CYCLE 4

1. **Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.**

Code:

package_graphics:

```
package package_graphics;


interface interface_package{
  public float recArea(int l, int h);
  public float cirArea(int r);
  public float squArea(int a);
  public float triArea(int l, int h);
}


public class package_graphics implements interface_package{


   public float recArea(int l, int h){
      return l*h;
   }


   public float cirArea(int r){
      return r*r*(float)3.14;
   }
   public float squArea(int a){
      return a*a;
   }

   public float triArea(int l, int h){
      return l*h*(float)(.5);
   }
}
```

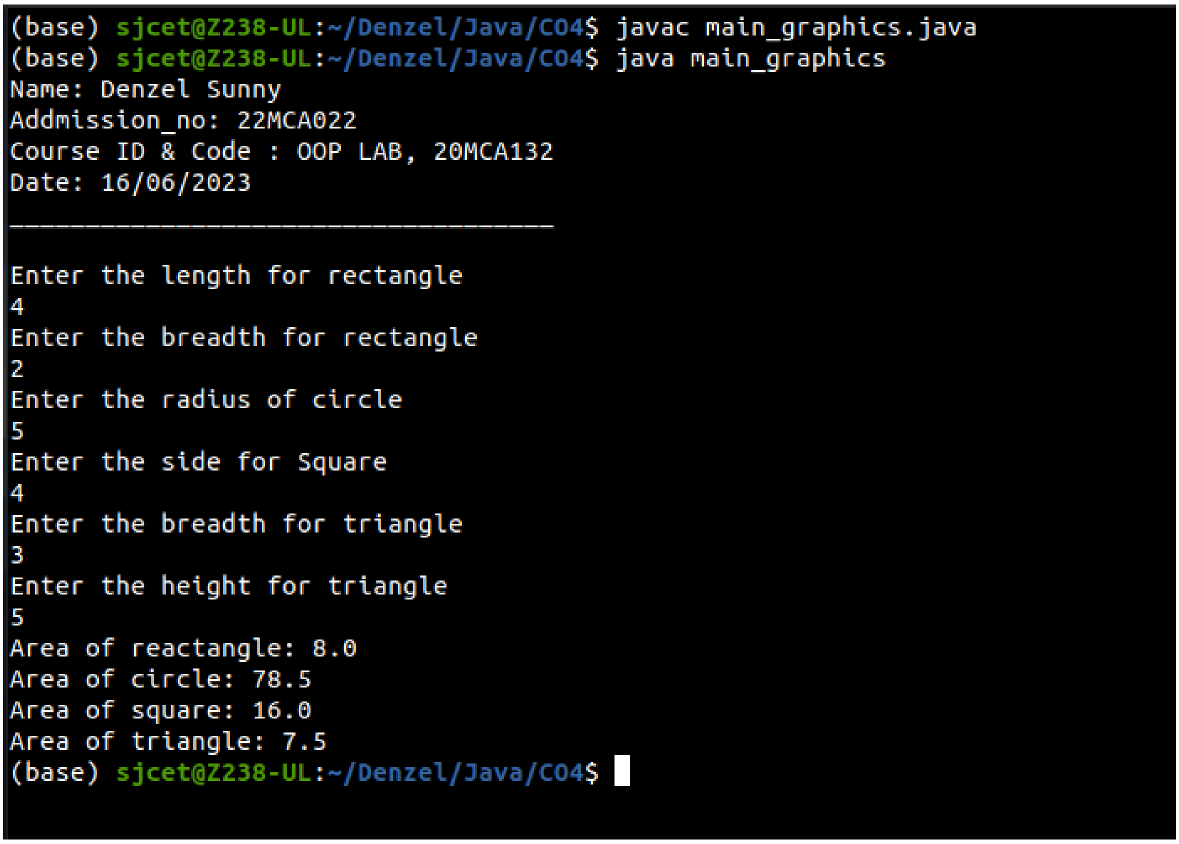
Main_graphics:

```
import package_graphics.*;
import java.util.*;

public class main_graphics{
  public static void main(String[] args){
  System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 16/06/2023");
System.out.println("______________________________________\n");

    package_graphics testObj = new package_graphics();

    int l,h,r,a,c,d;
```

```
Scanner s = new Scanner(System.in);


System.out.println("Enter the length for rectangle");
l = s.nextInt();


System.out.println("Enter the breadth for rectangle");
h = s.nextInt();


System.out.println("Enter the radius of circle");
r = s.nextInt();


System.out.println("Enter the side for Square");
a = s.nextInt();


System.out.println("Enter the breadth for triangle");
c = s.nextInt();


System.out.println("Enter the height for triangle");
d = s.nextInt();


System.out.println("Area of reactangle: "+testObj.recArea(l,h));
System.out.println("Area of circle: "+testObj.cirArea(r));
System.out.println("Area of square: "+testObj.squArea(a));
System.out.println("Area of triangle: "+testObj.triArea(c,d));
```

```
 }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ javac main_graphics.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ java main_graphics
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 16/06/2023
____________________________________

Enter the length for rectangle
4
Enter the breadth for rectangle
2
Enter the radius of circle
5
Enter the side for Square
4
Enter the breadth for triangle
3
Enter the height for triangle
5
Area of reactangle: 8.0
Area of circle: 78.5
Area of square: 16.0
Area of triangle: 7.5
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$
```

2. **Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers**

Code:

Arithmetic:

Addition:

```
package arithmetic;

public class Addition implements Arithmetic {
    public double calculate(double a, double b) {
        return a + b;
    }
}
```

Arithmetic:

```
package arithmetic;

public interface Arithmetic {
    double calculate(double a, double b);
}
```

Division:

```
package arithmetic;

public class Division implements Arithmetic {
    public double calculate(double a, double b) {
        if (b != 0) {
            return a / b;
        } else {
            throw new ArithmeticException("Cannot divide by zero");
        }
    }
}
```

Multiplication:

```
package arithmetic;

public class Multiplication implements Arithmetic {
    public double calculate(double a, double b) {
        return a * b;
    }
}
```

Subtraction:

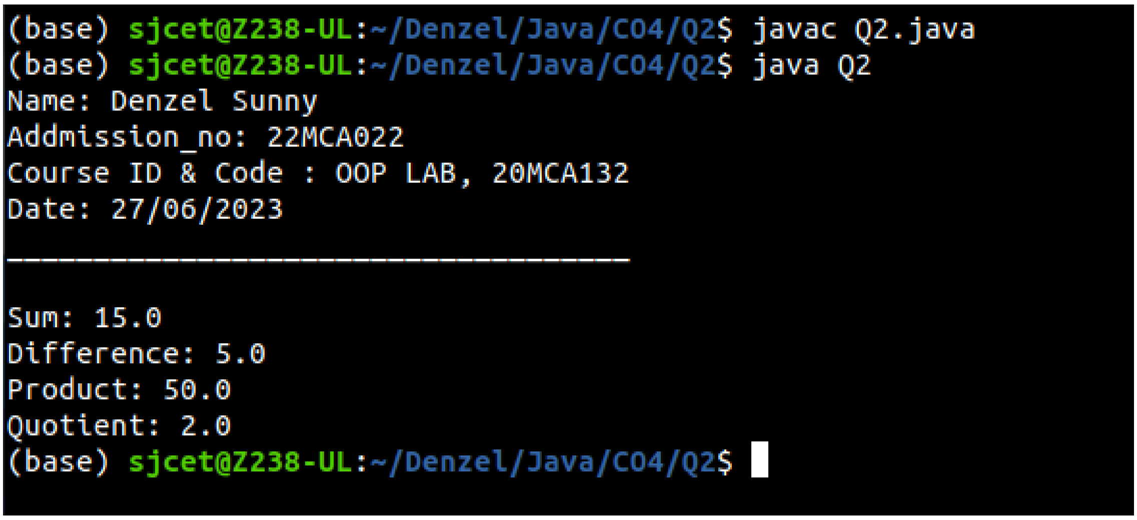
```
package arithmetic;

public class Subtraction implements Arithmetic {
    public double calculate(double a, double b) {
        return a - b;
    }
}
```

Main:

```
import arithmetic.*;

public class Q2 {
    public static void main(String[] args) {

    System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 27/06/2023");
System.out.println("___________________________________\n");
```

```
        double num1 = 10;
        double num2 = 5;

        Arithmetic addition = new Addition();
        double sum = addition.calculate(num1, num2);
        System.out.println("Sum: " + sum);

        Arithmetic subtraction = new Subtraction();
        double difference = subtraction.calculate(num1, num2);
        System.out.println("Difference: " + difference);

        Arithmetic multiplication = new Multiplication();
        double product = multiplication.calculate(num1, num2);
        System.out.println("Product: " + product);

        Arithmetic division = new Division();
        double quotient = division.calculate(num1, num2);
        System.out.println("Quotient: " + quotient);
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q2$ javac Q2.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q2$ java Q2
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
------------------------------------

Sum: 15.0
Difference: 5.0
Product: 50.0
Quotient: 2.0
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q2$
```

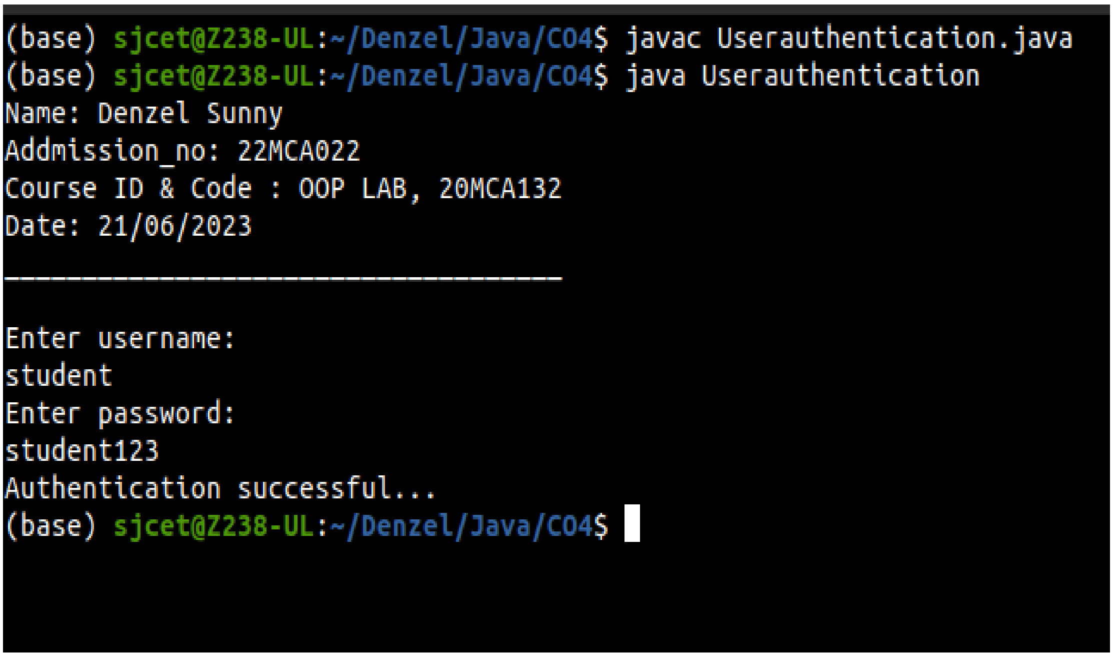
3. **Write a user defined exception class to authenticate the user name and password.**

Code:

```
import java.util.Scanner;


class authException extends Exception{


    public authException(String s){

       super(s);

    }

}


public class Userauthentication{


  public static void main(String[] args){

           System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 21/06/2023");

System.out.println("_____________________________________\n");
```

```
    String username = "student";

    String passcode = "student123";

    String user_name,password;

    Scanner sc = new Scanner(System.in);

    try

    {

      System.out.println("Enter username: ");

      user_name = sc.nextLine();


      System.out.println("Enter password: ");

      password = sc.nextLine();


      if(username.equals(user_name) && passcode.equals(password)){

        System.out.println("Authentication successful...");

      }

      else

         throw new authException("Invalid user credentials");

    }

    catch(authException e)

    {

      System.out.println("Exception caught "+e);

    }

  }

}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ javac Userauthentication.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ java Userauthentication
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 21/06/2023
____________________________________

Enter username:
student
Enter password:
student123
Authentication successful...
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$
```

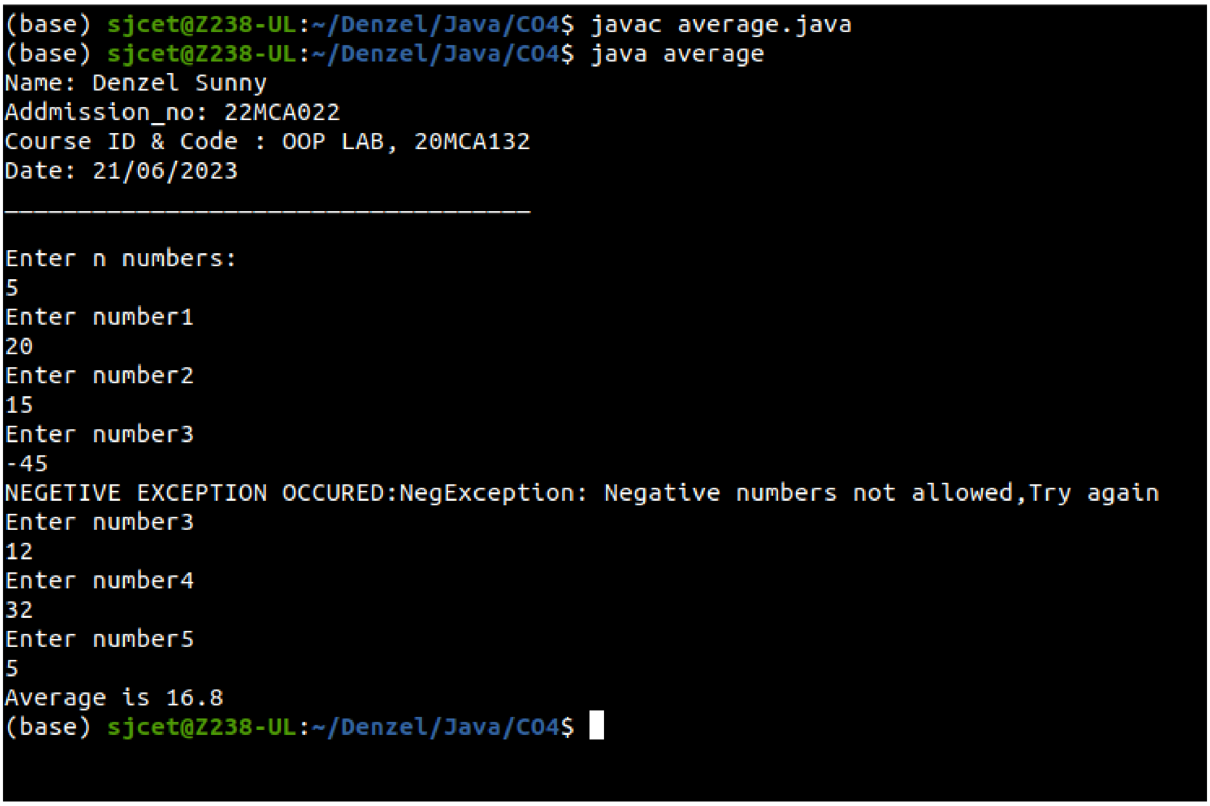
4. **Find the average of N positive integers, raising a user defined exception for each negative input**.

Code:

```
import java.util.Scanner;

class NegException extends Exception{

   public NegException(String s){

   super(s);

  }

}

public class average {

  public static void main(String[] args){


      System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 21/06/2023");

System.out.println("_____________________________________\n");



      int i;

      double sum=0,avg=0;

      Scanner sc=new Scanner(System.in);

      System.out.println("Enter n numbers:");
```

```
        int n=sc.nextInt();
        for(i=1;i<=n;i++){
          try{
            System.out.println("Enter number"+i);
            int a=sc.nextInt();
            if(a<0){
              i--;
              throw new NegException("Negative numbers not allowed,Try again");
            }
            else{
             sum=sum+a;
            }
           }
          catch(NegException e){
          System.out.println("NEGETIVE EXCEPTION OCCURED:"+e);
        }
      }
      avg=sum/n;
      System.out.println("Average is "+avg);
      sc.close();
    }
  }
```

Output:



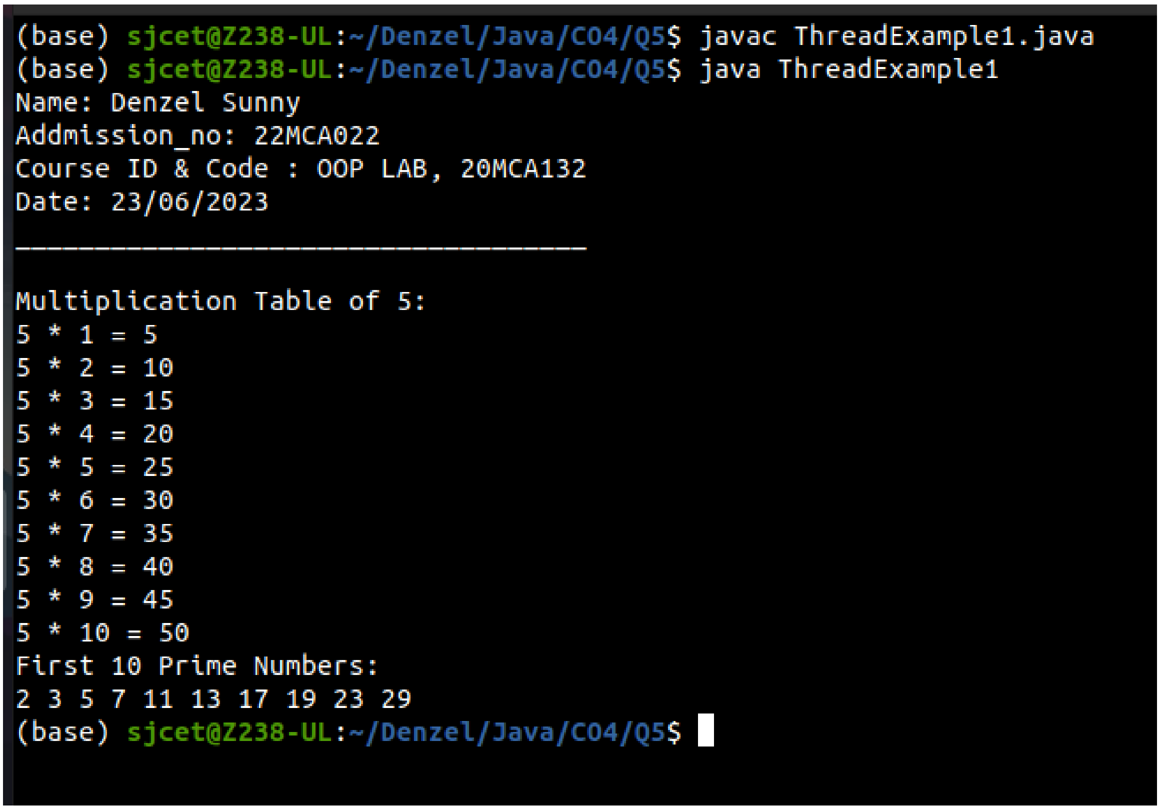
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ javac average.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ java average
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 21/06/2023
____________________________________

Enter n numbers:
5
Enter number1
20
Enter number2
15
Enter number3
-45
NEGETIVE EXCEPTION OCCURED:NegException: Negative numbers not allowed,Try again
Enter number3
12
Enter number4
32
Enter number5
5
Average is 16.8
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$
```

5. Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

Code:

```
class MultiplicationTable implements Runnable {
    @Override
    public void run() {
        System.out.println("Multiplication Table of 5:");
        for (int i = 1; i <= 10; i++) {
            System.out.println("5 * " + i + " = " + (5 * i));
        }
    }
}

class PrimeNumbers implements Runnable {
    @Override
    public void run() {
        System.out.println("First 10 Prime Numbers:");
        int count = 0;
        int num = 2;
        while (count < 10) {
            if (isPrime(num)) {
                System.out.print(num + " ");
                count++;
            }
            num++;
        }
        System.out.println();
    }

    private boolean isPrime(int num) {
        if (num < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}

public class ThreadExample1 {
    public static void main(String[] args) {

          System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 23/06/2023");
System.out.println("____________________________________\n");



        MultiplicationTable multiplicationTable = new MultiplicationTable();
        PrimeNumbers primeNumbers = new PrimeNumbers();

        Thread thread1 = new Thread(multiplicationTable);
        Thread thread2 = new Thread(primeNumbers);

        thread1.start();
        try {
            thread1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        thread2.start();
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q5$ javac ThreadExample1.java
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q5$ java ThreadExample1
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 23/06/2023
____________________________________

Multiplication Table of 5:
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
First 10 Prime Numbers:
2 3 5 7 11 13 17 19 23 29
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q5$
```

6. Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

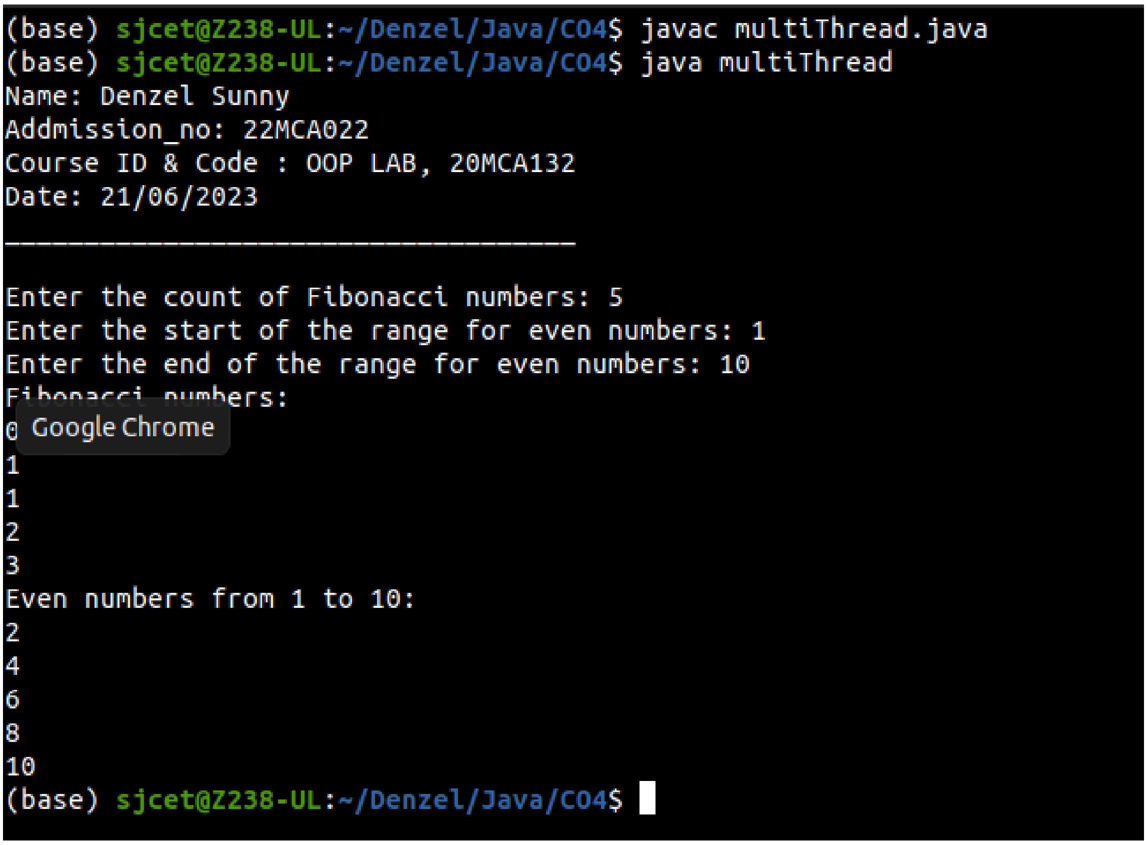
Code:

```
import java.util.Scanner;


class Fibonacci implements Runnable {
    private int count;


    public Fibonacci(int count) {
        this.count = count;
    }


    @Override
    public void run() {
        System.out.println("Fibonacci numbers:");
        int num1 = 0;
        int num2 = 1;
        System.out.println(num1);
        System.out.println(num2);


        for (int i = 2; i < count; i++) {
            int fib = num1 + num2;
            System.out.println(fib);
            num1 = num2;
```

```
                num2 = fib;
            }
        }
    }

    class EvenNumber implements Runnable {
        private int start;
        private int end;

        public EvenNumber(int start, int end) {
            this.start = start;
            this.end = end;
        }

        @Override
        public void run() {
            System.out.println("Even numbers from " + start + " to " + end + ":");
            for (int i = start; i <= end; i++) {
                if (i % 2 == 0) {
                    System.out.println(i);
                }
            }
        }
    }
```

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

        System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 21/06/2023");
System.out.println("___________________________________\n");



        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the count of Fibonacci numbers: ");
        int fibonacciCount = scanner.nextInt();

        System.out.print("Enter the start of the range for even numbers: ");
        int start = scanner.nextInt();

        System.out.print("Enter the end of the range for even numbers: ");
        int end = scanner.nextInt();

        scanner.close();
        Thread fibonacciThread = new Thread(new Fibonacci(fibonacciCount));
        Thread evenNumberThread = new Thread(new EvenNumber(start,
end));

        fibonacciThread.start();
        evenNumberThread.start();
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ javac multiThread.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$ java multiThread
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 21/06/2023
------------------------------------

Enter the count of Fibonacci numbers: 5
Enter the start of the range for even numbers: 1
Enter the end of the range for even numbers: 10
Fibonacci numbers:
0
1
1
2
3
Even numbers from 1 to 10:
2
4
6
8
10
(base) sjcet@Z238-UL:~/Denzel/Java/CO4$
```

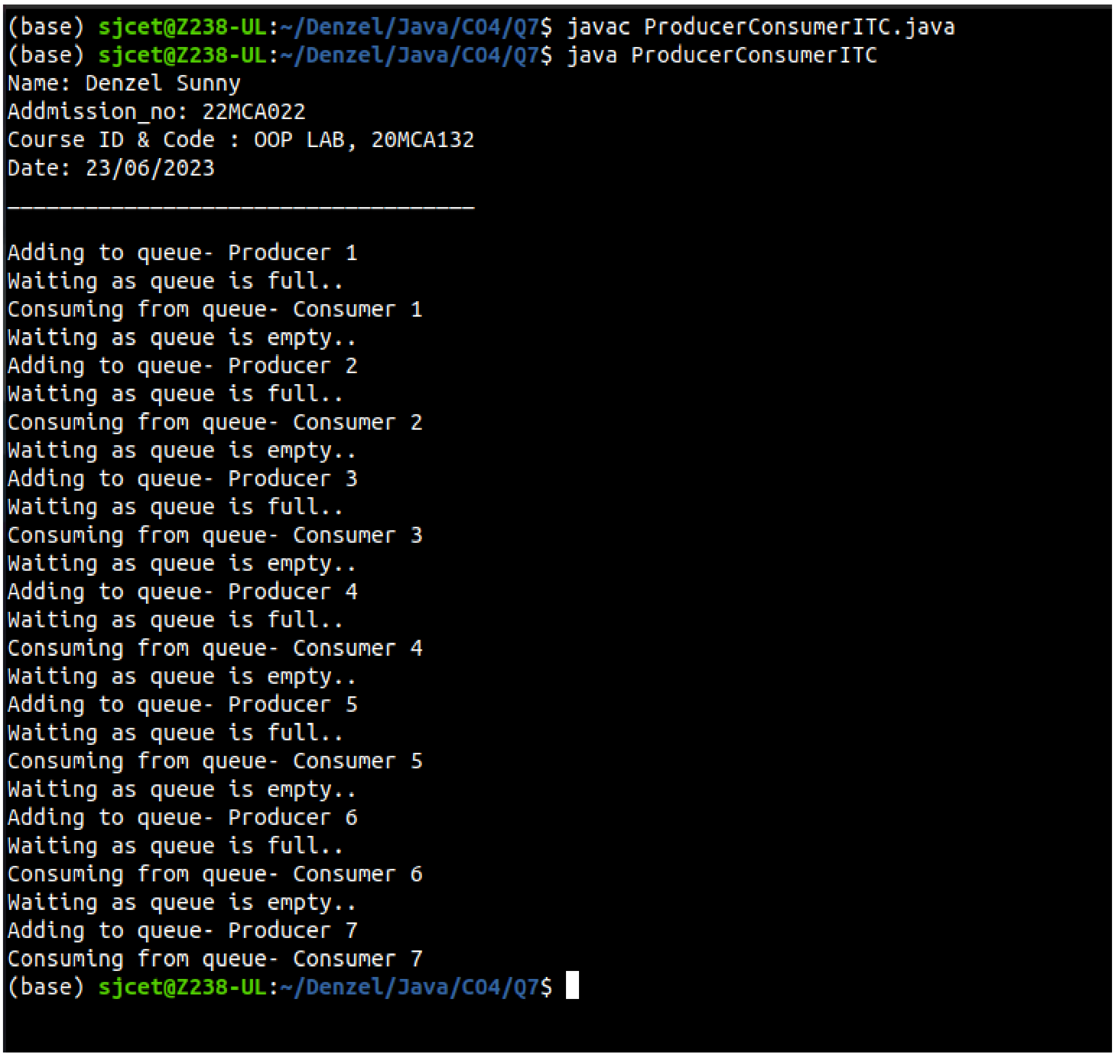
7. Producer/Consumer using ITC

Code:

```
import java.util.LinkedList;

class Producer implements Runnable{
LinkedList<Integer> list;
Producer(LinkedList<Integer> list){
 this.list = list;
}
@Override
public void run() {
 for(int i = 1; i <=7; i++){
   synchronized(list) {

     while(list.size() >= 1){
       System.out.println("Waiting as queue is full..");
       try {
         list.wait();
       } catch (InterruptedException e) {
         e.printStackTrace();
       }
     }
     System.out.println("Adding to queue- " +
Thread.currentThread().getName() + " " + i);
     list.add(i);
     list.notify();
   }
 }
}
}

class Consumer implements Runnable{
LinkedList<Integer> list;
```

```
Consumer(LinkedList<Integer> list){
 this.list = list;
}
@Override
public void run() {
 for(int i = 1; i <= 7; i++){
   synchronized(list) {

     while(list.size() < 1){
       System.out.println("Waiting as queue is empty..");
       try {
         list.wait();
       } catch (InterruptedException e) {
         e.printStackTrace();
       }
     }

     System.out.println("Consuming from queue- " +
Thread.currentThread().getName() + " " + list.remove());
     list.notify();
   }
 }
}
}

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

  System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 23/06/2023");
System.out.println("___________________________________\n");
LinkedList<Integer> list = new LinkedList<Integer>();
 Thread t1 = new Thread(new Producer(list), "Producer");
 Thread t2 = new Thread(new Consumer(list), "Consumer");
 t1.start();
 t2.start();
}
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q7$ javac ProducerConsumerITC.java
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q7$ java ProducerConsumerITC
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 23/06/2023
____________________________________

Adding to queue- Producer 1
Waiting as queue is full..
Consuming from queue- Consumer 1
Waiting as queue is empty..
Adding to queue- Producer 2
Waiting as queue is full..
Consuming from queue- Consumer 2
Waiting as queue is empty..
Adding to queue- Producer 3
Waiting as queue is full..
Consuming from queue- Consumer 3
Waiting as queue is empty..
Adding to queue- Producer 4
Waiting as queue is full..
Consuming from queue- Consumer 4
Waiting as queue is empty..
Adding to queue- Producer 5
Waiting as queue is full..
Consuming from queue- Consumer 5
Waiting as queue is empty..
Adding to queue- Producer 6
Waiting as queue is full..
Consuming from queue- Consumer 6
Waiting as queue is empty..
Adding to queue- Producer 7
Consuming from queue- Consumer 7
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q7$
```

8. Program to create a generic stack and do the Push and Pop operations.

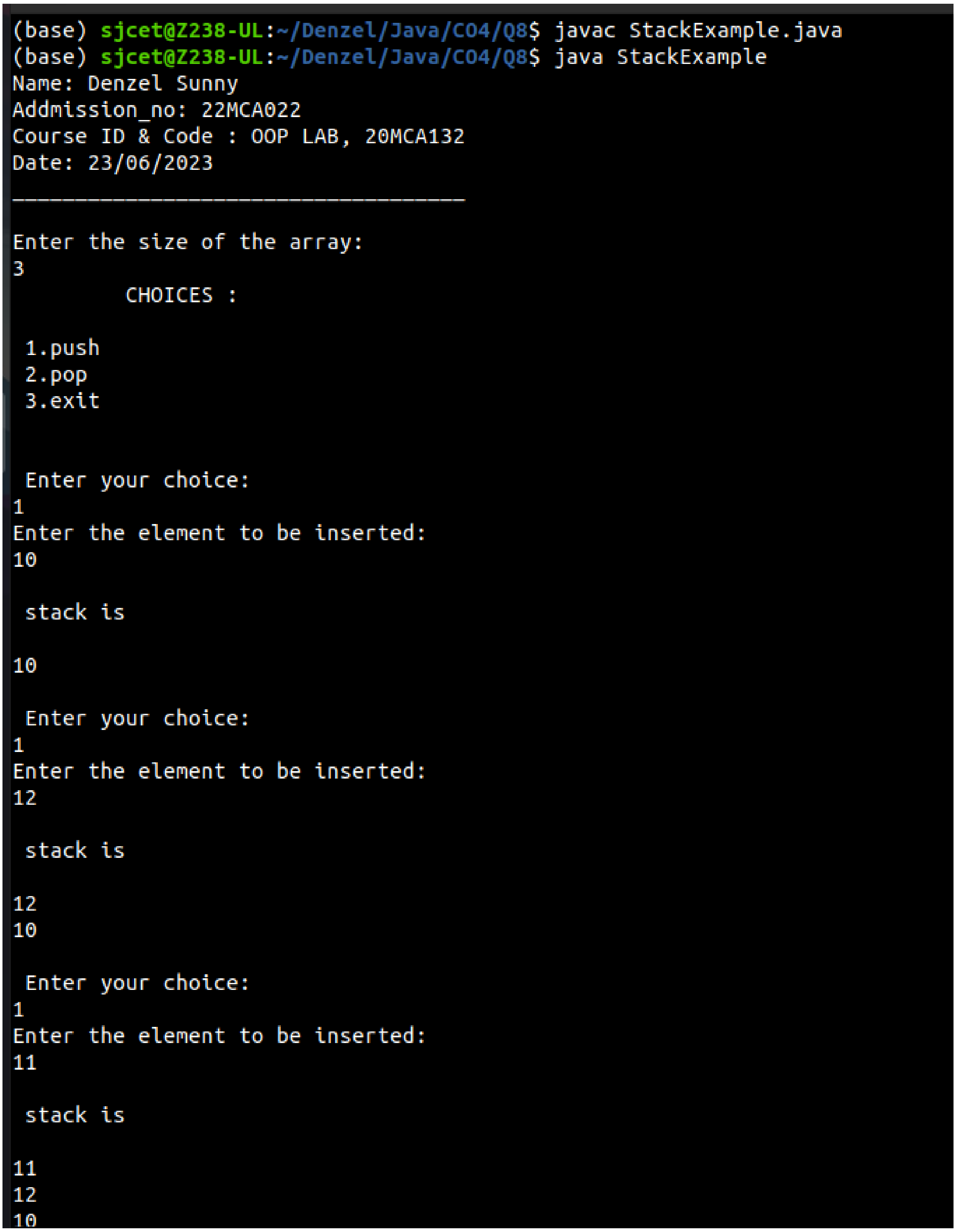
Code:

```
import java.util.Scanner;

public class StackExample {

        int top=-1,ch,item,i;

        int a[] = new int[10];

   Scanner sc = new Scanner(System.in);

        public static void main(String[] args) {


        System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("____________________________________\n");




                StackExample obj = new StackExample ();

                obj.stack();

                }


        public void stack(){

          System.out.println("Enter the size of the array:");

                        int N=sc.nextInt();
```

```
System.out.println("\t CHOICES : ");
           System.out.println("\n 1.push \n 2.pop \n 3.exit \n");
        while(ch<3) {
           System.out.println("\n Enter your choice:");
           ch=sc.nextInt();
        switch(ch){
        case 1:
                System.out.println("Enter the element to be inserted:");
                item=sc.nextInt();
                if(top==N-1) {
                        System.out.println("stack overflow!");
                }
                else {
                        top++;
                        a[top]=item;
                }
                break;
        case 2:
                if(top==-1) {
                        System.out.println("stack is empty");
                }
                else {
                        item=a[top];
```

```
                                top--;
                                System.out.println("deleted element is:" +item);
                           }
                        break;
                      case 3 : break;
                      default : System.out.println("\n Invalid choice");
                      }
                   if(top < 0){
                   System.out.println("\n stack is empty");
                     }
                   else{
                   System.out.println("\n stack is \n");
                   for(i=top;i>=0;i--){
                   System.out.println(a[i]);
                  }
                    }
                   }
                 }
            }
```

Output:



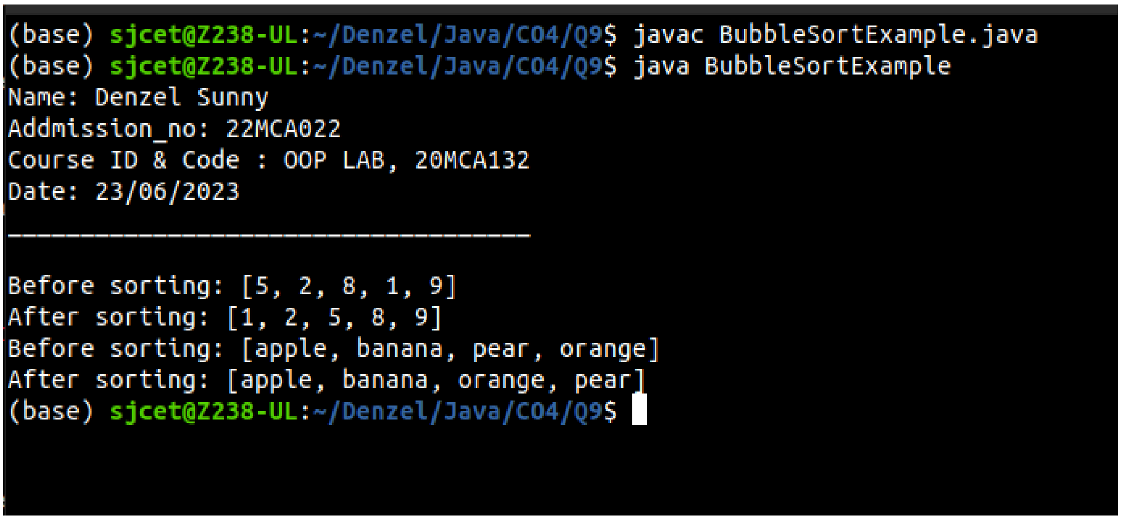
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q8$ javac StackExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q8$ java StackExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 23/06/2023
____________________________________

Enter the size of the array:
3
         CHOICES :

 1.push
 2.pop
 3.exit


 Enter your choice:
1
Enter the element to be inserted:
10

 stack is

10

 Enter your choice:
1
Enter the element to be inserted:
12

 stack is

12
10

 Enter your choice:
1
Enter the element to be inserted:
11

 stack is

11
12
10
```

9. Using generic method perform Bubble sort.

Code:

```
import java.util.Arrays;

class BubbleSort {
    public static <T extends Comparable<T>> void bubbleSort(T[] array) {
        int n = array.length;
        boolean swapped;

        for (int i = 0; i < n - 1; i++) {
            swapped = false;

            for (int j = 0; j < n - i - 1; j++) {
                if (array[j].compareTo(array[j + 1]) > 0) {
                    T temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                    swapped = true;
                }
            }

            if (!swapped) {
                break;
            }
        }
    }
}

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

    System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 23/06/2023");
System.out.println("___________________________________\n");

        Integer[] intArray = {5, 2, 8, 1, 9};
        System.out.println("Before sorting: " + Arrays.toString(intArray));
        BubbleSort.bubbleSort(intArray);
        System.out.println("After sorting: " + Arrays.toString(intArray));

        String[] stringArray = {"apple", "banana", "pear", "orange"};
        System.out.println("Before sorting: " + Arrays.toString(stringArray));
        BubbleSort.bubbleSort(stringArray);
        System.out.println("After sorting: " + Arrays.toString(stringArray));
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q9$ javac BubbleSortExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q9$ java BubbleSortExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 23/06/2023
___________________________________

Before sorting: [5, 2, 8, 1, 9]
After sorting: [1, 2, 5, 8, 9]
Before sorting: [apple, banana, pear, orange]
After sorting: [apple, banana, orange, pear]
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q9$
```

10. Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

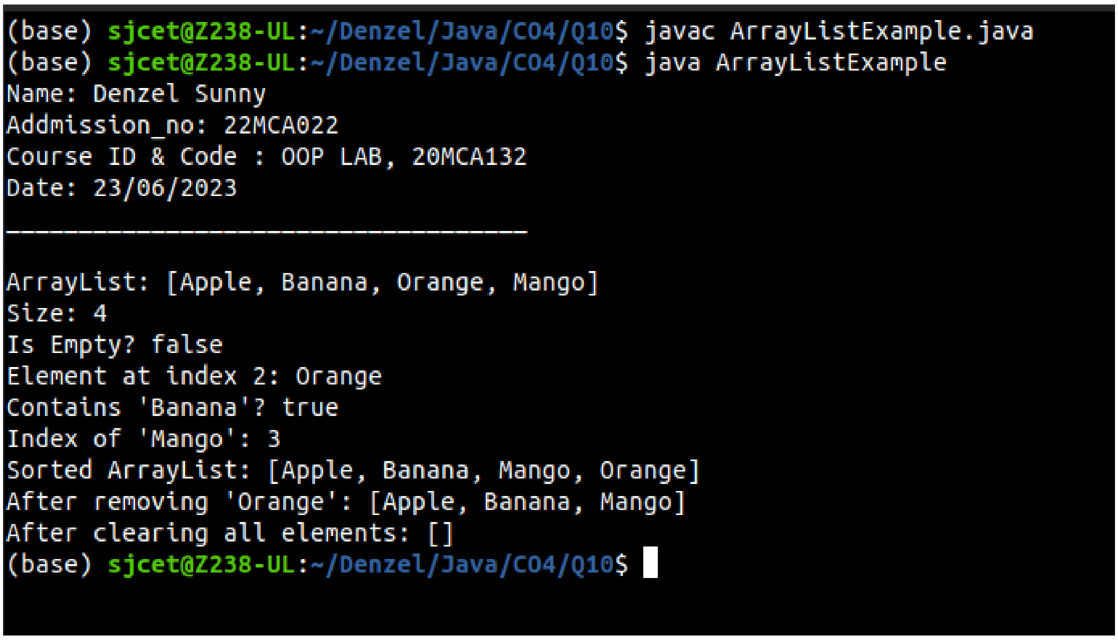
Code:

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.List;


public class ArrayListExample {

   public static void main(String[] args) {


   System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("___________________________________\n");



      List<String> stringList = new ArrayList<>();



      stringList.add("Apple");

      stringList.add("Banana");

      stringList.add("Orange");

      stringList.add("Mango");
```

```
System.out.println("ArrayList: " + stringList);

System.out.println("Size: " + stringList.size());

System.out.println("Is Empty? " + stringList.isEmpty());

String element = stringList.get(2);
System.out.println("Element at index 2: " + element);

boolean contains = stringList.contains("Banana");
System.out.println("Contains 'Banana'? " + contains);

int index = stringList.indexOf("Mango");
System.out.println("Index of 'Mango': " + index);

Collections.sort(stringList);
```

```
        System.out.println("Sorted ArrayList: " + stringList);



        stringList.remove("Orange");

        System.out.println("After removing 'Orange': " + stringList);



        stringList.clear();

        System.out.println("After clearing all elements: " + stringList);

    }

}
```

Output:



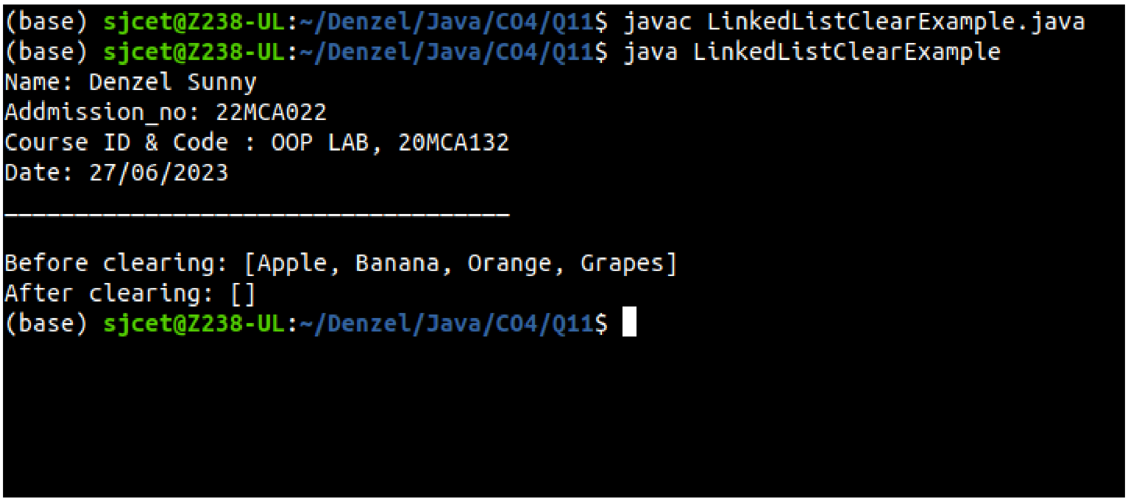
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q10$ javac ArrayListExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q10$ java ArrayListExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 23/06/2023
____________________________________

ArrayList: [Apple, Banana, Orange, Mango]
Size: 4
Is Empty? false
Element at index 2: Orange
Contains 'Banana'? true
Index of 'Mango': 3
Sorted ArrayList: [Apple, Banana, Mango, Orange]
After removing 'Orange': [Apple, Banana, Mango]
After clearing all elements: []
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q10$
```

11. Program to remove all the elements from a linked list

Code:

```
import java.util.LinkedList;

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

   System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 27/06/2023");
System.out.println("___________________________________\n");



      LinkedList<String> linkedList = new LinkedList<>();


      linkedList.add("Apple");
      linkedList.add("Banana");
      linkedList.add("Orange");
      linkedList.add("Grapes");

      System.out.println("Before clearing: " + linkedList);


      linkedList.clear();

      System.out.println("After clearing: " + linkedList);
   }
}
```

Output:



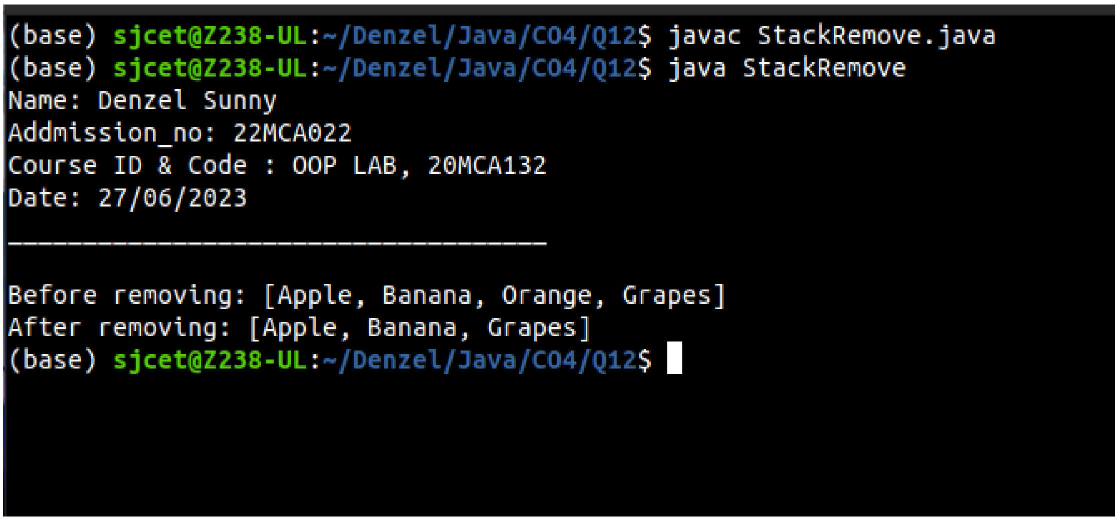
```
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q11$ javac LinkedListClearExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q11$ java LinkedListClearExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Before clearing: [Apple, Banana, Orange, Grapes]
After clearing: []
(base) sjcet@Z238-UL:~/Denzel/Java/C04/Q11$
```

12. Program to remove an object from the Stack when the position is passed as parameter

Code:

```
import java.util.Stack;


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


    System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 27/06/2023");
System.out.println("____________________________________\n");




        Stack<String> stack = new Stack<>();



        stack.push("Apple");
        stack.push("Banana");
        stack.push("Orange");
        stack.push("Grapes");
        System.out.println("Before removing: " + stack);


        int positionToRemove = 2; // Position starts from 1
        removeByPosition(stack, positionToRemove);

        System.out.println("After removing: " + stack);
    }

    public static void removeByPosition(Stack<String> stack, int position) {
        if (position < 1 || position > stack.size()) {
            System.out.println("Invalid position");
            return;
        }

        Stack<String> tempStack = new Stack<>();


        for (int i = 1; i < position; i++) {
            tempStack.push(stack.pop());
        }


        stack.pop();
        while (!tempStack.isEmpty()) {
            stack.push(tempStack.pop());
        }
    }
}
```

Output:



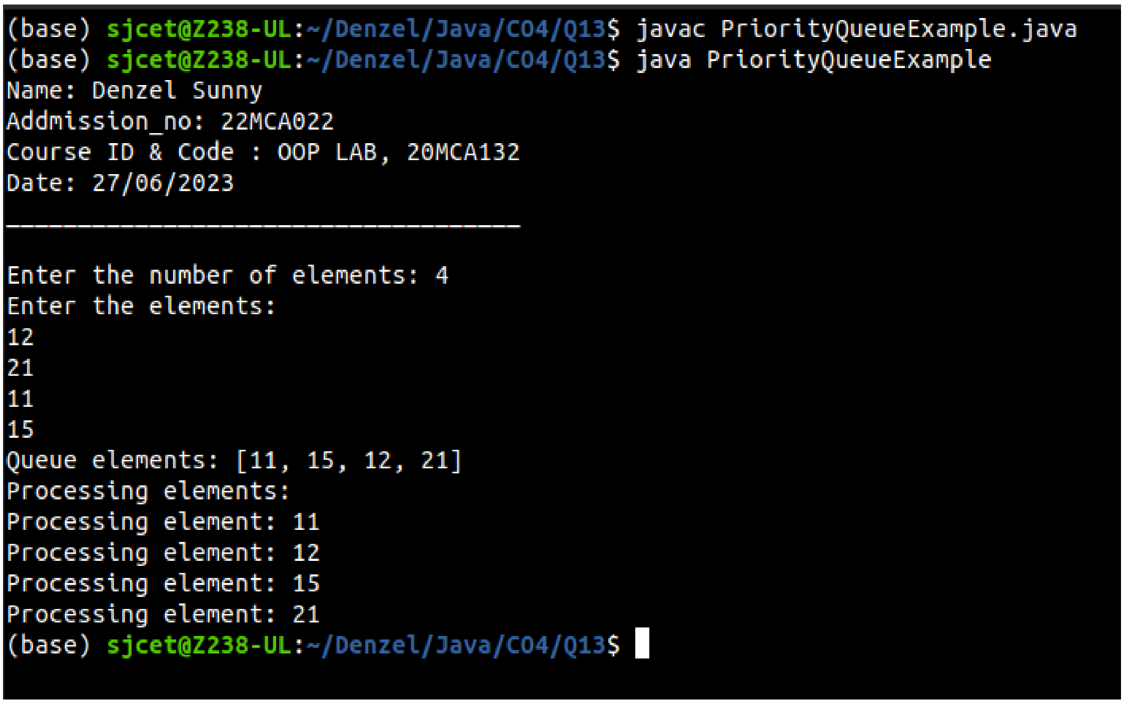
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q12$ javac StackRemove.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q12$ java StackRemove
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
------------------------------------

Before removing: [Apple, Banana, Orange, Grapes]
After removing: [Apple, Banana, Grapes]
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q12$
```

13. Program to demonstrate the creation of queue object using the PriorityQueue class

Code:

```
import java.util.PriorityQueue;

import java.util.Queue;

import java.util.Scanner;


public class PriorityQueueExample {

    public static void main(String[] args) {


    System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("_____________________________________\n");



        Queue<Integer> queue = new PriorityQueue<>();



        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of elements: ");

        int numElements = scanner.nextInt();
        System.out.println("Enter the elements:");

        for (int i = 0; i < numElements; i++) {
            int element = scanner.nextInt();
            queue.offer(element);
        }

        System.out.println("Queue elements: " + queue);

        System.out.println("Processing elements:");

        while (!queue.isEmpty()) {
            int element = queue.poll();
            System.out.println("Processing element: " + element);
        }

        scanner.close();
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q13$ javac PriorityQueueExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q13$ java PriorityQueueExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Enter the number of elements: 4
Enter the elements:
12
21
11
15
Queue elements: [11, 15, 12, 21]
Processing elements:
Processing element: 11
Processing element: 12
Processing element: 15
Processing element: 21
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q13$
```

14. Program to demonstrate the addition and deletion of elements in deque

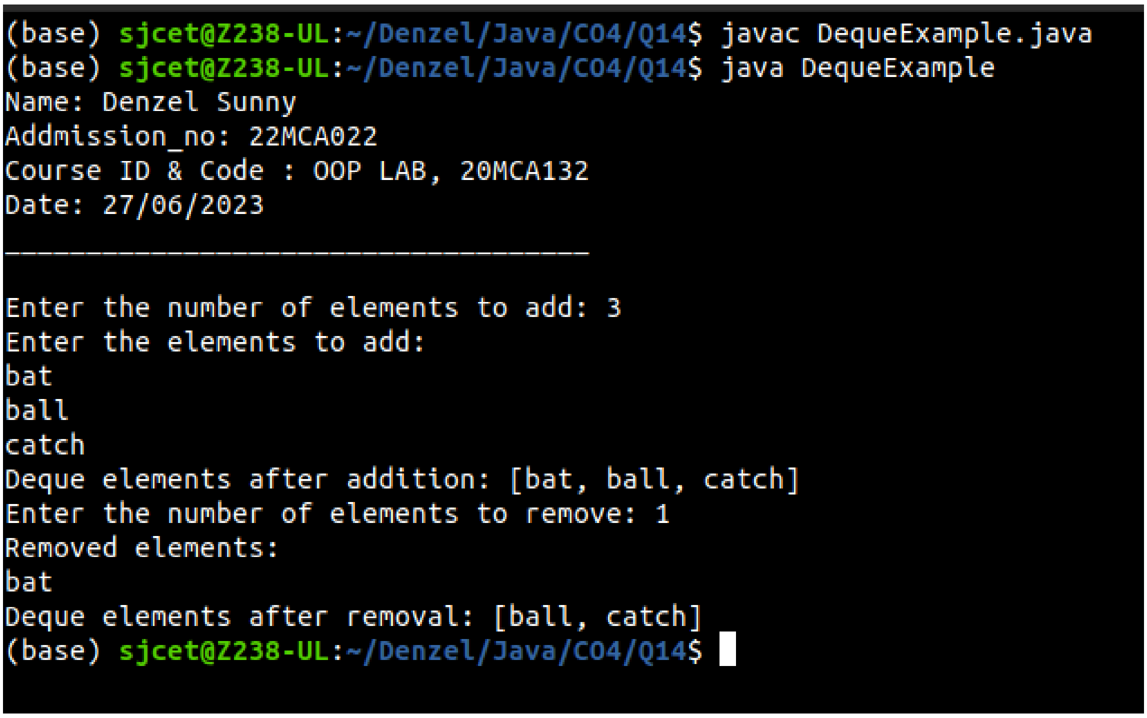
Code:

```
import java.util.Deque;

import java.util.LinkedList;

import java.util.Scanner;


public class DequeExample {

    public static void main(String[] args) {


    System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("_________________________________________\n");




        Deque<String> deque = new LinkedList<>();


        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of elements to add: ");

        int numElementsToAdd = scanner.nextInt();
```

```
System.out.println("Enter the elements to add:");

for (int i = 0; i < numElementsToAdd; i++) {
    String element = scanner.next();
    deque.addLast(element);
}

System.out.println("Deque elements after addition: " + deque);



System.out.print("Enter the number of elements to remove: ");
int numElementsToRemove = scanner.nextInt();

System.out.println("Removed elements:");

for (int i = 0; i < numElementsToRemove; i++) {
    String removedElement = deque.removeFirst();
    System.out.println(removedElement);
}

System.out.println("Deque elements after removal: " + deque);
```

```
        scanner.close();
    }
}
```

Output:



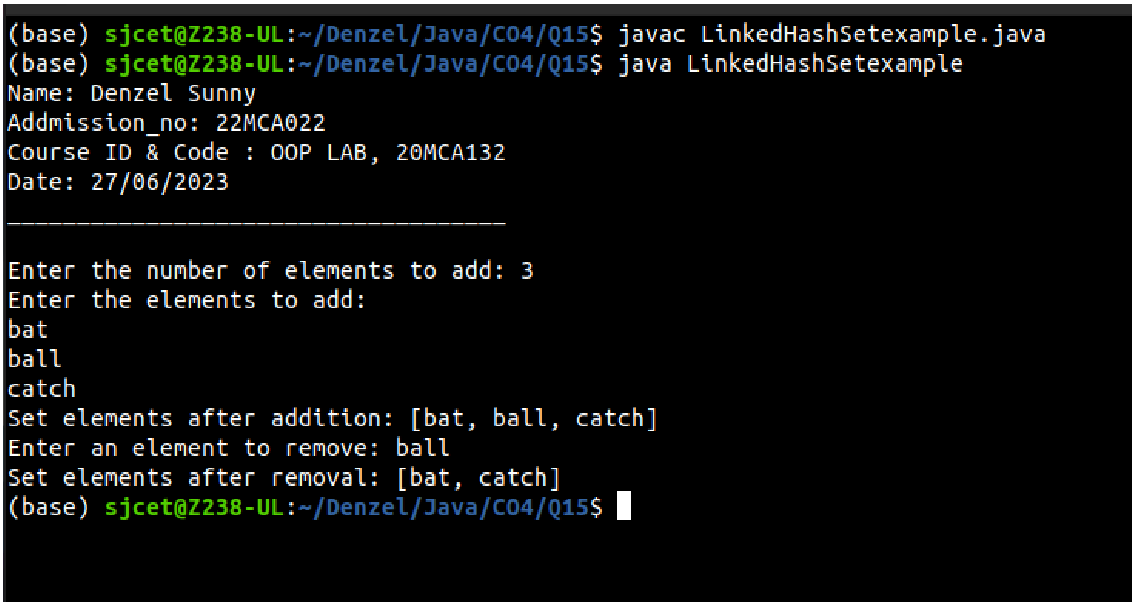
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q14$ javac DequeExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q14$ java DequeExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Enter the number of elements to add: 3
Enter the elements to add:
bat
ball
catch
Deque elements after addition: [bat, ball, catch]
Enter the number of elements to remove: 1
Removed elements:
bat
Deque elements after removal: [ball, catch]
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q14$
```

15. Program to demonstrate the creation of Set object using the LinkedHashset class

Code:

```
import java.util.LinkedHashSet;

import java.util.Scanner;

import java.util.Set;


public class LinkedHashSetexample {

    public static void main(String[] args) {


      System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("____________________________________\n");




        Set<String> set = new LinkedHashSet<>();



        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter the number of elements to add: ");
        int numElementsToAdd = scanner.nextInt();

        System.out.println("Enter the elements to add:");

        for (int i = 0; i < numElementsToAdd; i++) {
            String element = scanner.next();
            set.add(element);
        }

        System.out.println("Set elements after addition: " + set);

        System.out.print("Enter an element to remove: ");
        String elementToRemove = scanner.next();

        set.remove(elementToRemove);
        System.out.println("Set elements after removal: " + set);

        scanner.close();
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q15$ javac LinkedHashSetexample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q15$ java LinkedHashSetexample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Enter the number of elements to add: 3
Enter the elements to add:
bat
ball
catch
Set elements after addition: [bat, ball, catch]
Enter an element to remove: ball
Set elements after removal: [bat, catch]
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q15$
```

16. Write a Java program to compare two hash set

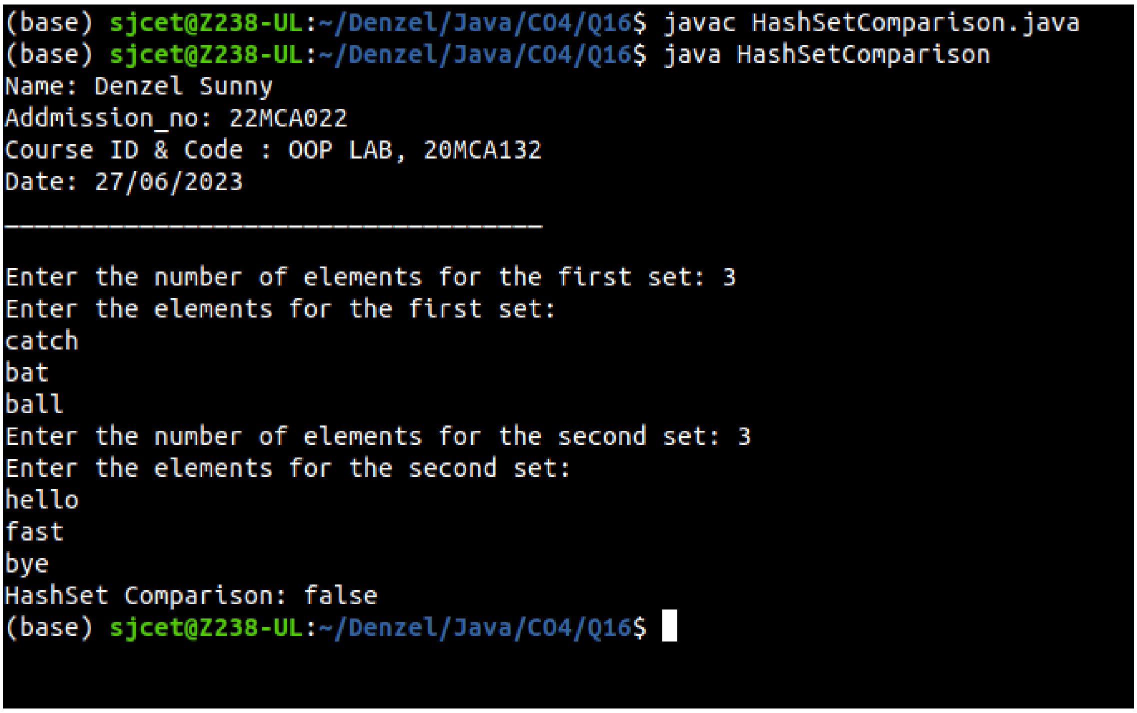
Code:

```
import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;


public class HashSetComparison {

    public static void main(String[] args) {


    System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("___________________________________\n");




        Set<String> set1 = new HashSet<>();


        Set<String> set2 = new HashSet<>();



        Scanner scanner = new Scanner(System.in);
```

```
System.out.print("Enter the number of elements for the first set: ");

int numElements1 = scanner.nextInt();


System.out.println("Enter the elements for the first set:");


for (int i = 0; i < numElements1; i++) {

    String element = scanner.next();

    set1.add(element);

}


System.out.print("Enter the number of elements for the second set: ");

int numElements2 = scanner.nextInt();


System.out.println("Enter the elements for the second set:");


for (int i = 0; i < numElements2; i++) {

    String element = scanner.next();

    set2.add(element);

}


boolean isEqual = set1.equals(set2);


System.out.println("HashSet Comparison: " + isEqual);
```

```
        scanner.close();
    }
}
```

Output:



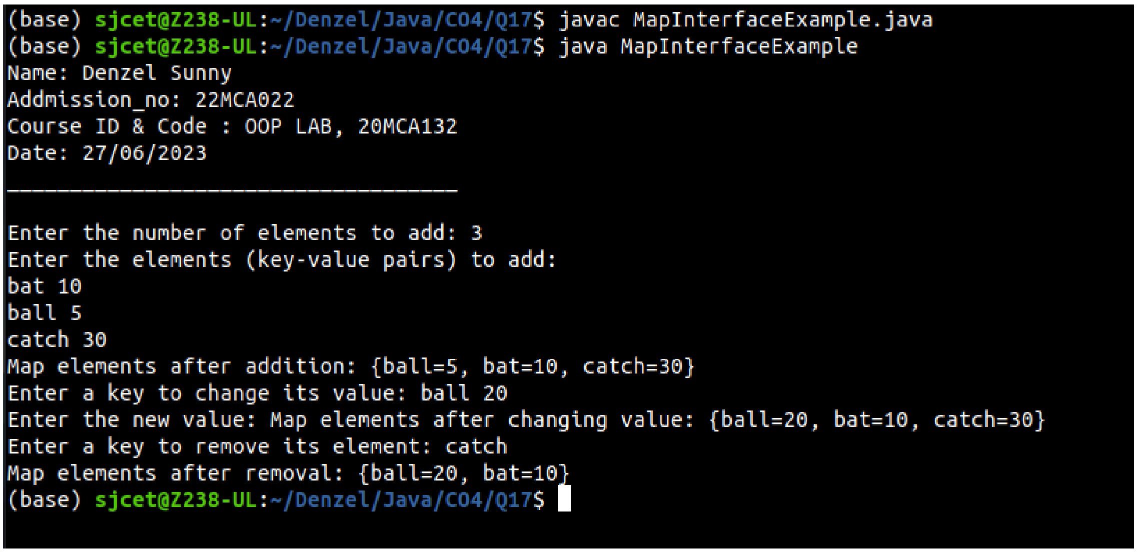
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q16$ javac HashSetComparison.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q16$ java HashSetComparison
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Enter the number of elements for the first set: 3
Enter the elements for the first set:
catch
bat
ball
Enter the number of elements for the second set: 3
Enter the elements for the second set:
hello
fast
bye
HashSet Comparison: false
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q16$
```

17. Program to demonstrate the working of Map interface by adding, changing and removing elements.

Code:

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

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

    System.out.println("Name: Denzel Sunny");
System.out.println("Addmission_no: 22MCA022");
System.out.println("Course ID & Code : OOP LAB, 20MCA132");
System.out.println("Date: 27/06/2023");
System.out.println("____________________________________\n");


        Map<String, Integer> map = new HashMap<>();



        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements to add: ");
        int numElementsToAdd = scanner.nextInt();

        System.out.println("Enter the elements (key-value pairs) to add:");

        for (int i = 0; i < numElementsToAdd; i++) {
            String key = scanner.next();
            int value = scanner.nextInt();
            map.put(key, value);
        }
        System.out.println("Map elements after addition: " + map);



        System.out.print("Enter a key to change its value: ");
        String keyToChange = scanner.next();
        System.out.print("Enter the new value: ");
        int newValue = scanner.nextInt();
        map.put(keyToChange, newValue);

        System.out.println("Map elements after changing value: " + map);



        System.out.print("Enter a key to remove its element: ");
        String keyToRemove = scanner.next();
        map.remove(keyToRemove);

        System.out.println("Map elements after removal: " + map);

        scanner.close();
    }
}
```

Output:



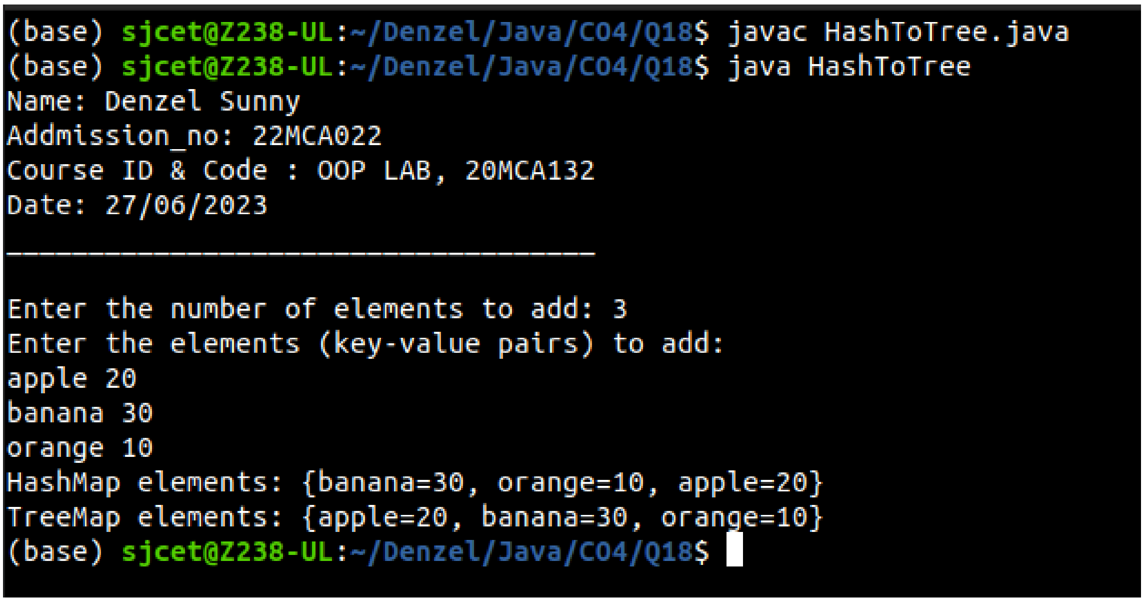
```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q17$ javac MapInterfaceExample.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q17$ java MapInterfaceExample
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023
____________________________________

Enter the number of elements to add: 3
Enter the elements (key-value pairs) to add:
bat 10
ball 5
catch 30
Map elements after addition: {ball=5, bat=10, catch=30}
Enter a key to change its value: ball 20
Enter the new value: Map elements after changing value: {ball=20, bat=10, catch=30}
Enter a key to remove its element: catch
Map elements after removal: {ball=20, bat=10}
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q17$
```

18. Program to Convert HashMap to TreeMap

Code:

```
import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

import java.util.TreeMap;


public class HashToTree {

    public static void main(String[] args) {


    System.out.println("Name: Denzel Sunny");

System.out.println("Addmission_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("____________________________________\n");



        Map<String, Integer> hashMap = new HashMap<>();



        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements to add: ");
        int numElementsToAdd = scanner.nextInt();

        System.out.println("Enter the elements (key-value pairs) to add:");

        for (int i = 0; i < numElementsToAdd; i++) {
            String key = scanner.next();
            int value = scanner.nextInt();
            hashMap.put(key, value);
        }

        System.out.println("HashMap elements: " + hashMap);



        Map<String, Integer> treeMap = new TreeMap<>(hashMap);

        System.out.println("TreeMap elements: " + treeMap);

        scanner.close();
    }
}
```

Output:



```
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q18$ javac HashToTree.java
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q18$ java HashToTree
Name: Denzel Sunny
Addmission_no: 22MCA022
Course ID & Code : OOP LAB, 20MCA132
Date: 27/06/2023

____________________________________

Enter the number of elements to add: 3
Enter the elements (key-value pairs) to add:
apple 20
banana 30
orange 10
HashMap elements: {banana=30, orange=10, apple=20}
TreeMap elements: {apple=20, banana=30, orange=10}
(base) sjcet@Z238-UL:~/Denzel/Java/CO4/Q18$
```