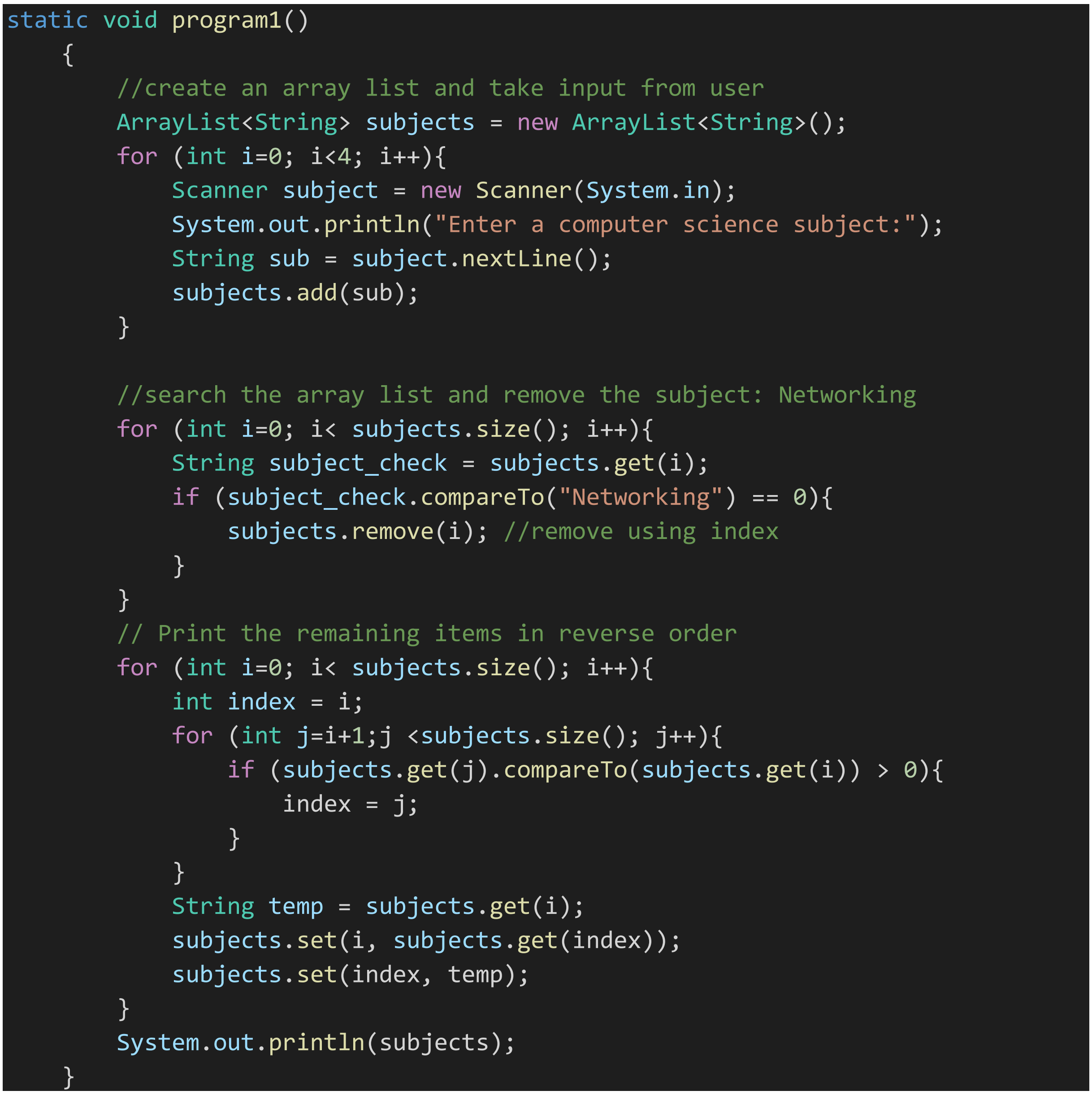
Java Assignment 4

Student number: 121376141

Student name: Peace Samuel

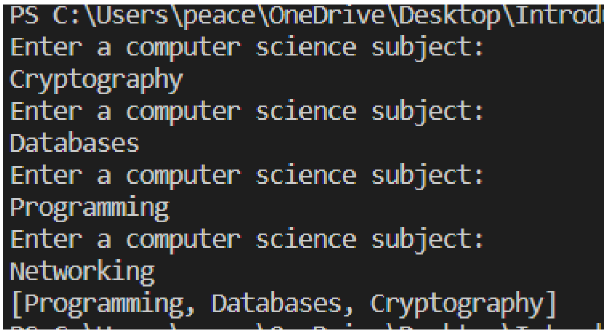
Program1:

Write a program in Java that creates an ArrayList, where it stores the subjects taught in the computer science department <Cryptography, Database, Programming, Networking>, taken as input from the user. Then
perform the following:
Search the ArrayList and remove the subject Networking.
Print the remaining items of the ArrayList in reverse alphabetical order.



```java
static void program1()
    {
        //create an array list and take input from user
        ArrayList<String> subjects = new ArrayList<String>();
        for (int i=0; i<4; i++){
            Scanner subject = new Scanner(System.in);
            System.out.println("Enter a computer science subject:");
            String sub = subject.nextLine();
            subjects.add(sub);
        }

        //search the array list and remove the subject: Networking
        for (int i=0; i< subjects.size(); i++){
            String subject_check = subjects.get(i);
            if (subject_check.compareTo("Networking") == 0){
                subjects.remove(i); //remove using index
            }
        }
        // Print the remaining items in reverse order
        for (int i=0; i< subjects.size(); i++){
            int index = i;
            for (int j=i+1;j <subjects.size(); j++){
                if (subjects.get(j).compareTo(subjects.get(i)) > 0){
                    index = j;
                }
            }
            String temp = subjects.get(i);
            subjects.set(i, subjects.get(index));
            subjects.set(index, temp);
        }
        System.out.println(subjects);
    }
```

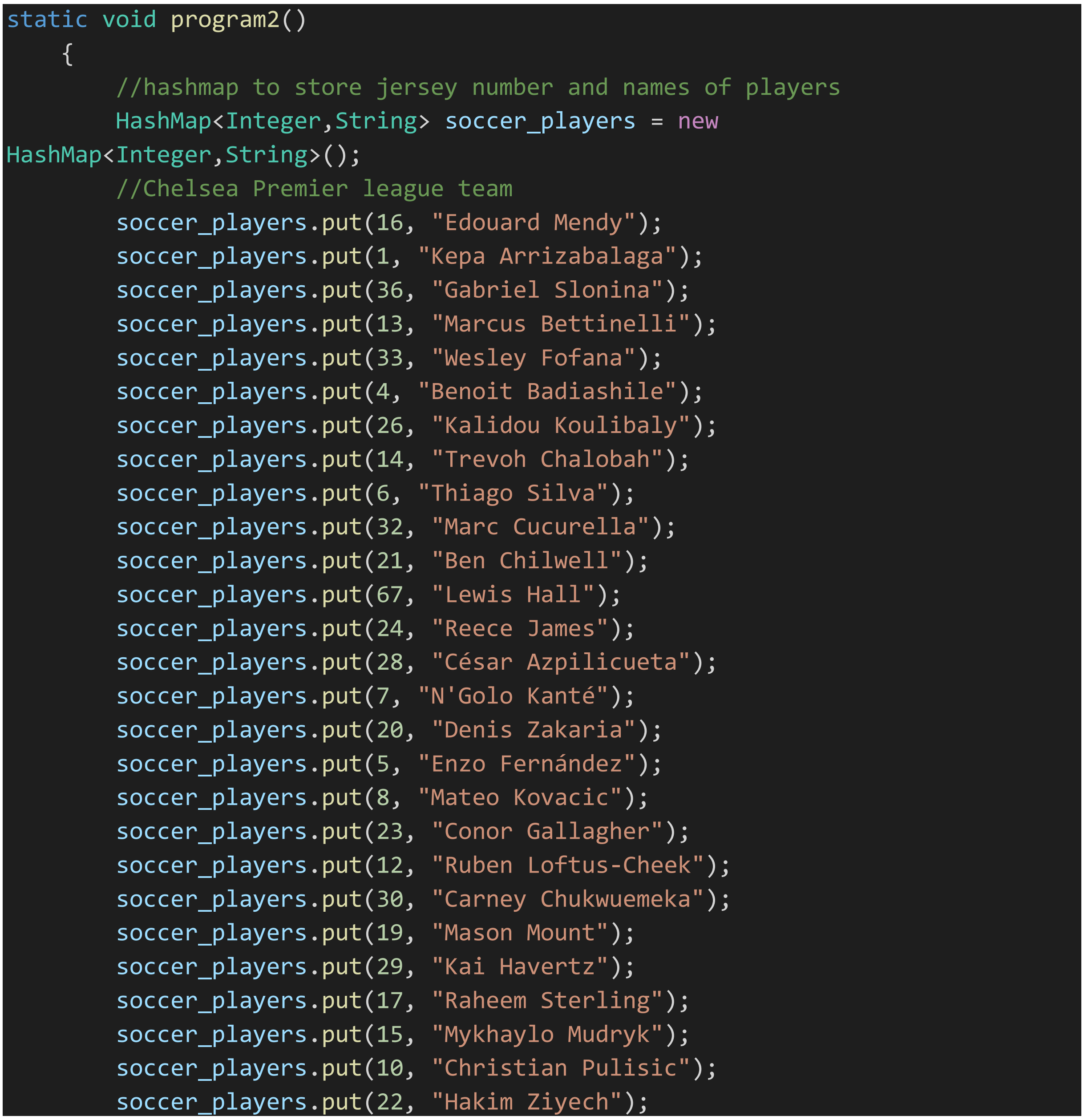
Output:



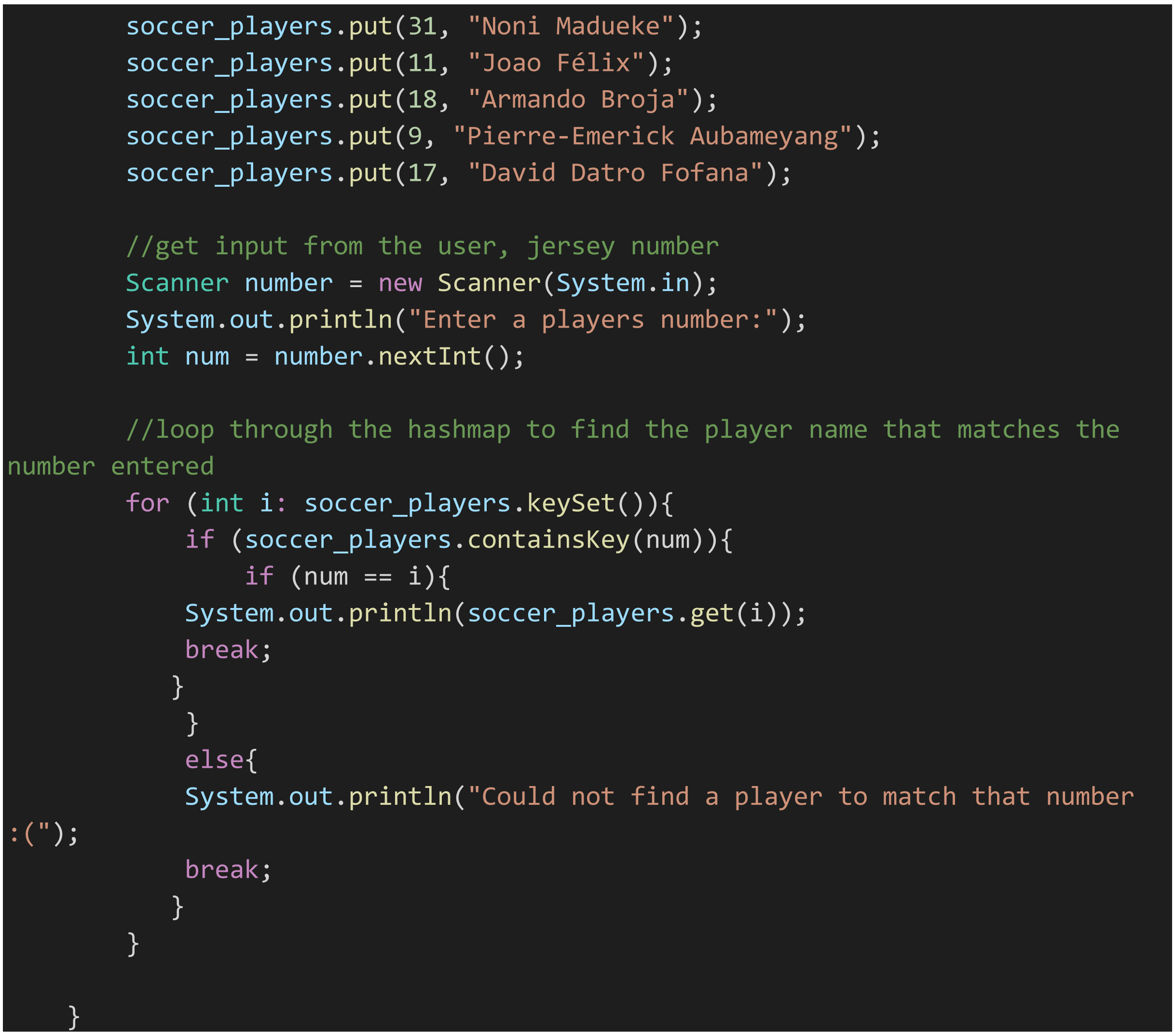
```
PS C:\Users\peace\OneDrive\Desktop\Introd
Enter a computer science subject:
Cryptography
Enter a computer science subject:
Databases
Enter a computer science subject:
Programming
Enter a computer science subject:
Networking
[Programming, Databases, Cryptography]
```

Program2:

Write a program in Java that uses a HashMap to store the jersey number and names of players of a soccer club. After this, the program should be able to output the names of the players, whose jersey number is given as input by the user.

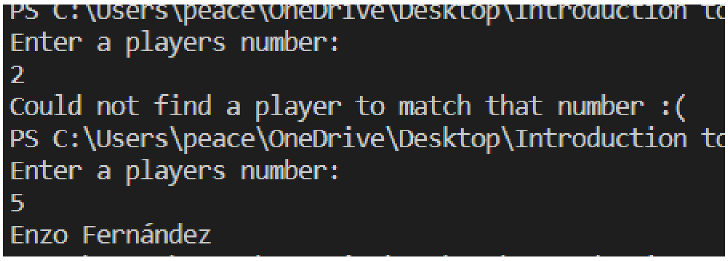


```java
static void program2()
    {
        //hashmap to store jersey number and names of players
        HashMap<Integer,String> soccer_players = new 
HashMap<Integer,String>();
        //Chelsea Premier league team
        soccer_players.put(16, "Edouard Mendy");
        soccer_players.put(1, "Kepa Arrizabalaga");
        soccer_players.put(36, "Gabriel Slonina");
        soccer_players.put(13, "Marcus Bettinelli");
        soccer_players.put(33, "Wesley Fofana");
        soccer_players.put(4, "Benoit Badiashile");
        soccer_players.put(26, "Kalidou Koulibaly");
        soccer_players.put(14, "Trevoh Chalobah");
        soccer_players.put(6, "Thiago Silva");
        soccer_players.put(32, "Marc Cucurella");
        soccer_players.put(21, "Ben Chilwell");
        soccer_players.put(67, "Lewis Hall");
        soccer_players.put(24, "Reece James");
        soccer_players.put(28, "César Azpilicueta");
        soccer_players.put(7, "N'Golo Kanté");
        soccer_players.put(20, "Denis Zakaria");
        soccer_players.put(5, "Enzo Fernández");
        soccer_players.put(8, "Mateo Kovacic");
        soccer_players.put(23, "Conor Gallagher");
        soccer_players.put(12, "Ruben Loftus-Cheek");
        soccer_players.put(30, "Carney Chukwuemeka");
        soccer_players.put(19, "Mason Mount");
        soccer_players.put(29, "Kai Havertz");
        soccer_players.put(17, "Raheem Sterling");
        soccer_players.put(15, "Mykhaylo Mudryk");
        soccer_players.put(10, "Christian Pulisic");
        soccer_players.put(22, "Hakim Ziyech");
```



```
        soccer_players.put(31, "Noni Madueke");
        soccer_players.put(11, "Joao Félix");
        soccer_players.put(18, "Armando Broja");
        soccer_players.put(9, "Pierre-Emerick Aubameyang");
        soccer_players.put(17, "David Datro Fofana");

        //get input from the user, jersey number
        Scanner number = new Scanner(System.in);
        System.out.println("Enter a players number:");
        int num = number.nextInt();

        //loop through the hashmap to find the player name that matches the
number entered
        for (int i: soccer_players.keySet()){
            if (soccer_players.containsKey(num)){
                if (num == i){
            System.out.println(soccer_players.get(i));
            break;
           }
            }
            else{
            System.out.println("Could not find a player to match that number
:(");
            break;
           }
        }

    }
```

Output:

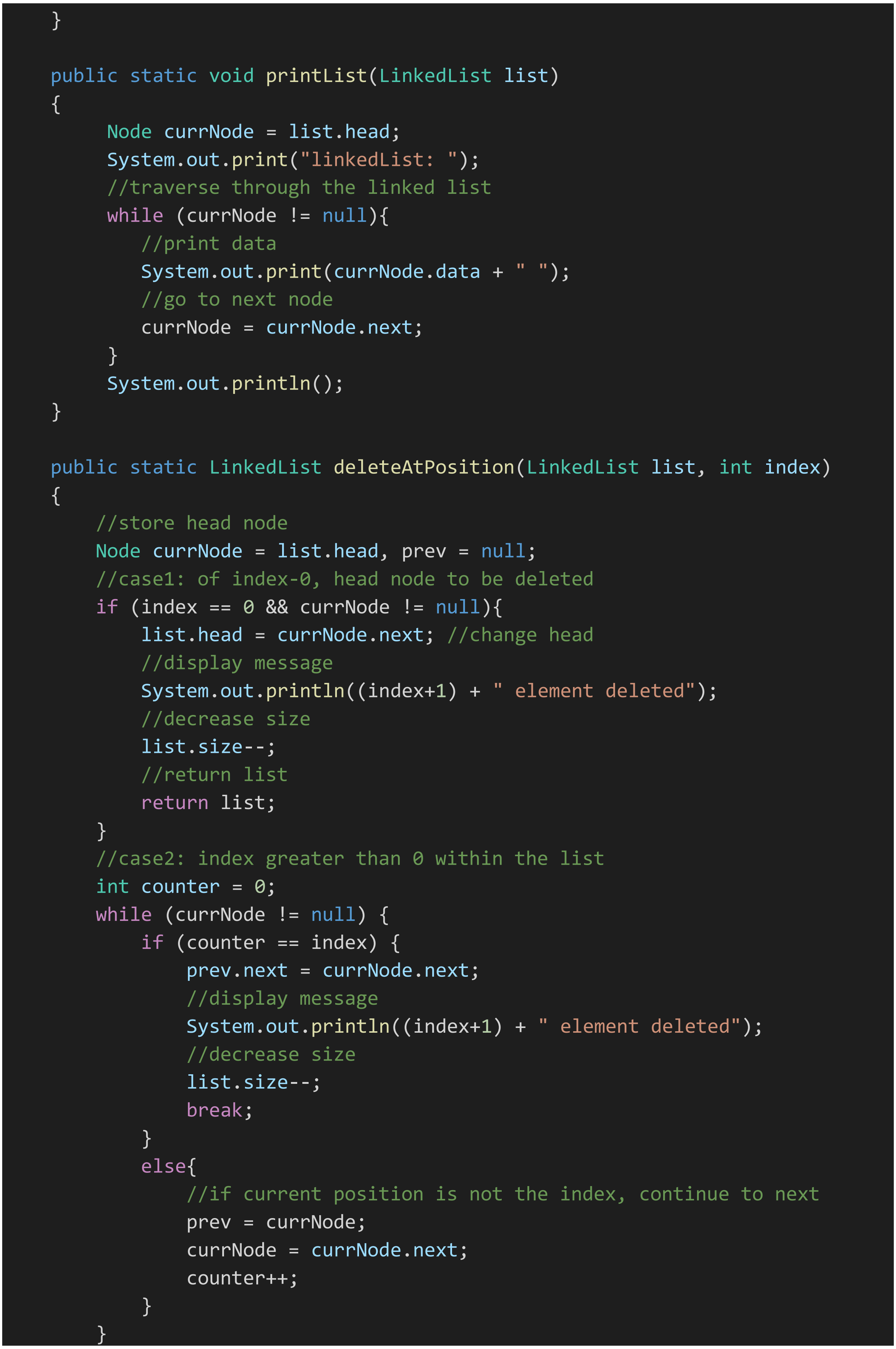


```
PS C:\Users\peace\OneDrive\Desktop\Introduction to
Enter a players number:
2
Could not find a player to match that number :(
PS C:\Users\peace\OneDrive\Desktop\Introduction to
Enter a players number:
5
Enzo Fernández
```

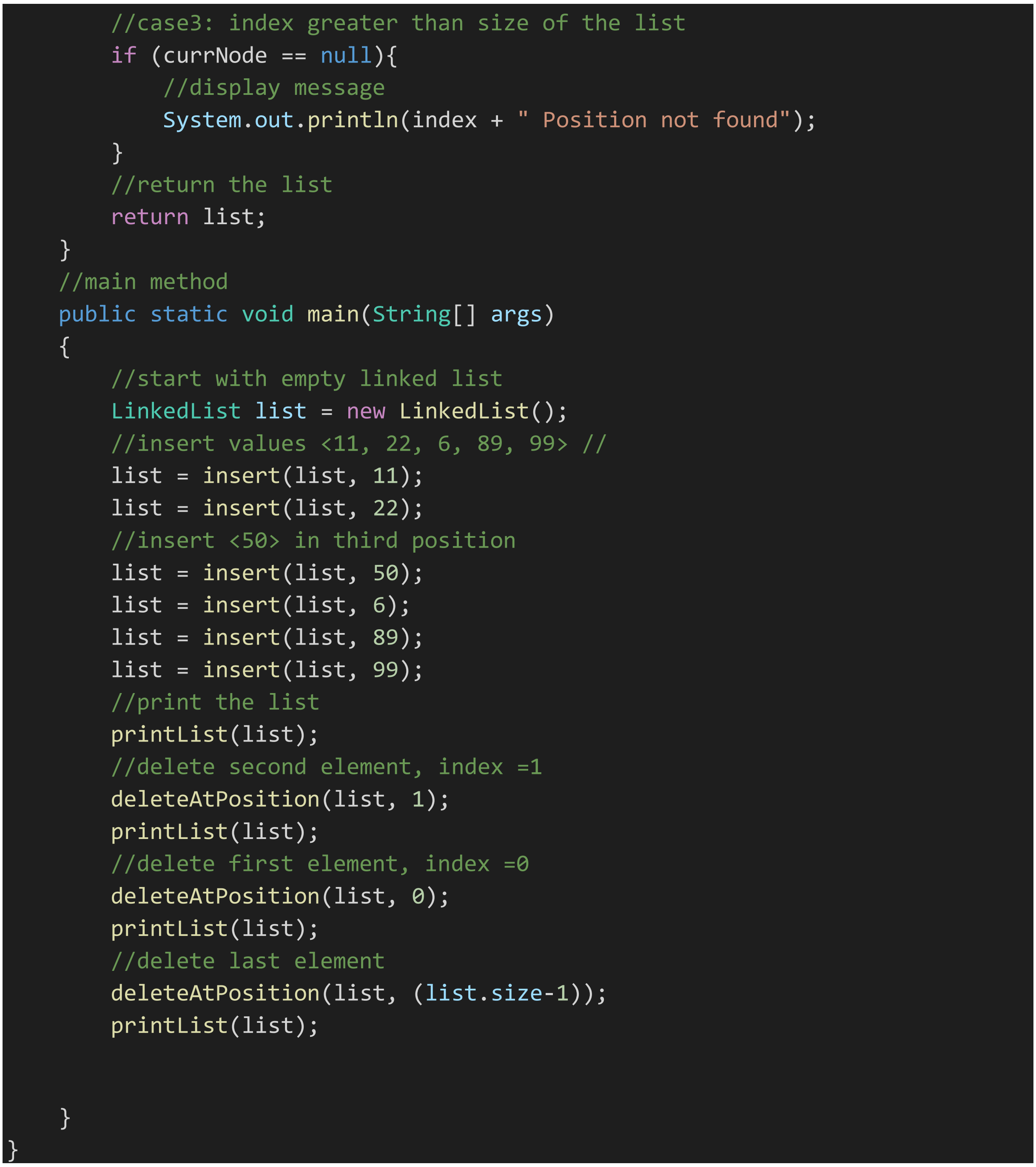
Program3:

Write a program in Java to implement a linked list containing 5 numbers <11, 22, 6, 89, 99> and then perform the following:
Then insert a number <50> in the third position of the linked list and print the new linked list <11, 22, 50, 6, 89, 99>
Delete the 2nd element of the linked list and print the remaining linked list <11, 50, 6, 89, 99>
Delete the 1st element of the linked list and print the remaining linked list <50, 6, 89, 99>
Delete the last element of the linked list and print the remaining linked list <50,6,89>
You are not allowed to use java.util.LinkedList

```
public class LinkedList {
    Node head; // head of the list
    int size; //size if the list

    /*Linked list node */
    static class Node {
        int data;
        Node next;
        //constructor to create a new node
        //next is default initialized as null
        Node(int d)
        {
            data = d;
            next = null;
        }
    }
    public static LinkedList insert(LinkedList list, int data)
    {
        //create a new node with given data
        Node new_node = new Node(data);
        new_node.next = null;
        //if empty make that new node the head
        if (list.head == null) {
            list.head = new_node;
        }
        else {
            //traverse till last node insert at the end
            Node last = list.head;
            while (last.next != null){
                last = last.next;
            }
            // insert the new_node at the last node
            last.next = new_node;
        }
        //increase the size of the list
        list.size++;
        //return the list by head
        return list;
```

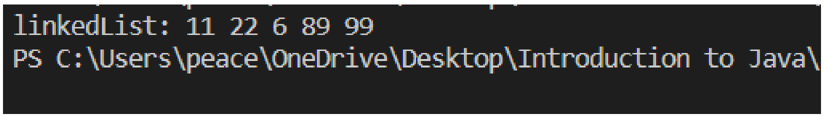


```
    }

    public static void printList(LinkedList list)
    {
         Node currNode = list.head;
         System.out.print("linkedList: ");
         //traverse through the linked list
         while (currNode != null){
            //print data
            System.out.print(currNode.data + " ");
            //go to next node
            currNode = currNode.next;
         }
         System.out.println();
    }

    public static LinkedList deleteAtPosition(LinkedList list, int index)
    {
        //store head node
        Node currNode = list.head, prev = null;
        //case1: of index-0, head node to be deleted
        if (index == 0 && currNode != null){
            list.head = currNode.next; //change head
            //display message
            System.out.println((index+1) + " element deleted");
            //decrease size
            list.size--;
            //return list
            return list;
        }
        //case2: index greater than 0 within the list
        int counter = 0;
        while (currNode != null) {
            if (counter == index) {
                prev.next = currNode.next;
                //display message
                System.out.println((index+1) + " element deleted");
                //decrease size
                list.size--;
                break;
            }
            else{
                //if current position is not the index, continue to next
                prev = currNode;
                currNode = currNode.next;
                counter++;
            }
        }
```

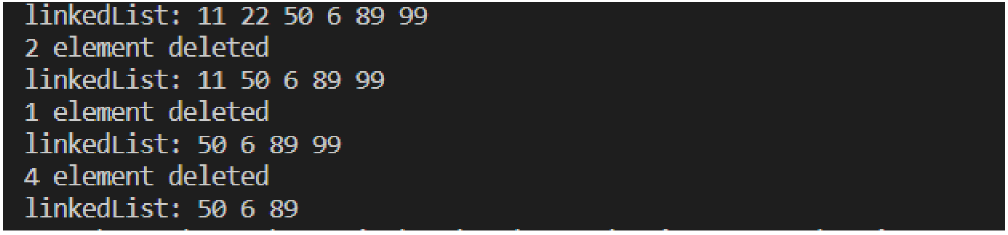


```
        //case3: index greater than size of the list
        if (currNode == null){
            //display message
            System.out.println(index + " Position not found");
        }
        //return the list
        return list;
    }
    //main method
    public static void main(String[] args)
    {
        //start with empty linked list
        LinkedList list = new LinkedList();
        //insert values <11, 22, 6, 89, 99> //
        list = insert(list, 11);
        list = insert(list, 22);
        //insert <50> in third position
        list = insert(list, 50);
        list = insert(list, 6);
        list = insert(list, 89);
        list = insert(list, 99);
        //print the list
        printList(list);
        //delete second element, index =1
        deleteAtPosition(list, 1);
        printList(list);
        //delete first element, index =0
        deleteAtPosition(list, 0);
        printList(list);
        //delete last element
        deleteAtPosition(list, (list.size-1));
        printList(list);


    }
}
```

Output:



```
linkedList: 11 22 6 89 99
PS C:\Users\peace\OneDrive\Desktop\Introduction to Java\
```



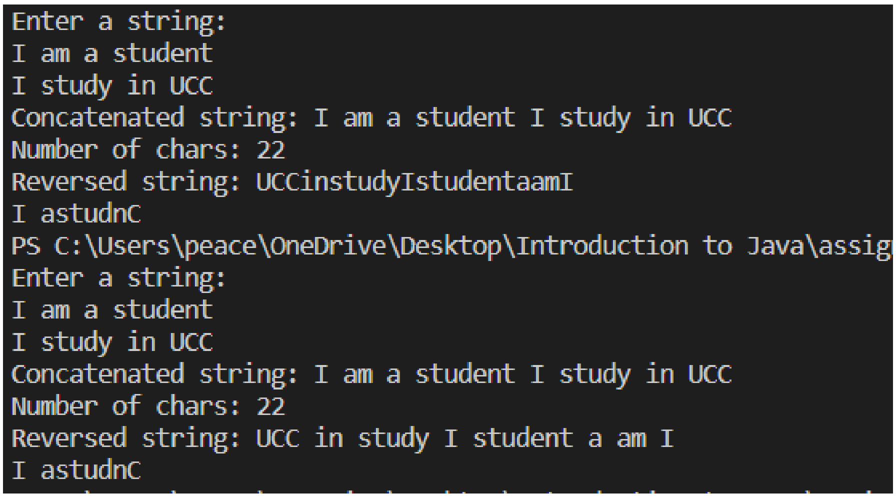
```
linkedList: 11 22 50 6 89 99
2 element deleted
linkedList: 11 50 6 89 99
1 element deleted
linkedList: 50 6 89 99
4 element deleted
linkedList: 50 6 89
```

Program4:

Write a program in Java that inputs two strings from the user and performs the following:
(i) Concatenates the strings
(ii) Prints the number of characters present in the strings (this will exclude the blank spaces)
(iii) Prints the concatenated string in the reverse order
(iv) Prints the characters which occurs twice

```
static void program4()
    {
        //inputs two strings from the user
        Scanner mystr = new Scanner(System.in);
        System.out.println("Enter a string:");
        String string1 = mystr.nextLine();
        String string2 = mystr.nextLine();
        // (i) Concatenate the strings
        String con_string = string1 +" "+ string2;
        System.out.println("Concatenated string: "+con_string);

        //(ii) print number of chars in the concatenated string
        int count = 0;//initialize count value to 0
        //count characters if it's not a space
        for (int i=0; i< con_string.length();  i++){
            if (con_string.charAt(i) != ' '){
                count++;
            }
        }
        System.out.println("Number of chars: " + count);

        //(iii) print concatenated string
        String rev_string = "";//initialize reverse string
        String[] parts = con_string.split(" ");
        for(int i=parts.length-1; i>=0; i--){
            rev_string += parts[i] + " ";
        }
        //rev_string =parts[1] +" "+ parts[0];
        System.out.println("Reversed string: "+ rev_string);

        //(iv)print chars that occur twice
        for (int i=0; i < con_string.length(); i++){
            //loop over each character in the string
            char current = con_string.charAt(i);
            //check if a char appears more than once and hasn't already been
printed
            if (con_string.indexOf(current, i+1) != -1 &&
con_string.indexOf(current) == i){
                System.out.print(current);
            }
        }
    }
```

Output:



```
Enter a string:
I am a student
I study in UCC
Concatenated string: I am a student I study in UCC
Number of chars: 22
Reversed string: UCCinstudyIstudentaamI
I astudnC
PS C:\Users\peace\OneDrive\Desktop\Introduction to Java\assig
Enter a string:
I am a student
I study in UCC
Concatenated string: I am a student I study in UCC
Number of chars: 22
Reversed string: UCC in study I student a am I
I astudnC
```