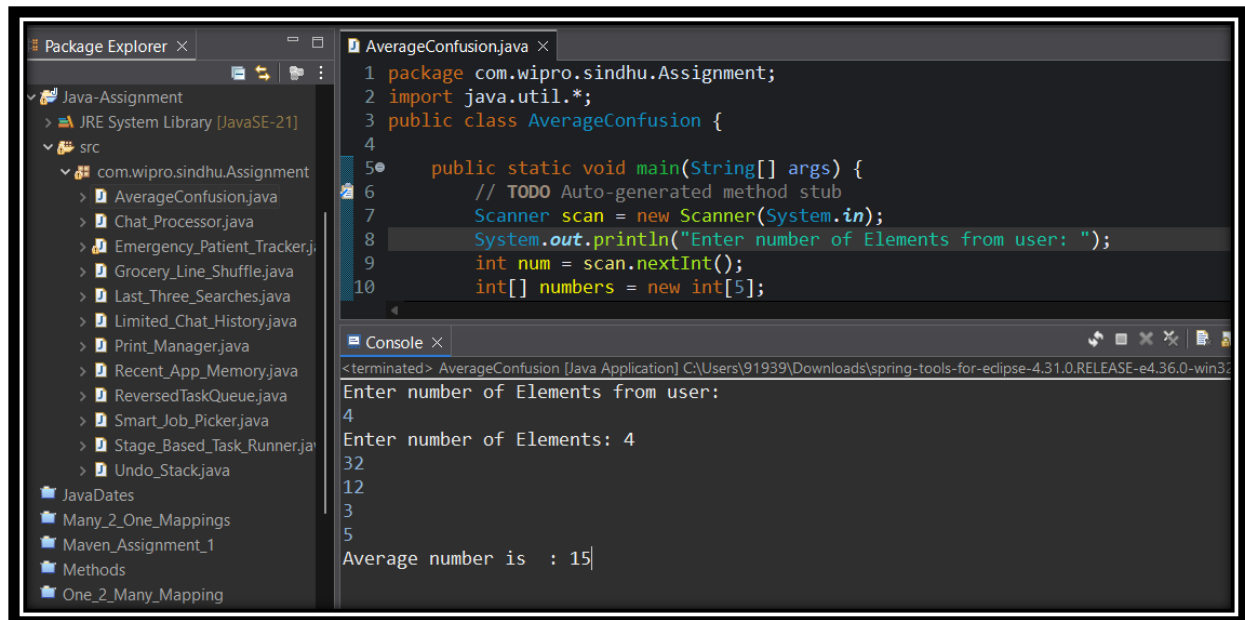


## Java\_Assignment\_Outputs

**AverageConfusion** :- This program reads a list of numbers from the user, applies a transformation to numbers less than 10, then calculates and displays the average of the modified numbers.



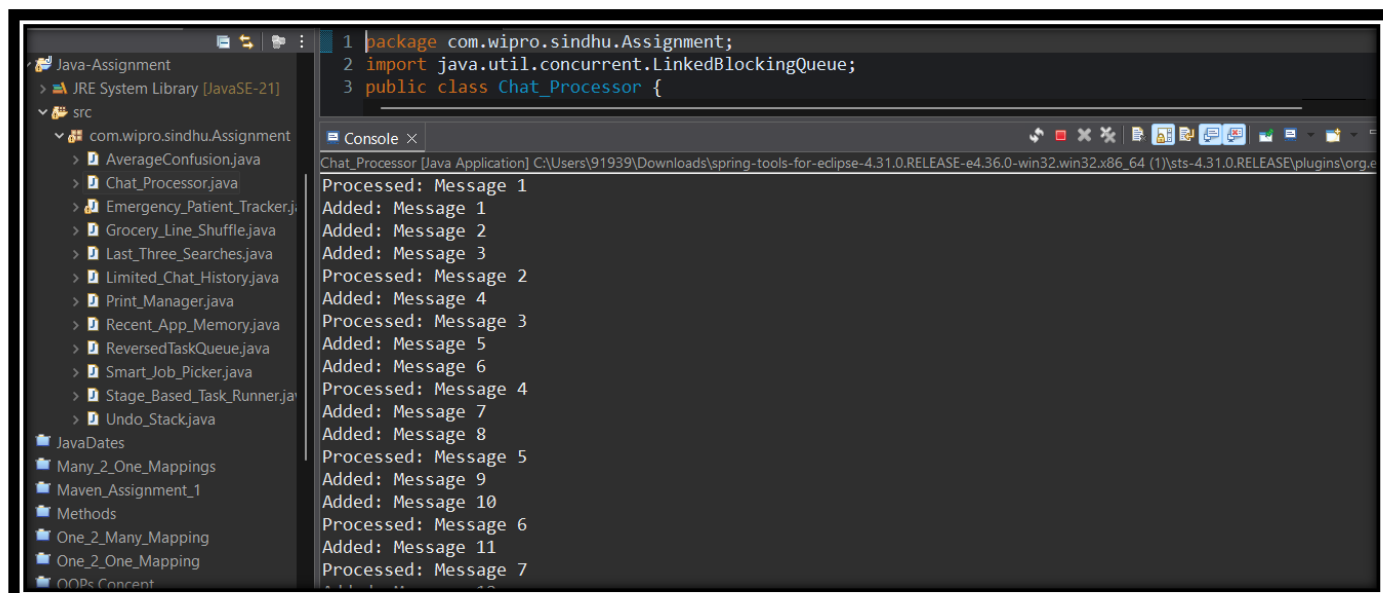
The screenshot shows the Eclipse IDE with the 'Package Explorer' on the left and the 'AverageConfusion.java' file open in the editor. The console at the bottom displays the program's execution.

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class AverageConfusion {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner scan = new Scanner(System.in);
8         System.out.println("Enter number of Elements from user: ");
9         int num = scan.nextInt();
10        int[] numbers = new int[5];
```

Console Output:

```
<terminated> AverageConfusion [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32
Enter number of Elements from user:
4
Enter number of Elements: 4
32
12
3
5
Average number is : 15
```

**Chat Processor** :- It is a simple chat message processing system that means a Producer thread generates messages and adds them to a queue, a Consumer thread takes messages from the queue and processes them.



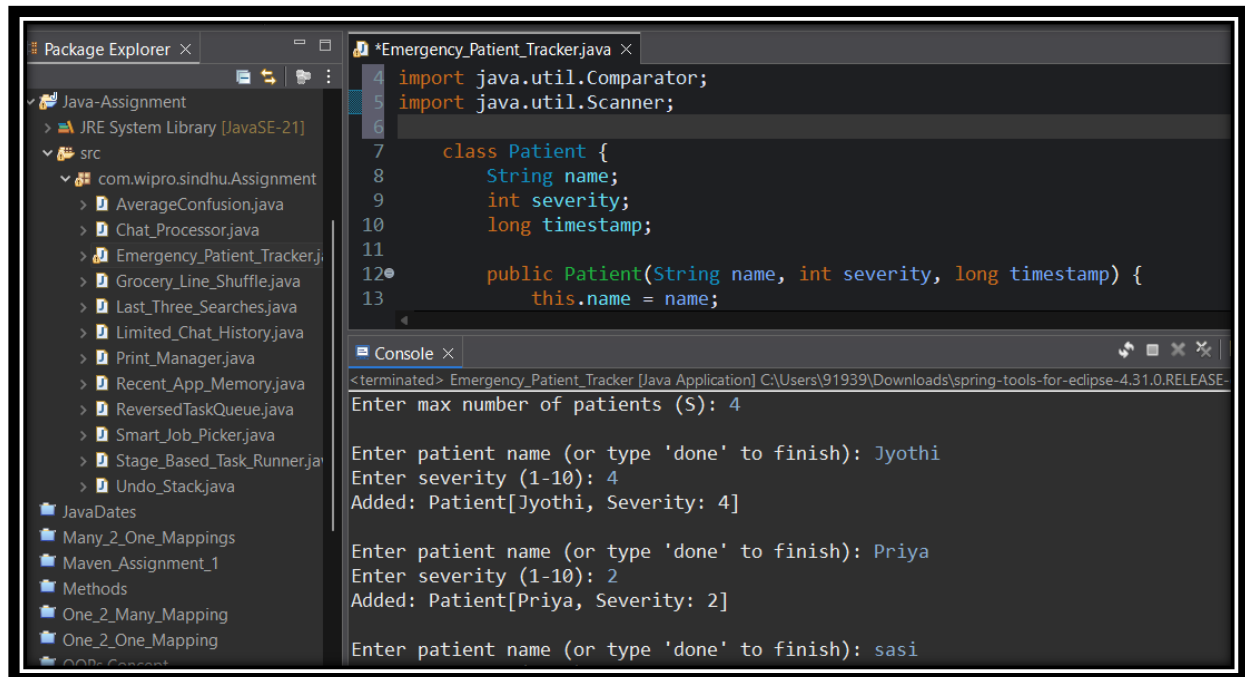
The screenshot shows the Eclipse IDE with the 'Package Explorer' on the left and the 'Chat\_Processor.java' file open in the editor. The console at the bottom displays the program's execution.

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.concurrent.LinkedBlockingQueue;
3 public class Chat_Processor {
```

Console Output:

```
Chat_Processor [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86_64 (1)\sts-4.31.0.RELEASE\plugins\org.e
Processed: Message 1
Added: Message 1
Added: Message 2
Added: Message 3
Processed: Message 2
Added: Message 4
Processed: Message 3
Added: Message 5
Added: Message 6
Processed: Message 4
Added: Message 7
Added: Message 8
Processed: Message 5
Added: Message 9
Added: Message 10
Processed: Message 6
Added: Message 11
Processed: Message 7
```

**Emergency Patient Tracker** :- Keeps track of emergency patients based on severity and arrival order, and treats them in the correct priority.



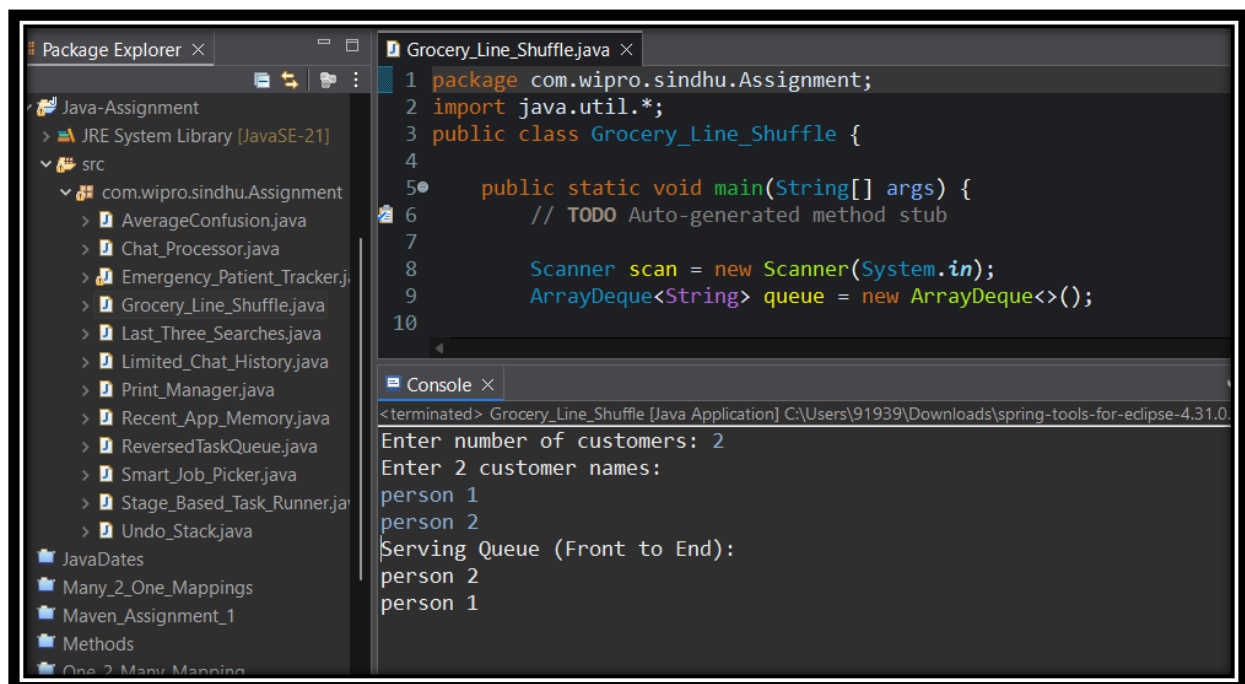
The screenshot shows the Eclipse IDE with the Package Explorer on the left and the Editor on the right. The Package Explorer shows the project structure: Java-Assignment > JRE System Library [JavaSE-21] > src > com.wipro.sindhu.Assignment. The Editor shows the file Emergency\_Patient\_Tracker.java with the following code:

```
4 import java.util.Comparator;
5 import java.util.Scanner;
6
7 class Patient {
8     String name;
9     int severity;
10    long timestamp;
11
12    public Patient(String name, int severity, long timestamp) {
13        this.name = name;
14    }
15 }
```

The Console shows the following output:

```
<terminated> Emergency_Patient_Tracker [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-
Enter max number of patients (S): 4
Enter patient name (or type 'done' to finish): Jyothi
Enter severity (1-10): 4
Added: Patient[Jyothi, Severity: 4]
Enter patient name (or type 'done' to finish): Priya
Enter severity (1-10): 2
Added: Patient[Priya, Severity: 2]
Enter patient name (or type 'done' to finish): sasi
```

**Grocery Line Shuffle** :- A grocery store queue where customers are arranged differently based on whether their name length is even or odd.



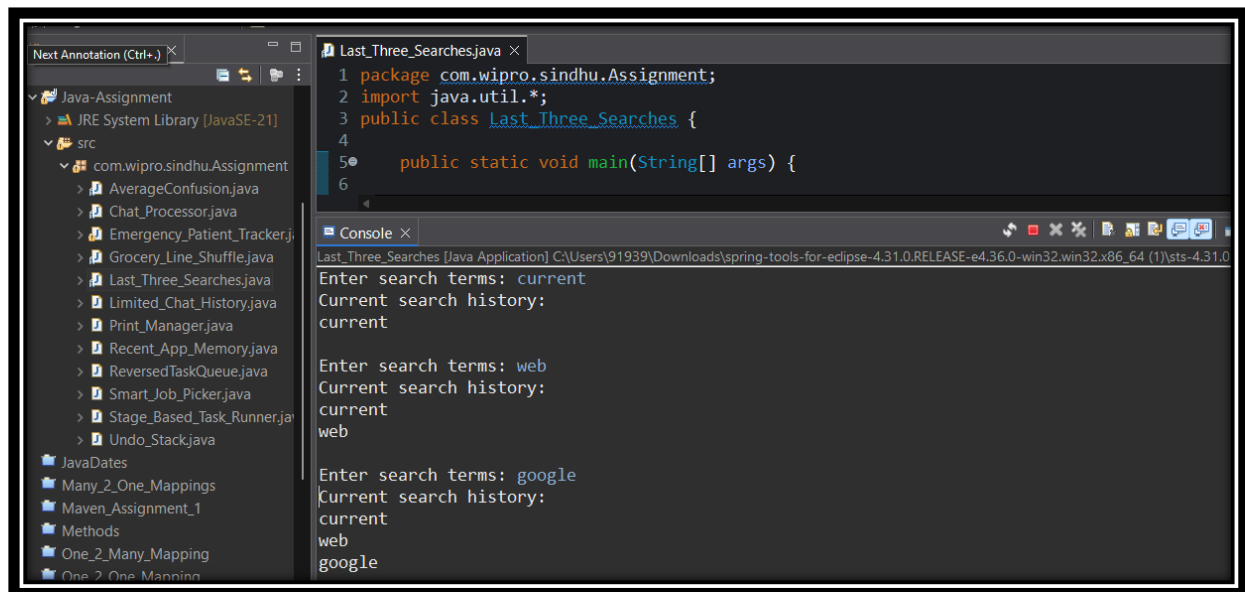
The screenshot shows the Eclipse IDE with the Package Explorer on the left and the Editor on the right. The Package Explorer shows the project structure: Java-Assignment > JRE System Library [JavaSE-21] > src > com.wipro.sindhu.Assignment. The Editor shows the file Grocery\_Line\_Shuffle.java with the following code:

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class Grocery_Line_Shuffle {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         Scanner scan = new Scanner(System.in);
9         ArrayDeque<String> queue = new ArrayDeque<>();
10    }
11 }
```

The Console shows the following output:

```
<terminated> Grocery_Line_Shuffle [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.
Enter number of customers: 2
Enter 2 customer names:
person 1
person 2
Serving Queue (Front to End):
person 2
person 1
```

**Last Three Searches** :- It Keeps track of the last three search terms entered by the user and displays them after each entry.



The screenshot shows an IDE with the 'Last\_Three\_Searches.java' file open. The code defines a class with a main method that prompts the user for search terms and updates a search history. The console output shows three iterations of user input and the resulting search history.

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class Last_Three_Searches {
4
5     public static void main(String[] args) {
6
7         Scanner scan = new Scanner(System.in);
8         ArrayList<String> chatBox = new ArrayList<>();
9
10        System.out.println("Type chat messages (type 'exit' to stop):");
11
12        while (true) {
13            String input = scan.nextLine();
14            if (input.equals("exit")) break;
15            chatBox.add(input);
16            if (chatBox.size() > 3) chatBox.remove(0);
17            System.out.println("Current search history:");
18            for (String term : chatBox) {
19                System.out.println(term);
20            }
21        }
22    }
23 }
```

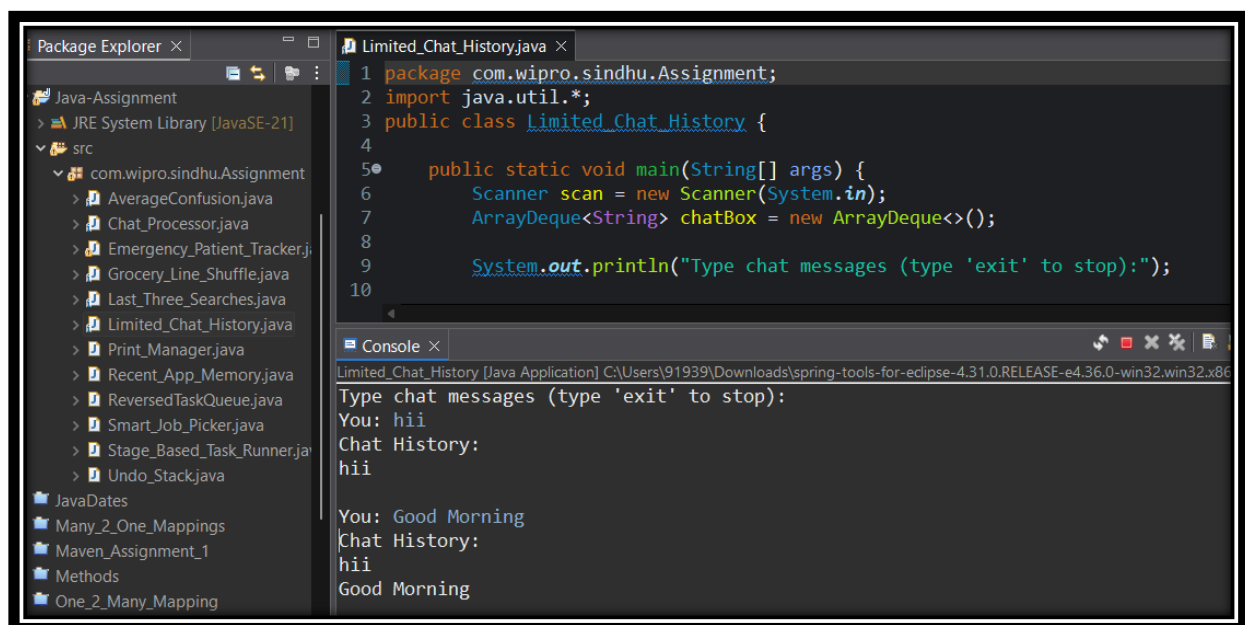
Console Output:

```
Last_Three_Searches [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86_64 (1)\sts-4.31.0
Enter search terms: current
Current search history:
current

Enter search terms: web
Current search history:
current
web

Enter search terms: google
Current search history:
current
web
google
```

**Limited Chat History** :- It Stores and displays only the last 4 chat messages entered by the user.



The screenshot shows an IDE with the 'Limited\_Chat\_History.java' file open. The code defines a class with a main method that prompts the user for chat messages and updates a chat history. The console output shows three iterations of user input and the resulting chat history.

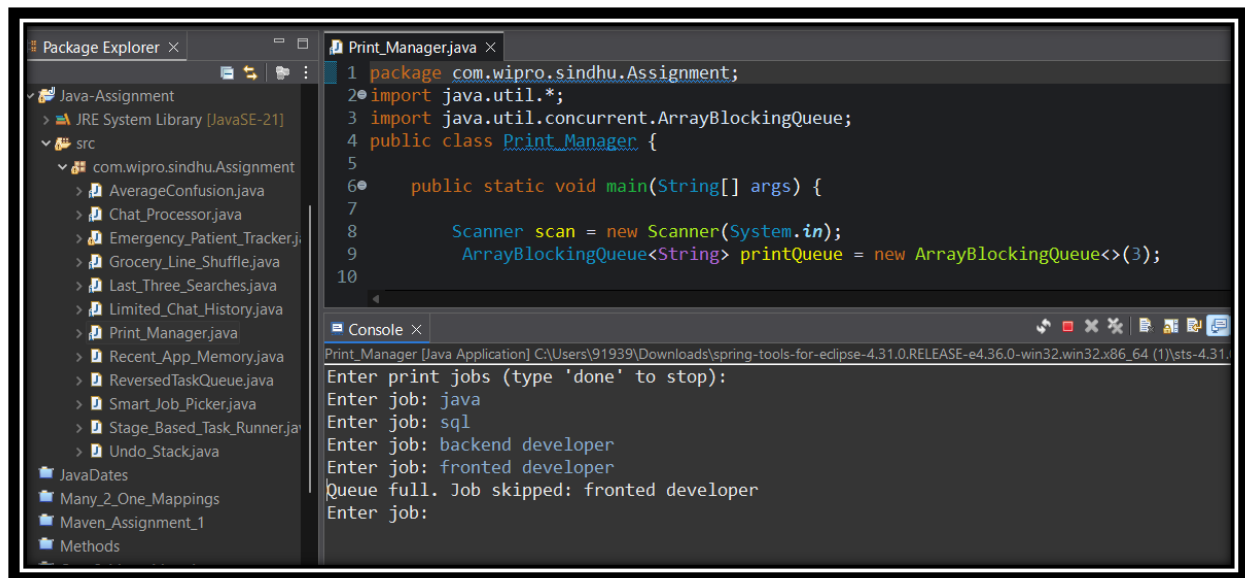
```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class Limited_Chat_History {
4
5     public static void main(String[] args) {
6
7         Scanner scan = new Scanner(System.in);
8         ArrayDeque<String> chatBox = new ArrayDeque<>();
9
10        System.out.println("Type chat messages (type 'exit' to stop):");
11
12        while (true) {
13            String input = scan.nextLine();
14            if (input.equals("exit")) break;
15            chatBox.add(input);
16            if (chatBox.size() > 4) chatBox.remove(0);
17            System.out.println("Chat History:");
18            for (String term : chatBox) {
19                System.out.println(term);
20            }
21        }
22    }
23 }
```

Console Output:

```
Limited_Chat_History [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86_64 (1)\sts-4.31.0
Type chat messages (type 'exit' to stop):
You: hii
Chat History:
hii

You: Good Morning
Chat History:
hii
Good Morning
```

**Print Manager** :- A printer job manager where print jobs are stored in a fixed-size queue.



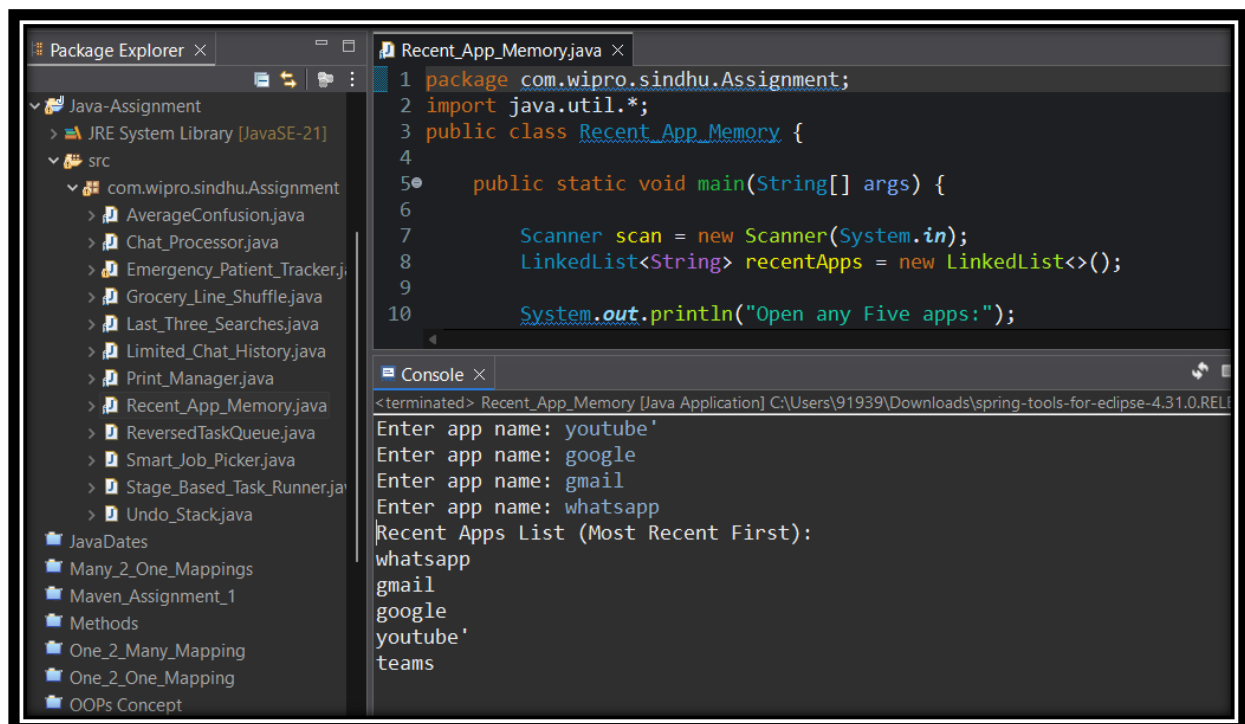
The screenshot shows the Eclipse IDE with the Package Explorer on the left and the Editor on the right. The Package Explorer shows a project named 'Java-Assignment' with a source folder 'src' containing a package 'com.wipro.sindhu.Assignment'. The Editor displays the 'Print\_Manager.java' file with the following code:

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 import java.util.concurrent.ArrayBlockingQueue;
4 public class Print_Manager {
5
6     public static void main(String[] args) {
7
8         Scanner scan = new Scanner(System.in);
9         ArrayBlockingQueue<String> printQueue = new ArrayBlockingQueue<>(3);
10    }
```

The Console window at the bottom shows the output of the program:

```
Print_Manager [Java Application] C:\Users\91939\Downloads\spring-tools-for-ecclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86_64 (1)\sts-4.31.
Enter print jobs (type 'done' to stop):
Enter job: java
Enter job: sql
Enter job: backend developer
Enter job: fronted developer
Queue full. Job skipped: fronted developer
Enter job:
```

**Recent App Memory** :- The Recent Apps feature in mobile operating systems, where the most recently opened app appears at the top and duplicates are removed.



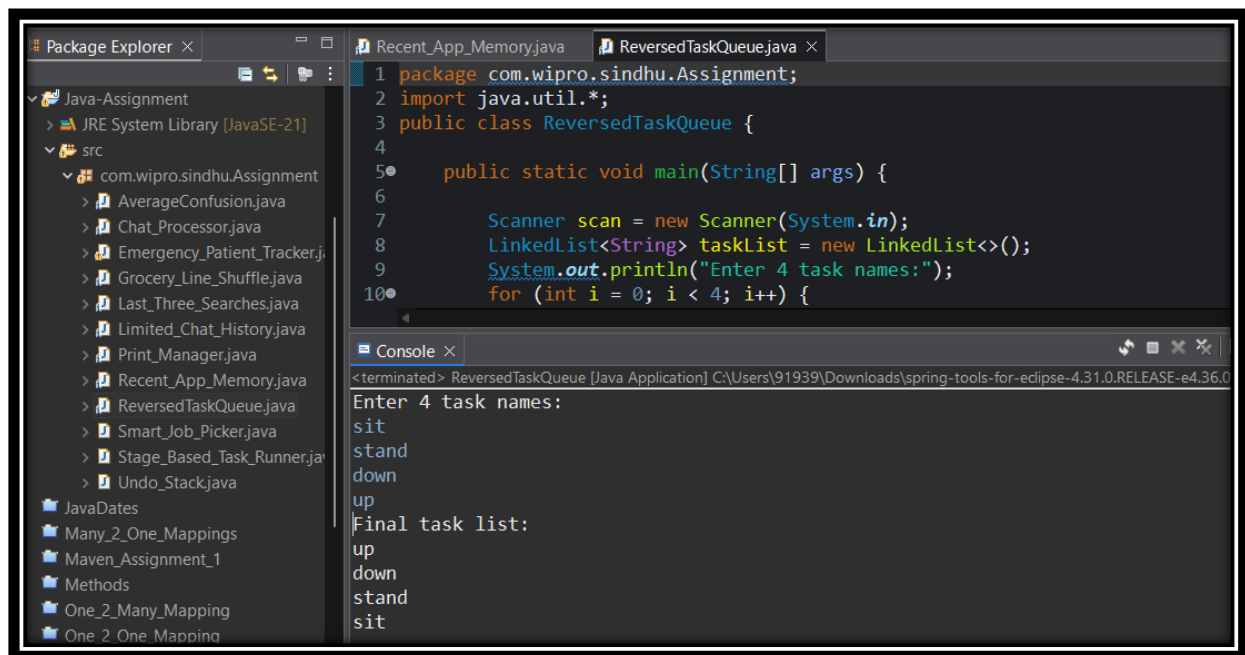
The screenshot shows the Eclipse IDE with the Package Explorer on the left and the Editor on the right. The Package Explorer shows a project named 'Java-Assignment' with a source folder 'src' containing a package 'com.wipro.sindhu.Assignment'. The Editor displays the 'Recent\_App\_Memory.java' file with the following code:

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class Recent_App_Memory {
4
5     public static void main(String[] args) {
6
7         Scanner scan = new Scanner(System.in);
8         LinkedList<String> recentApps = new LinkedList<>();
9
10        System.out.println("Open any Five apps:");
11    }
```

The Console window at the bottom shows the output of the program:

```
<terminated> Recent_App_Memory [Java Application] C:\Users\91939\Downloads\spring-tools-for-ecclipse-4.31.0.REL
Enter app name: youtube'
Enter app name: google
Enter app name: gmail
Enter app name: whatsapp
Recent Apps List (Most Recent First):
whatsapp
gmail
google
youtube'
teams
```

**ReversedTaskQueue** :- It Takes 4 task names from the user and stores them in a linked list, placing each task at the front or back depending on its ending.



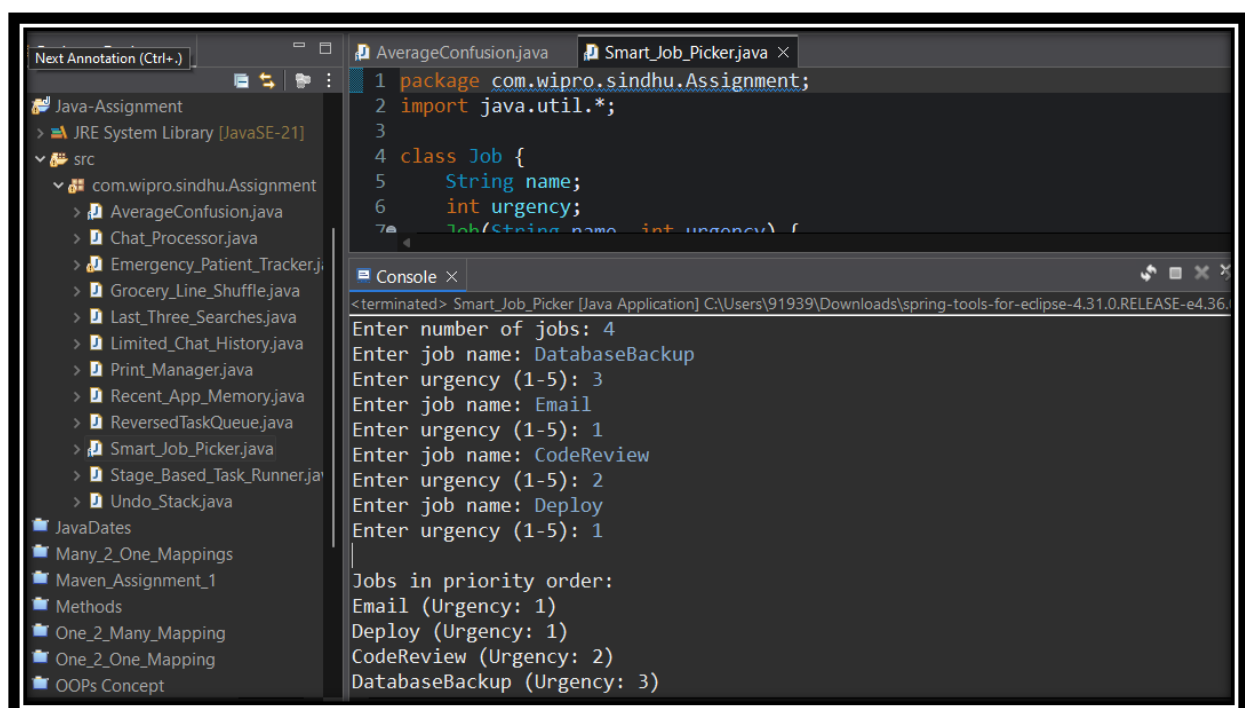
The screenshot shows the Eclipse IDE with the `ReversedTaskQueue.java` file open. The code defines a package `com.wipro.sindhu.Assignment`, imports `java.util.*`, and creates a `ReversedTaskQueue` class. The `main` method uses a `Scanner` to read 4 task names from the user and stores them in a `LinkedList`. The console output shows the user entering the task names: `sit`, `stand`, `down`, and `up`. The final task list is printed as: `up`, `down`, `stand`, and `sit`.

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3 public class ReversedTaskQueue {
4
5     public static void main(String[] args) {
6
7         Scanner scan = new Scanner(System.in);
8         LinkedList<String> taskList = new LinkedList<>();
9         System.out.println("Enter 4 task names:");
10        for (int i = 0; i < 4; i++) {
```

Console Output:

```
<terminated> ReversedTaskQueue [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0
Enter 4 task names:
sit
stand
down
up
Final task list:
up
down
stand
sit
```

**Smart Job Picker** :- Takes a list of jobs from the user, each with a name and urgency level, then prints them in priority order.



The screenshot shows the Eclipse IDE with the `SmartJobPicker.java` file open. The code defines a package `com.wipro.sindhu.Assignment`, imports `java.util.*`, and creates a `Job` class with attributes `String name` and `int urgency`. The `main` method uses a `Scanner` to read the number of jobs and their names and urgency levels. The console output shows the user entering 4 jobs: `DatabaseBackup` (urgency 3), `Email` (urgency 1), `CodeReview` (urgency 2), and `Deploy` (urgency 1). The jobs are printed in priority order: `Email (Urgency: 1)`, `Deploy (Urgency: 1)`, `CodeReview (Urgency: 2)`, and `DatabaseBackup (Urgency: 3)`.

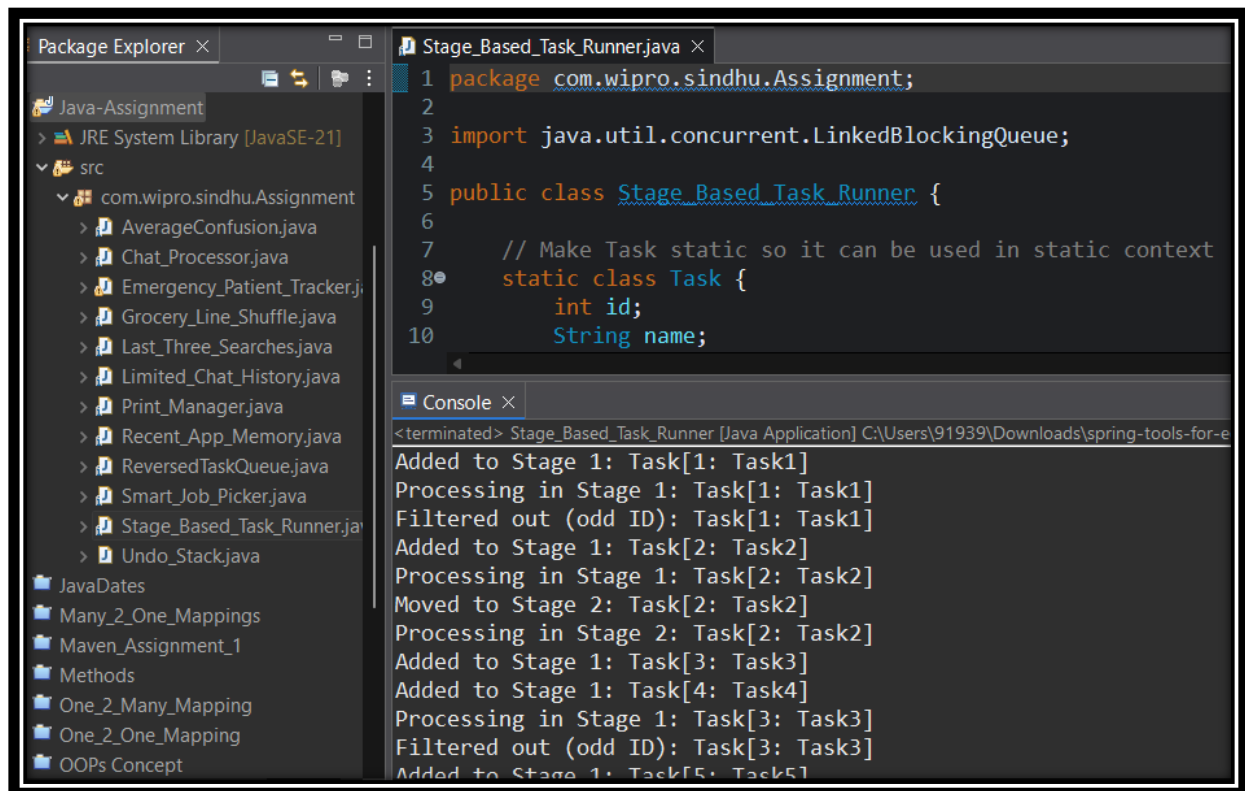
```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3
4 class Job {
5     String name;
6     int urgency;
7     Job(String name, int urgency) {
```

Console Output:

```
<terminated> SmartJobPicker [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0
Enter number of jobs: 4
Enter job name: DatabaseBackup
Enter urgency (1-5): 3
Enter job name: Email
Enter urgency (1-5): 1
Enter job name: CodeReview
Enter urgency (1-5): 2
Enter job name: Deploy
Enter urgency (1-5): 1

Jobs in priority order:
Email (Urgency: 1)
Deploy (Urgency: 1)
CodeReview (Urgency: 2)
DatabaseBackup (Urgency: 3)
```

**Stage Based Task Runner** :- Tasks are produced. Certain tasks are filtered and moved to another queue.



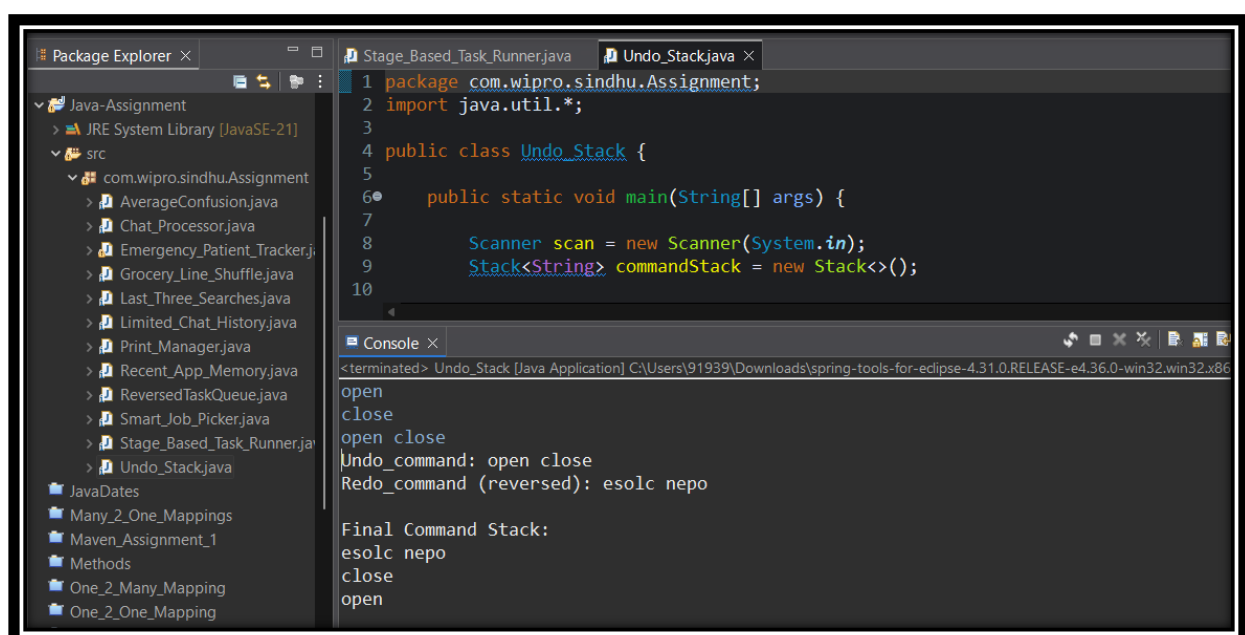
The screenshot shows the Eclipse IDE with the Package Explorer on the left and the editor on the right. The Package Explorer shows a project named 'Java-Assignment' with a source folder 'src' containing a package 'com.wipro.sindhu.Assignment'. The editor displays the file 'Stage\_Based\_Task\_Runner.java' with the following code:

```
1 package com.wipro.sindhu.Assignment;
2
3 import java.util.concurrent.LinkedBlockingQueue;
4
5 public class Stage_Based_Task_Runner {
6
7     // Make Task static so it can be used in static context
8     static class Task {
9         int id;
10        String name;
```

The Console window at the bottom shows the output of the application:

```
<terminated> Stage_Based_Task_Runner [Java Application] C:\Users\91939\Downloads\spring-tools-for-e
Added to Stage 1: Task[1: Task1]
Processing in Stage 1: Task[1: Task1]
Filtered out (odd ID): Task[1: Task1]
Added to Stage 1: Task[2: Task2]
Processing in Stage 1: Task[2: Task2]
Moved to Stage 2: Task[2: Task2]
Processing in Stage 2: Task[2: Task2]
Added to Stage 1: Task[3: Task3]
Added to Stage 1: Task[4: Task4]
Processing in Stage 1: Task[3: Task3]
Filtered out (odd ID): Task[3: Task3]
Added to Stage 1: Task[5: Task5]
```

**Undo Stack** :- To store commands in a stack. Perform Undo by removing the last command. Perform Redo by reversing that undone command and adding it back.



The screenshot shows the Eclipse IDE with the Package Explorer on the left and the editor on the right. The Package Explorer shows a project named 'Java-Assignment' with a source folder 'src' containing a package 'com.wipro.sindhu.Assignment'. The editor displays the file 'Undo\_Stack.java' with the following code:

```
1 package com.wipro.sindhu.Assignment;
2 import java.util.*;
3
4 public class Undo_Stack {
5
6     public static void main(String[] args) {
7
8         Scanner scan = new Scanner(System.in);
9         Stack<String> commandStack = new Stack<>();
10    }
```

The Console window at the bottom shows the output of the application:

```
<terminated> Undo_Stack [Java Application] C:\Users\91939\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x86
open
close
open close
Undo_command: open close
Redo_command (reversed): esolc nepo

Final Command Stack:
esolc nepo
close
open
```