



PAMANTASAN NG LUNGSOD NG MAYNILA

College of Information Systems and Technology Management (CISTM)

ICC 0104-1 – Data Structures and Algorithms

A.Y. 2023- 2024

Group 4: Searching

Submitted by:

Abundo, Jonalene Ryza B.

Dela Peña, Daniella Mae N.

Diamzon, Momer Ailes M.

Lau, Trisha Mae R.

Mahusay, Lindsay G.

Matanga, Sophia Vien V.

Rivera, Ramyll C.

Sibayan, Joan F.

I. Main Menu

Source Code in Java

```
J Singly.java > Singly > scanner
1  import java.util.Scanner;
2
3  class Node {
4      String studno;
5      String studname;
6      String studcrsyr;
7      float GWA;
8      Node next;
9
10     public Node(String studno, String studname, String studcrsyr, float GWA) {
11         this.studno = studno;
12         this.studname = studname;
13         this.studcrsyr = studcrsyr;
14         this.GWA = GWA;
15         this.next = null;
16     }
17 }
18
```

```
19 public class Singly {
20     Node start;
21     Scanner scanner = new Scanner(System.in);
22
23     Run | Debug
24     public static void main(String[] args) {
25         Singly srm = new Singly();
26         int choice;
27
28         while (true) {
29             srm.MainMenu();
30             System.out.print(s:"Enter your choice: ");
31             choice = srm.scanner.nextInt();
32             srm.scanner.nextLine();
33
34             switch (choice) {
35                 case 1:
36                     srm.CreateList();
37                     break;
38                 case 2:
39                     srm.DisplayList();
40                     break;
41                 case 3:
42                     srm.AddStart();
43                     System.out.println(x:"New record is added!");
44                     break;
45                 case 4:
46                     srm.AddEnd();
47                     System.out.println(x:"New record is added!");
48                     break;
49                 case 5:
50                     srm.AddBef();
51                     System.out.println(x:"New record is added!");
52                     break;
53             }
54         }
55     }
56 }
```

```

52         case 6:
53             srm.AddAft();
54             System.out.println(x:"New record is added!");
55             break;
56         case 7:
57             srm.DelStart();
58             System.out.println(x:"Successfully deleted!");
59             break;
60         case 8:
61             srm.DelEnd();
62             System.out.println(x:"Successfully deleted!");
63             break;
64         case 9:
65             srm.DelVal();
66             break;
67         case 10:
68             srm.scanner.close();
69             return;
70         default:
71             System.out.println(x:"Invalid Choice!!!");
72             break;
73     }
74 }
75 }
76

```

```

77 void MainMenu() {
78     System.out.println(x:"\n----- MAIN MENU -----");
79     System.out.println(x:" [1] Create a Student Record");
80     System.out.println(x:" [2] Display Student Record");
81     System.out.println(x:" [3] Add a New Record at the Start");
82     System.out.println(x:" [4] Add a New Record at the End");
83     System.out.println(x:" [5] Add/Insert a New Record Before a Data");
84     System.out.println(x:" [6] Add/Insert a New Record After a Data");
85     System.out.println(x:" [7] Deletion of Node at the Start");
86     System.out.println(x:" [8] Deletion of Node at the End");
87     System.out.println(x:" [9] Deletion of Node by Value");
88     System.out.println(x:" [10] Exit");
89     System.out.println(x:"-----\n");
90 }
91

```

Output

```
----- MAIN MENU -----  
[1] Create a Student Record  
[2] Display Student Record  
[3] Add a New Record at the Start  
[4] Add a New Record at the End  
[5] Add/Insert a New Record Before a Data  
[6] Add/Insert a New Record After a Data  
[7] Deletion of Node at the Start  
[8] Deletion of Node at the End  
[9] Deletion of Node by Value  
[10] Exit  
-----  
Enter your choice: █
```

II. Creation of Single/Singly Linked List

Source Code in Java

```
92  void Createlist() {
93      Node temp = null;
94      char choice;
95
96      do {
97          System.out.println(x:"\n----- CREATING A STUDENT RECORD ----- \n");
98          System.out.print(s:"Student Number: ");
99          String studno = scanner.nextLine();
100          System.out.print(s:"Student Name: ");
101          String studname = scanner.nextLine();
102          System.out.print(s:"Course and Year: ");
103          String studcrsyr = scanner.nextLine();
104          System.out.print(s:"GWA: ");
105          float GWA = scanner.nextFloat();
106          scanner.nextLine();
107
108          Node newNode = new Node(studno, studname, studcrsyr, GWA);
109
110          if (start == null) {
111              start = newNode;
112          } else {
113              temp.next = newNode;
114          }
115
116          temp = newNode;
117
118          System.out.print(s:"Add another record? [Y/N]: ");
119          choice = scanner.nextLine().charAt(index:0);
120      } while (Character.toUpperCase(choice) == 'Y');
121  }
122
```

Output

```
----- CREATING A STUDENT RECORD -----
Student Number: 202334067
Student Name: Ramy11
Course and Year: BSCS 1-1
GWA: 1.00
Add another record? [Y/N]: Y

----- CREATING A STUDENT RECORD -----
Student Number: 202334080
Student Name: Ram
Course and Year: BSCS 1-2
GWA: 1.25
Add another record? [Y/N]: N
```

III. Traversal of Single/Singly Linked List

Source Code in Java

```
123 void DisplayList() {
124     Node current = start;
125     int ctr = 1;
126
127     System.out.println(x:"\n----- STUDENT'S GWA REPORT -----");
128
129     if (current == null) {
130         System.out.println(x:"\nStudent Record is empty!");
131     } else {
132         System.out.printf(format:"%-15s%-25s%-25s%-15s\n", ...args:"Student No.", "Student Name", "Course & Year", "GWA");
133
134         while (current != null) {
135             System.out.printf(format:"%-15s%-25s%-25s%-15.2f\n", current.studno, current.studname, current.studcrsyr, current.GWA);
136             current = current.next;
137             ctr++;
138         }
139     }
140
141     System.out.println(x:"-----\n");
142 }

144 void AddStart() {
145     System.out.println(x:"\n----- ADDING A NEW RECORD AT THE START -----");
146     System.out.print(s:"Student Number: ");
147     String studno = scanner.nextLine();
148     System.out.print(s:"Student Name: ");
149     String studname = scanner.nextLine();
150     System.out.print(s:"Course & Year: ");
151     String studcrsyr = scanner.nextLine();
152     System.out.print(s:"GWA: ");
153     float GWA = scanner.nextFloat();
154     scanner.nextLine();
155
156     Node newNode = new Node(studno, studname, studcrsyr, GWA);
157     newNode.next = start;
158     start = newNode;
159 }
160
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.    Student Name    Course & Year    GWA
202334067      Ramyll          BSCS 1-1        1.00
202334080      Ram             BSCS 1-2        1.25
-----
```

IV. Adding a New Node at the Start

Source Code in Java

```
144 void AddStart() {
145     System.out.println(x:"\n----- ADDING A NEW RECORD AT THE START -----\n");
146     System.out.print(s:"Student Number: ");
147     String studno = scanner.nextLine();
148     System.out.print(s:"Student Name: ");
149     String studname = scanner.nextLine();
150     System.out.print(s:"Course & Year: ");
151     String studcrsyr = scanner.nextLine();
152     System.out.print(s:"GWA: ");
153     float GWA = scanner.nextFloat();
154     scanner.nextLine();
155
156     Node newNode = new Node(studno, studname, studcrsyr, GWA);
157     newNode.next = start;
158     start = newNode;
159 }
160
```

Output:

```
----- ADDING A NEW RECORD AT THE START -----

Student Number: 202334057
Student Name: Lly
Course & Year: BSCS 1-3
GWA: 1.25
New record is added!
```

```
----- STUDENT'S GWA REPORT -----

Student No.   Student Name   Course & Year   GWA
202334057     Lly            BSCS 1-3       1.25
202334067     Ramy11        BSCS 1-1       1.00
202334080     Ram           BSCS 1-2       1.00
-----
```

V. Adding a New Node at the End

Source Code in Java

```
161 void AddEnd() {
162     System.out.println(x:"\n----- ADDING A NEW RECORD AT THE END ----- \n");
163     System.out.print(s:"Student Number: ");
164     String studno = scanner.nextLine();
165     System.out.print(s:"Student Name: ");
166     String studname = scanner.nextLine();
167     System.out.print(s:"Course & Year: ");
168     String studcrsyr = scanner.nextLine();
169     System.out.print(s:"GWA: ");
170     float GWA = scanner.nextFloat();
171     scanner.nextLine();
172
173     Node newNode = new Node(studno, studname, studcrsyr, GWA);
174     if (start == null) {
175         start = newNode;
176     } else {
177         Node temp = start;
178         while (temp.next != null) {
179             temp = temp.next;
180         }
181         temp.next = newNode;
182     }
183 }
184
```

Output

```
----- ADDING A NEW RECORD AT THE END -----

Student Number: 202334057
Student Name: Lly
Course & Year: BSCS 1-3
GWA: 1.25
New record is added!
```



```
----- STUDENT'S GWA REPORT -----

Student No.   Student Name   Course & Year   GWA
202334067     Ramy11        BSCS 1-1       1.00
202334080     Ram           BSCS 1-2       1.00
202334057     Lly           BSCS 1-3       1.25
-----
```


VI. Adding/Insertion of New Node Before a Node (data)

Source Code in Java

```
185 void AddBef() {
186     System.out.println(x:"\n----- ADDING A NEW RECORD BEFORE A GIVEN STUDENT RECORD ----- \n");
187
188     if (start == null) {
189         System.out.println(x:"Student Record is empty!");
190         return;
191     }
192
193     System.out.print(s:"Enter the Student Number before which you want to add the new record: ");
194     String studentNoToFind = scanner.nextLine();
195
196     Node current = start;
197     Node prev = null;
198
199     while (current != null && !current.studno.equals(studentNoToFind)) {
200         prev = current;
201         current = current.next;
202     }
203
204     if (current == null) {
205         System.out.println(x:"Error: The specified student record not found.");
206         return;
207     }
208
209     System.out.println(x:"\n----- INSERT THE NEW RECORD -----");
210     System.out.print(s:"Student Number: ");
211     String studno = scanner.nextLine();
212     System.out.print(s:"Student Name: ");
213     String studname = scanner.nextLine();
214     System.out.print(s:"Course and Year: ");
215     String studcrsyr = scanner.nextLine();
216     System.out.print(s:"GWA: ");
217     float GWA = scanner.nextFloat();
218     scanner.nextLine();
219
220     Node newRecord = new Node(studno, studname, studcrsyr, GWA);
221     if (prev == null) {
222         newRecord.next = start;
223         start = newRecord;
224     } else {
225         newRecord.next = current;
226         prev.next = newRecord;
227     }
228
229     System.out.println(x:"New Student Record has been added before the specified Student Number.");
230 }
231 }
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.    Student Name    Course & Year    GWA
202334067      Ramyll          BSCS 1-1         1.00
202334080      Ram             BSCS 1-2         1.00
-----
```

----- ADDING A NEW RECORD BEFORE A GIVEN STUDENT RECORD -----

Enter the Student Number before which you want to add the new record: 202334080

----- INSERT THE NEW RECORD -----

Student Number: 202334057

Student Name: Lly

Course and Year: BSCS 1-3

GWA: 1.25

New Student Record has been added before the specified Student Number.

New record is added!

----- STUDENT'S GWA REPORT -----

Student No.	Student Name	Course & Year	GWA
202334067	Ramyll	BSCS 1-1	1.00
202334057	Lly	BSCS 1-3	1.25
202334080	Ram	BSCS 1-2	1.00

VII. Adding/Insertion of New Node After a Node (data)

Source Code in Java

```
232 void AddAft() {
233     System.out.println(x:"\n----- ADDING A NEW RECORD AFTER A GIVEN STUDENT RECORD ----- \n");
234
235     if (start == null) {
236         System.out.println(x:"Student Record is empty!");
237         return;
238     }
239
240     System.out.print(s:"Enter the Student Number after which you want to add the new record: ");
241     String studentNoToFind = scanner.nextLine();
242
243     Node current = start;
244
245     while (current != null && !current.studno.equals(studentNoToFind)) {
246         current = current.next;
247     }
248
249     if (current == null) {
250         System.out.println(x:"Error: The specified student record not found.");
251         return;
252     }
253
254     System.out.println(x:"\n----- INSERT THE NEW RECORD -----");
255     System.out.print(s:"Student Number: ");
256     String studno = scanner.nextLine();
257     System.out.print(s:"Student Name: ");
258     String studname = scanner.nextLine();
259     System.out.print(s:"Course and Year: ");
260     String studcrsyr = scanner.nextLine();
261     System.out.print(s:"GWA: ");
262     float GWA = scanner.nextFloat();
263     scanner.nextLine();
264
265     Node newRecord = new Node(studno, studname, studcrsyr, GWA);
266     newRecord.next = current.next;
267     current.next = newRecord;
268
269     System.out.println(x:"New Student Record has been added after the specified Student Number.");
270 }
271 }
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.    Student Name    Course & Year    GWA
202334067      Ramyyl         BSCS 1-1        1.00
202334080      Ram            BSCS 1-2        1.25
-----

----- ADDING A NEW RECORD AFTER A GIVEN STUDENT RECORD -----

Enter the Student Number after which you want to add the new record: 202334067

----- INSERT THE NEW RECORD -----
Student Number: 202334057
Student Name: Lly
Course and Year: BSCS 1-3
GWA: 1.50
New Student Record has been added after the specified Student Number.
New record is added!
```

----- STUDENT'S GWA REPORT -----			
Student No.	Student Name	Course & Year	GWA
202334067	Ramyll	BSCS 1-1	1.00
202334057	Lly	BSCS 1-3	1.50
202334080	Ram	BSCS 1-2	1.25

VIII. Deletion of Node at the Start

Source Code in Java

```
272 void DelStart() {
273     System.out.println(x:"\n----- DELETING THE FIRST RECORD ----- \n");
274
275     if (start == null) {
276         System.out.println(x:"Student Record is empty!");
277         return;
278     }
279
280     start = start.next;
281
282     System.out.println(x:"The first student record has been successfully deleted.");
283 }
284
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334067     Ramyll        BSCS 1-1       1.00
202334080     Ram           BSCS 1-2       1.00
202334057     Lly           BSCS 1-3       1.25
-----
```

```
----- DELETING THE FIRST RECORD -----

The first student record has been successfully deleted.
Successfully deleted!
```

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334080     Ram           BSCS 1-2       1.00
202334057     Lly           BSCS 1-3       1.25
-----
```

IX. Deletion of Node at the End

Source Code in Java

```
285 void DelEnd() {
286     System.out.println(x:"\n----- DELETING THE LAST RECORD ----- \n");
287
288     if (start == null) {
289         System.out.println(x:"Student Record is empty!");
290         return;
291     }
292
293     if (start.next == null) {
294         start = null;
295     } else {
296         Node current = start;
297         Node prev = null;
298         while (current.next != null) {
299             prev = current;
300             current = current.next;
301         }
302         prev.next = null;
303     }
304
305     System.out.println(x:"The last student record has been successfully deleted.");
306 }
307
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334080     Ram           BSCS 1-2       1.00
202334057     Lly           BSCS 1-3       1.25
-----
```

```
----- DELETING THE LAST RECORD -----

The last student record has been successfully deleted.
Successfully deleted!
```

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334080     Ram           BSCS 1-2       1.00
-----
```

X. Deletion of Node by Value (Data)

Source Code in Java

```
308 void DelVal() {
309     System.out.println(x:"\n----- DELETING A RECORD BY VALUE ----- \n");
310
311     if (start == null) {
312         System.out.println(x:"Student Record is empty!");
313         return;
314     }
315
316     System.out.print(s:"Enter the Student Number of the record you want to delete: ");
317     String studentNoToDelete = scanner.nextLine();
318
319     Node current = start;
320     Node prev = null;
321
322     while (current != null && !current.studno.equals(studentNoToDelete)) {
323         prev = current;
324         current = current.next;
325     }
326
327     if (current == null) {
328         System.out.println(x:"Error: The specified student record not found.");
329         return;
330     }
331
332     if (prev == null) {
333         start = start.next;
334     } else {
335         prev.next = current.next;
336     }
337
338     System.out.println(x:"The student record has been successfully deleted.");
339 }
340 }
```

Output

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334067    Ramyll        BSCS 1-1       1.00
202334057    Lly           BSCS 1-3       1.50
202334080    Ram           BSCS 1-2       1.25
-----
```

```
----- DELETING A RECORD BY VALUE -----

Enter the Student Number of the record you want to delete: 202334057
The student record has been successfully deleted.
```

```
----- STUDENT'S GWA REPORT -----
Student No.   Student Name   Course & Year   GWA
202334067    Ramyll        BSCS 1-1       1.00
202334080    Ram           BSCS 1-2       1.25
-----
```