SListNode.java
oop-lab C:\Users\User\IdeaProjects\oop-lab
.idea
src
com.company
Main
SList
SListNode
Test
oop-lab.iml
External Libraries
Scratches and Consoles
Find TODO Problems Terminal Profiler

```
package com.company;

public class SList {

    public SListNode head;
    public int size;
    private int length;

    public SList(){
        head = null;
        size = 0;
    }
    public int length() {
        return size;
    }

    public void insertFront(Object item){
        head = new SListNode(item, head);
        size++;
    }
    public String toString() {

        int i;

        Object obj;
        String result = "[  ";
        SListNode cur = head;

        while (cur != null) {
            obj = cur.item;
            result = result + obj.toString() + "  ";
            cur = cur.next;
        }
        result = result + "]";
        return result;
    }
}
```

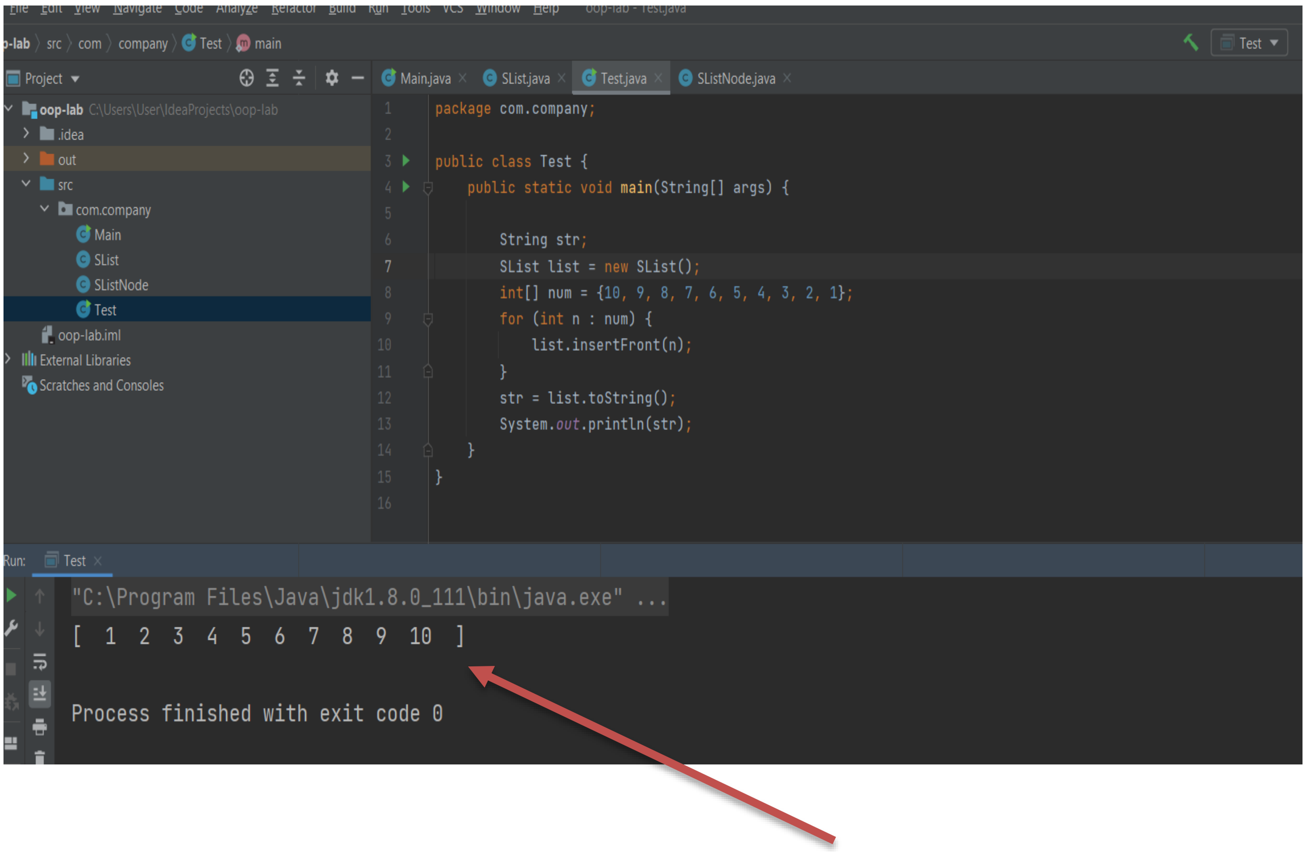
Download pre-built shared indexes: Pre-built JDK shared indexes reduce the indexing time and CPU load // Always download // Download once // Don't show again // Configure... (ye

Code Screenshot of Test class (Test.java) -


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help oop-lab - Test.java
-lab src com company Test
Project
Main.java SList.java Test.java SListNode.java
oop-lab C:\Users\User\IdeaProjects\oop-lab
.idea
src
com.company
Main
SList
SListNode
Test
oop-lab.iml
External Libraries
Scratches and Consoles

```java
package com.company;

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

        String str;
        SList list = new SList();
        int[] num = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
        for (int n : num) {
            list.insertFront(n);
        }
        str = list.toString();
        System.out.println(str);
    }
}
```

Screenshot of the Output -


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help oop-lab - Test.java
-lab src com company Test main
Test
Project
Main.java SList.java Test.java SListNode.java
oop-lab C:\Users\User\IdeaProjects\oop-lab
.idea
out
src
com.company
Main
SList
SListNode
Test
oop-lab.iml
External Libraries
Scratches and Consoles
Run: Test

```java
package com.company;

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

        String str;
        SList list = new SList();
        int[] num = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
        for (int n : num) {
            list.insertFront(n);
        }
        str = list.toString();
        System.out.println(str);
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...
[  1  2  3  4  5  6  7  8  9  10  ]

Process finished with exit code 0
```

Explanation

First, the starter code has been typed out. Now, a variable list has been declared with the type of Class SList. Then, an integer array has been defined with 5 numbers in it which will be assigned in the list.

Then, a for each loop is executed where integer array has been iterated over for every element on which SList method insertFront is applied. Then after the method toString is applied on the list to convert the list to String for the purpose of printing it.

In the end, the string is printed via println function.