

Course: CSC220_02

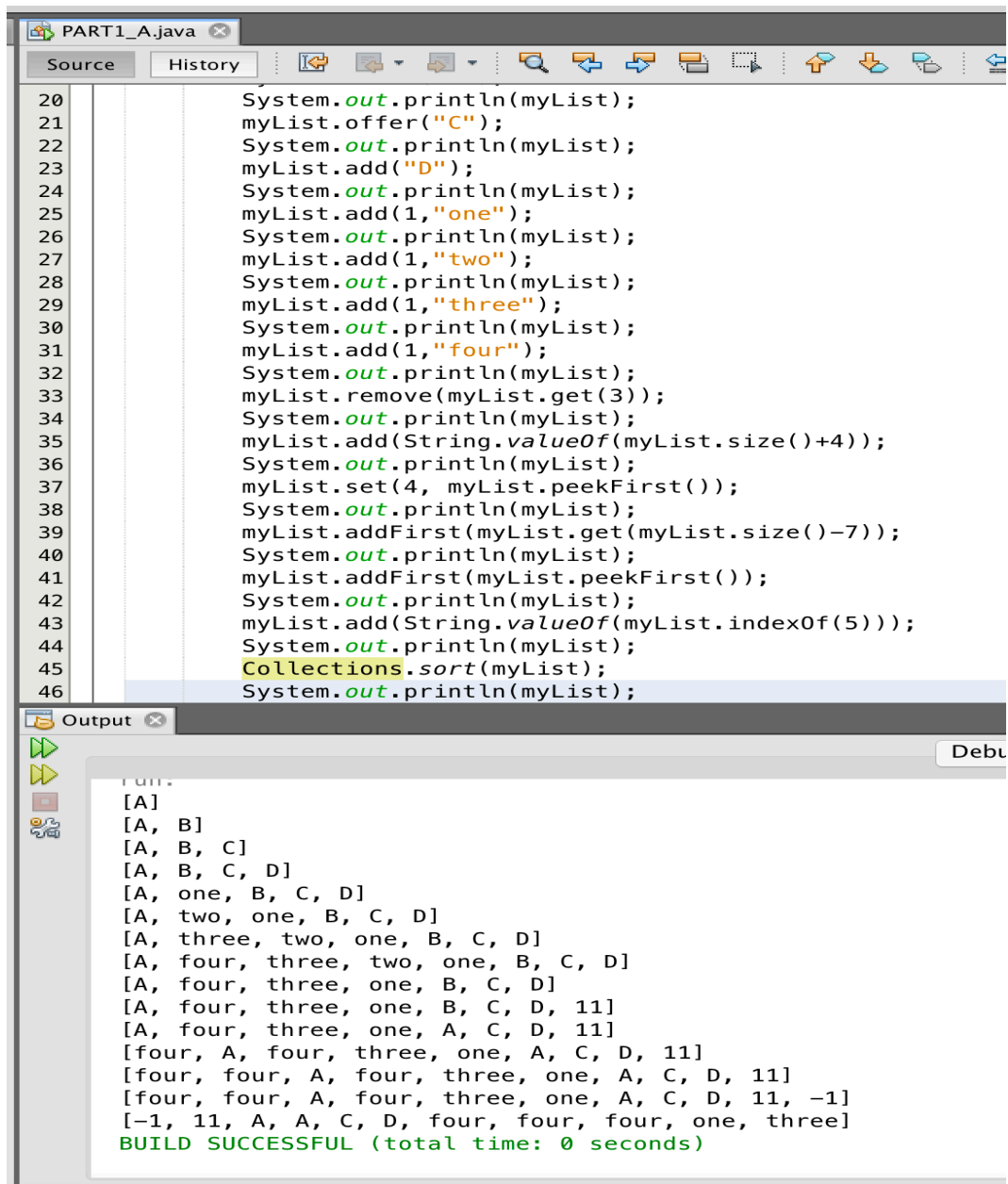
Student: Raya Farshad

Assignment Number 04

Due Date & time: 05-01-2018 at 11:59 PM

PART 1 – Lists, 20 points

A.



The screenshot shows a Java IDE with a file named PART1_A.java. The source code is as follows:

```
20 System.out.println(myList);
21 myList.offer("C");
22 System.out.println(myList);
23 myList.add("D");
24 System.out.println(myList);
25 myList.add(1, "one");
26 System.out.println(myList);
27 myList.add(1, "two");
28 System.out.println(myList);
29 myList.add(1, "three");
30 System.out.println(myList);
31 myList.add(1, "four");
32 System.out.println(myList);
33 myList.remove(myList.get(3));
34 System.out.println(myList);
35 myList.add(String.valueOf(myList.size()+4));
36 System.out.println(myList);
37 myList.set(4, myList.peekFirst());
38 System.out.println(myList);
39 myList.addFirst(myList.get(myList.size()-7));
40 System.out.println(myList);
41 myList.addFirst(myList.peekFirst());
42 System.out.println(myList);
43 myList.add(String.valueOf(myList.indexOf(5)));
44 System.out.println(myList);
45 Collections.sort(myList);
46 System.out.println(myList);
```

The output window shows the following sequence of lists:

```
[A]
[A, B]
[A, B, C]
[A, B, C, D]
[A, one, B, C, D]
[A, two, one, B, C, D]
[A, three, two, one, B, C, D]
[A, four, three, two, one, B, C, D]
[A, four, three, one, B, C, D]
[A, four, three, one, B, C, D, 11]
[A, four, three, one, A, C, D, 11]
[four, A, four, three, one, A, C, D, 11]
[four, four, A, four, three, one, A, C, D, 11]
[four, four, A, four, three, one, A, C, D, 11, -1]
[-1, 11, A, A, C, D, four, four, four, one, three]
BUILD SUCCESSFUL (total time: 0 seconds)
```

We are creating a generic linkedList object myList which stores strings only. We use different methods like add, offer, remove, set from Collection interface.

B. 15 Points

run:

Creating profiles and adding to database.

name: John Doe

Status: My name is John.

Pictures; BufferedImage@53d8d1

#of friends: 0

Friends:

name: Jane Doe

Status: My name is Jane.

Pictures; BufferedImage@e9e54c

#of friends: 0

Friends:

name: Billy Bob

Status: My name is Billy Bob!

Pictures; BufferedImage@65ab77

#of friends: 0

Friends:

name: John Smith

Status: My name is also John.

Pictures; BufferedImage@1b28cd

#of friends: 0

Friends:

Creating friendships.

name: John Doe

Status: My name is John.

Pictures; BufferedImage@53d8d1

#of friends: 2

Friends:

Jane Doe

John Smith

name: Jane Doe

Status: My name is Jane.

Pictures; BufferedImage@e9e54c

#of friends: 3

Friends:

John Doe

Billy Bob

John Smith

name: Billy Bob

Status: My name is Billy Bob!

Pictures; BufferedImage@65ab77

#of friends: 2

Friends:

Jane Doe

John Smith

name: John Smith

Status: My name is also John.

Pictures; BufferedImage@1b28cd

#of friends: 3

Friends:

John Doe

Jane Doe

Billy Bob

Changing statuses / names.

name: John Doe

Status: My name is John.

Pictures; BufferedImage@53d8d1

#of friends: 2

Friends:

Jane Smith

John Smith

name: Jane Smith

Status: Now Mrs. Smith!

Pictures; BufferedImage@e9e54c

#of friends: 3

Friends:

John Doe

Billy Bob

John Smith

name: Billy Bob

Status: My name is Billy Bob!

Pictures; BufferedImage@65ab77

#of friends: 2

Friends:

Jane Smith

John Smith

name: John Smith

Status: Just got married!

Pictures; BufferedImage@1b28cd

#of friends: 3

Friends:

John Doe

Jane Smith

Billy Bob

Ending a friendship.

name: John Doe

Status: My name is John.

Pictures; BufferedImage@53d8d1

#of friends: 2

Friends:

Jane Smith

John Smith

name: Jane Smith

Status: Now Mrs. Smith!

Pictures; BufferedImage@e9e54c

#of friends: 3

Friends:

John Doe

Billy Bob

John Smith

name: Billy Bob

Status: My name is Billy Bob!

Pictures; BufferedImage@65ab77

#of friends: 2

Friends:

Jane Smith

John Smith

name: John Smith

Status: Just got married!

Pictures; BufferedImage@1b28cd

#of friends: 3

Friends:

John Doe

Jane Smith

Billy Bob

Removing John.

name: Jane Smith

Status: Now Mrs. Smith!

Pictures; BufferedImage@e9e54c

#of friends: 2

Friends:

Billy Bob

John Smith

name: Billy Bob

Status: My name is Billy Bob!

Pictures; BufferedImage@65ab77

#of friends: 2

Friends:

Jane Smith

John Smith

name: John Smith

Status: Just got married!

Pictures; BufferedImage@1b28cd

#of friends: 2

Friends:

Jane Smith

Billy Bob

BUILD SUCCESSFUL (total time: 3 seconds)

PART 2- Iterators, 15 points

A.

As we all know, an iterator is an object which implements either the `Iterator` or the `ListIterator` interface.

`Iterator` enable us to cycle through a collection, obtaining or removing elements.

`ListIterator` extends `Iterator` enable us to allow bidirectional traversal of a list, and the modification of elements.

```
import java.util.Iterator;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author rayafarshad
 */
public class PART2_A {

    public static void main(String[] args) {

        //Creat an ArrayList
        List nameList = new ArrayList();
        nameList.add("Kyle");
        nameList.add("Cathy");
        nameList.add("Sam");
        nameList.add("Austin");
        nameList.add("Sara");
        Iterator<String>nameIterator = nameList.iterator();

        System.out.println(nameIterator.next());
        nameIterator.next();
        System.out.println(nameIterator.next());
        nameIterator.remove();
        System.out.println(nameIterator.next());
        System.out.println(nameList);

    }

}
```

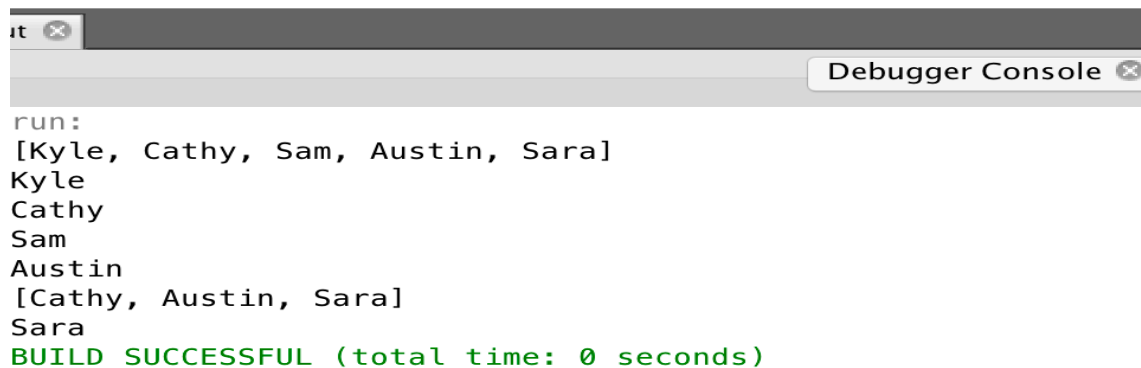
tput x

Debugger Console x asmt04 (run) x

```
run:
Kyle
Sam
Austin
[Kyle, Cathy, Austin, Sara]
BUILD SUCCESSFUL (total time: 0 seconds)
```

- B. As we see here on the very last line, the only item in the list is Sara. As the result on the next line when we call next item in the list the compiler gives us an error cause there is no item in the next().

```
public class PART2_B {  
  
    public static void main(String[] args) {  
  
        List nameList = new ArrayList();  
        nameList.add("Kyle");  
        nameList.add("Cathy");  
        nameList.add("Sam");  
        nameList.add("Austin");  
        nameList.add("Sara");  
        Iterator<String>nameIterator = nameList.iterator();  
  
        System.out.println(nameList);  
        System.out.println(nameIterator.next());  
        nameIterator.remove();  
        System.out.println(nameIterator.next());  
        System.out.println(nameIterator.next());  
        nameIterator.remove();  
        System.out.println(nameIterator.next());  
        System.out.println(nameList);  
        System.out.println(nameIterator.next());  
        // System.out.println(nameIterator.next());  
  
    }  
}
```



```
run:  
[Kyle, Cathy, Sam, Austin, Sara]  
Kyle  
Cathy  
Sam  
Austin  
[Cathy, Austin, Sara]  
Sara  
BUILD SUCCESSFUL (total time: 0 seconds)
```



```

public class PART2_B {

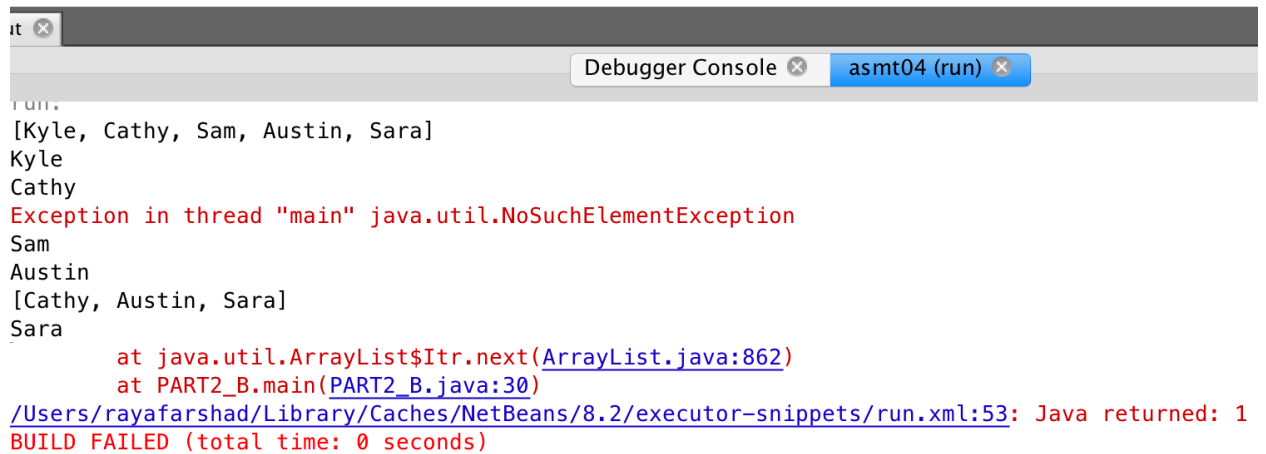
    public static void main(String[] args) {

        List nameList = new ArrayList();
        nameList.add("Kyle");
        nameList.add("Cathy");
        nameList.add("Sam");
        nameList.add("Austin");
        nameList.add("Sara");
        Iterator<String>nameIterator = nameList.iterator();

        System.out.println(nameList);
        System.out.println(nameIterator.next());
        nameIterator.remove();
        System.out.println(nameIterator.next());
        System.out.println(nameIterator.next());
        nameIterator.remove();
        System.out.println(nameIterator.next());
        System.out.println(nameList);
        System.out.println(nameIterator.next());
        System.out.println(nameIterator.next());

    }
}

```



```

Run.
[Kyle, Cathy, Sam, Austin, Sara]
Kyle
Cathy
Exception in thread "main" java.util.NoSuchElementException
Sam
Austin
[Cathy, Austin, Sara]
Sara
    at java.util.ArrayList$Itr.next(ArrayList.java:862)
    at PART2_B.main(PART2_B.java:30)
/Users/rayafarshad/Library/Caches/NetBeans/8.2/executor-snippets/run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)

```

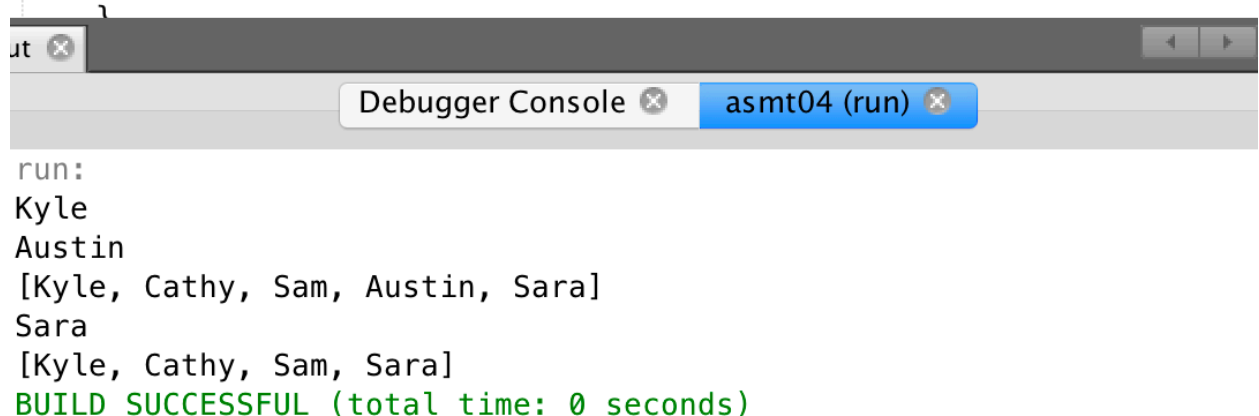
C.

```
import java.util.ListIterator;
import java.util.ArrayList;
import java.util.List;

/**
 * @author rayafarshad
 */
public class PART2_C {

    public static void main(String[] args) {
        List nameList = new ArrayList();
        nameList.add("Kyle");
        nameList.add("Cathy");
        nameList.add("Sam");
        nameList.add("Austin");
        nameList.add("Sara");
        ListIterator<String>nameIterator = nameList.listIterator();

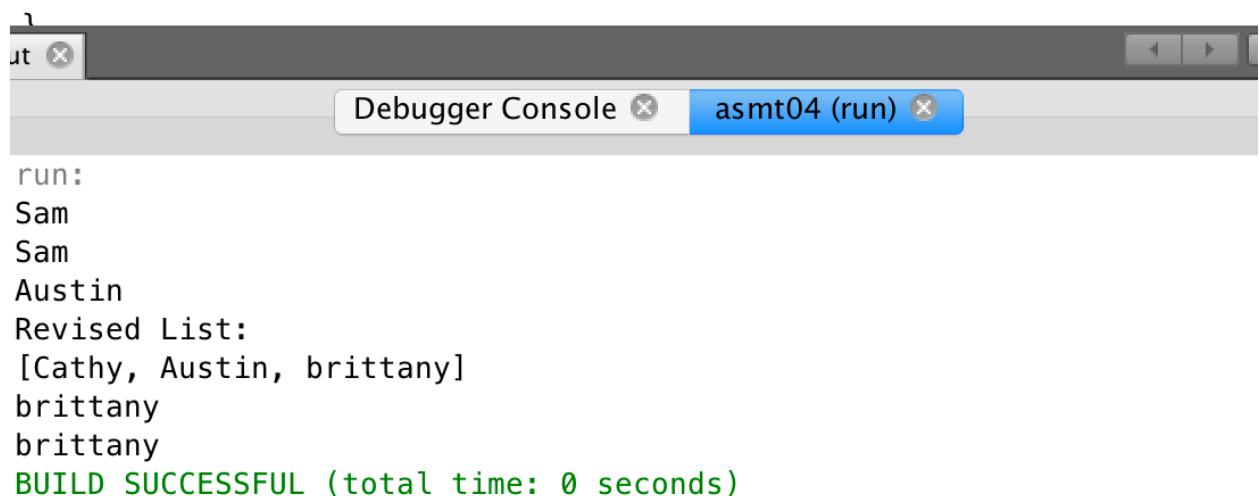
        System.out.println(nameIterator.next());
        nameIterator.next();
        nameIterator.next();
        System.out.println(nameIterator.next());
        // nameIterator.set("Brittany");
        System.out.println(nameList);
        nameIterator.previous();
        nameIterator.remove();
        System.out.println(nameIterator.next());
        System.out.println(nameList);
    }
}
```



```
run:
Kyle
Austin
[Kyle, Cathy, Sam, Austin, Sara]
Sara
[Kyle, Cathy, Sam, Sara]
BUILD SUCCESSFUL (total time: 0 seconds)
```

D.

```
public class PART2_D {  
  
    public static void main(String[] args) {  
  
        List nameList = new ArrayList();  
        nameList.add("Kyle");  
        nameList.add("Cathy");  
        nameList.add("Sam");  
        nameList.add("Austin");  
        nameList.add("Sara");  
        ListIterator<String>nameIterator = nameList.listIterator();  
  
        nameIterator.next();  
        nameIterator.remove();  
        nameIterator.next();  
        System.out.println(nameIterator.next());  
        System.out.println(nameIterator.previous());  
        nameIterator.remove();  
        System.out.println(nameIterator.next());  
        nameIterator.next();  
        nameIterator.set("brittany");  
        System.out.println("Revised List: ");  
        System.out.println(nameList);  
        System.out.println(nameIterator.previous());  
        System.out.println(nameIterator.next());  
  
    }  
}
```



```
run:  
Sam  
Sam  
Austin  
Revised List:  
[Cathy, Austin, brittany]  
brittany  
brittany  
BUILD SUCCESSFUL (total time: 0 seconds)
```

- E. 7 points- Given a list of Strings and an iterator nameIterator whose data type is ListIterator, write statements/code that add the string Bob after the first occurrence of the string Sam. Explain your approach.

I am using Iterator to access the index that has Sam and add Bob to the same index.

```
public class PART2_E {  
  
    public static void main(String[] args) {  
        int pos = 0;  
        List nameList = new ArrayList();  
        nameList.add("Kyle");  
        nameList.add("Cathy");  
        nameList.add("Sam");  
        nameList.add("Austin");  
        nameList.add("Sara");  
        ListIterator<String> nameIterator = nameList.listIterator();  
  
        nameIterator.next();  
        for(int i = 0; i < nameList.size(); i++){  
            if(nameIterator.hasNext()) {  
                nameIterator.next();  
                pos++;  
                if(nameList.get(i).equals("Sam")){  
                    System.out.println(pos-1);  
                }  
            }  
        }  
        nameList.add(pos-1, "Bob");  
        System.out.println(nameList);  
    }  
}
```



```
run:  
2  
[Kyle, Cathy, Sam, Bob, Austin, Sara]  
BUILD SUCCESSFUL (total time: 0 seconds)
```

PART 3 – Dictionaries and Hashing 15 points

```
public class Driver {  
  
    public static void main(String args[]) {  
        PatientRecord test1;  
        PatientRecord test2;  
  
        try {  
            test1 = new PatientRecord(101, 12, 15, 2006, "cough", "bed rest");  
            System.out.println("Patient Record created: " + test1);  
            System.out.println("It has hash code " + test1.hashCode());  
        }  
        catch (BadVisitDateException e) {  
            System.out.println("Creation failed " + e);  
        }  
  
        try {  
            test2 = new PatientRecord(101, 12, 17, 2006, "high fever", "antibiotics");  
            System.out.println("Patient Record created: " + test2);  
            System.out.println("It has hash code " + test2.hashCode());  
        }  
        catch (BadVisitDateException e) {
```



```
run:  
Patient Record created: 101 [12/15/2006] Complaint: cough Prescribed: bed rest  
It has hash code 1190365153  
Patient Record created: 101 [12/17/2006] Complaint: high fever Prescribed: antibiotics  
It has hash code 1247623455  
Creation failed BadVisitDateException: Month not in range 1-12  
Creation failed BadVisitDateException: Day not in range 1-31  
Creation failed BadVisitDateException: Year not greater than 1900  
BUILD SUCCESSFUL (total time: 0 seconds)
```