PROGRAM 17 :

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 :

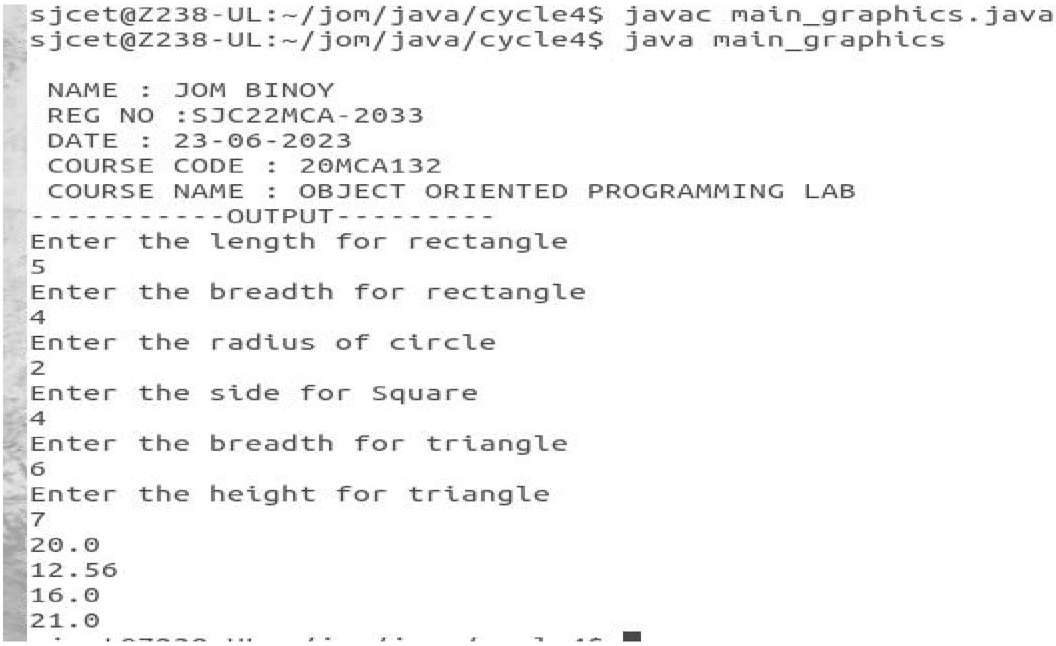
```
import package_graphics.*;
import java.util.*;
public class main_graphics {
   public static void main(String []args){
      package_graphics testObj = new package_graphics();
      int l,h,r,a,c,d;
      Scanner s=new Scanner(System.in);
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
      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(testObj.recArea(l,h));
      System.out.println(testObj.cirArea(r));
```

```
        System.out.println(testObj.squArea(a));
        System.out.println(testObj.triArea(c,d));
   }
}
```

package_graphics

```
package package_graphics;
interface interface_graphics{
   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_graphics {
    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);
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac main_graphics.java
sjcet@Z238-UL:~/jom/java/cycle4$ java main_graphics

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Enter the length for rectangle
5
Enter the breadth for rectangle
4
Enter the radius of circle
2
Enter the side for Square
4
Enter the breadth for triangle
6
Enter the height for triangle
7
20.0
12.56
16.0
21.0
sjcet@Z238-UL:~/jom/java/cycle4$
```

PROGRAM 18 :

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 :

```
import arithmetic.*;
public class Arithmetic_opt {
   public static void main(String[] args) {

System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 27-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        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);
   }
}
```

Arithmetic

```
package arithmetic;

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

Addition

```
package arithmetic;

public class Addition 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;
   }
}
```

Division

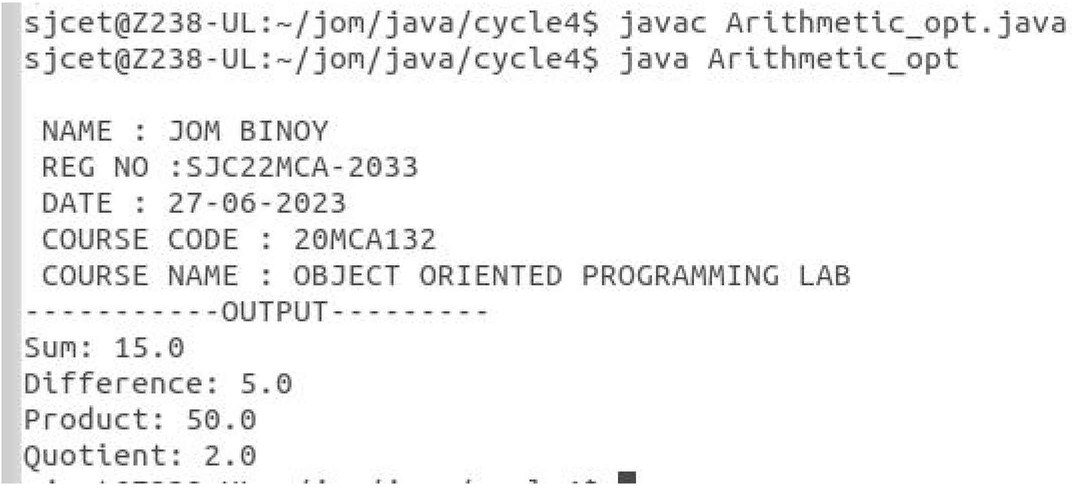
```
package arithmetic;

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;
   }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac Arithmetic_opt.java
sjcet@Z238-UL:~/jom/java/cycle4$ java Arithmetic_opt

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 27-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Sum: 15.0
Difference: 5.0
Product: 50.0
Quotient: 2.0
```

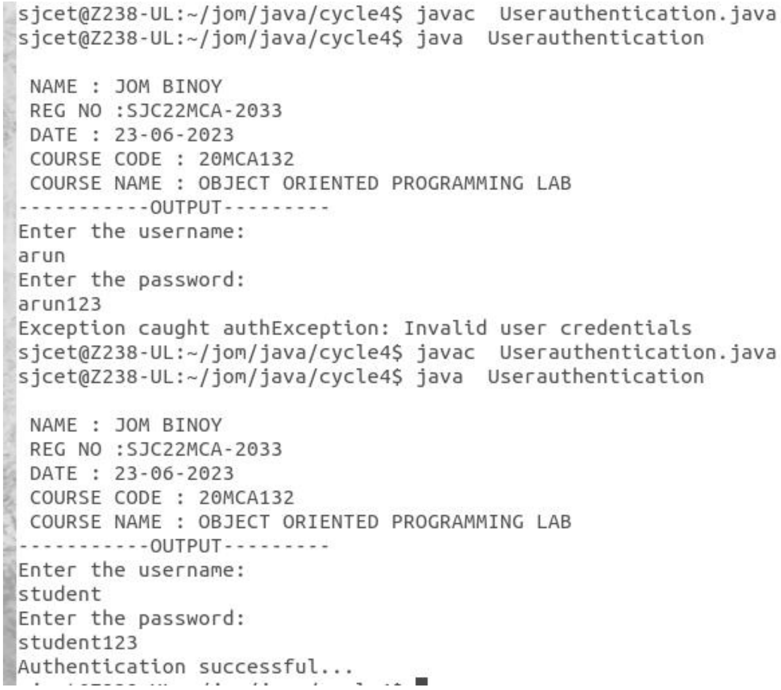
PROGRAM 19 :

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) {
String username = "student";
String passcode = "student123";
String user_name,password;
Scanner sc = new Scanner(System.in);
try
{
  System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
System.out.println("Enter the username:");
user_name = sc.nextLine();
System.out.println("Enter the 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 :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac  Userauthentication.java
sjcet@Z238-UL:~/jom/java/cycle4$ java  Userauthentication

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Enter the username:
arun
Enter the password:
arun123
Exception caught authException: Invalid user credentials
sjcet@Z238-UL:~/jom/java/cycle4$ javac  Userauthentication.java
sjcet@Z238-UL:~/jom/java/cycle4$ java  Userauthentication

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Enter the username:
student
Enter the password:
student123
Authentication successful...
```

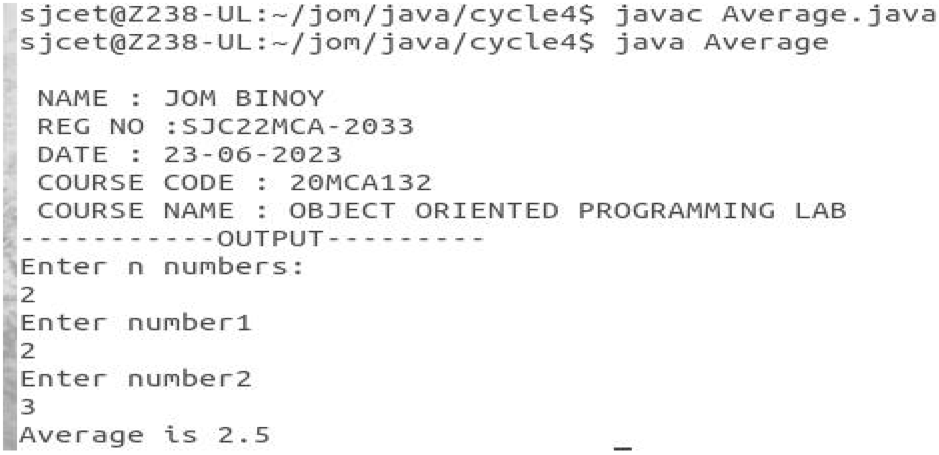
PROGRAM 20 :

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)
{
int i;
double sum=0,avg=0;
Scanner sc=new Scanner(System.in);
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
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 :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac Average.java
sjcet@Z238-UL:~/jom/java/cycle4$ java Average

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Enter n numbers:
2
Enter number1
2
Enter number2
3
Average is 2.5
```

PROGRAM 21 :

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

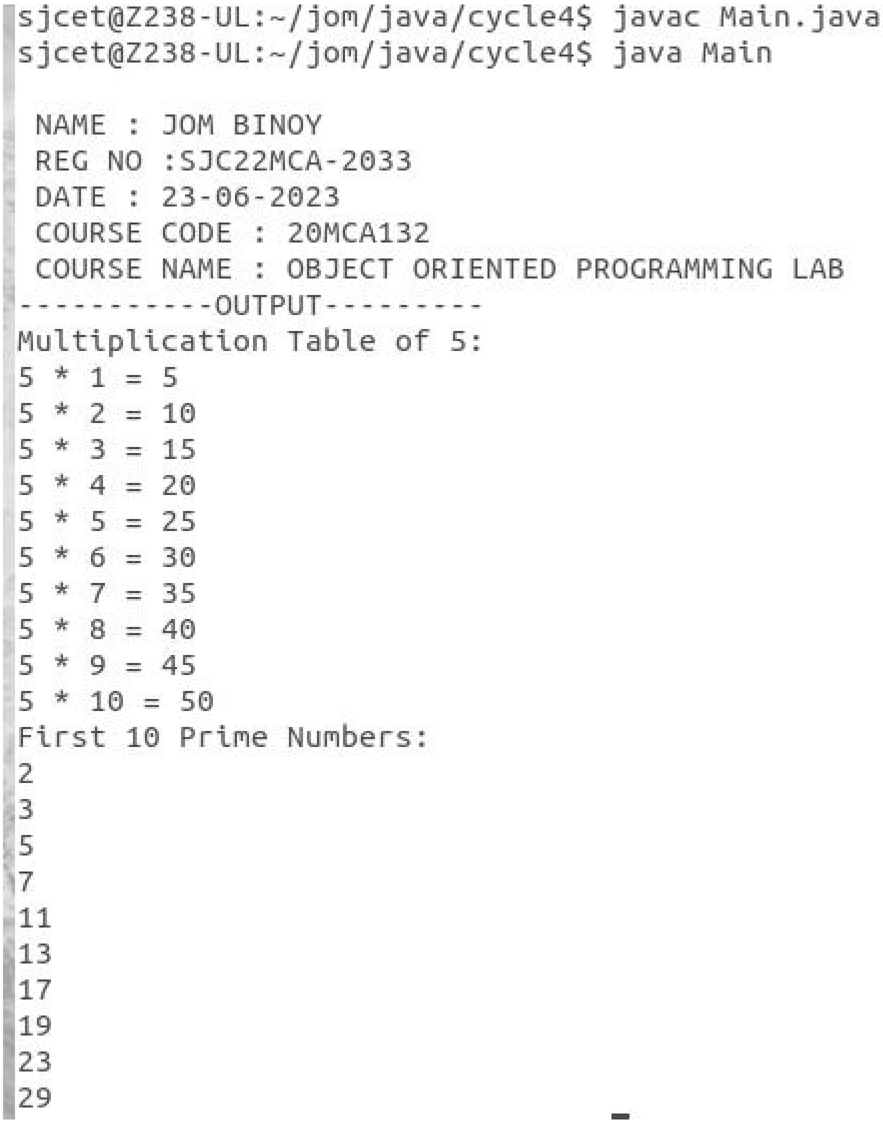
CODE :

```
class MultiplicationTableThread extends Thread {
    @Override
    public void run() {
  System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        System.out.println("Multiplication Table of 5:");
        for (int i = 1; i <= 10; i++) {
            System.out.println("5 * " + i + " = " + (5 * i));
        }
        new PrimeNumbersThread(10).start();
    }
}
class PrimeNumbersThread extends Thread {
    private int count;

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

    @Override
    public void run() {
        System.out.println("First " + count + " Prime Numbers:");
        int num = 2;
        int primeCount = 0;
        while (primeCount < count) {
            if (isPrime(num)) {
                System.out.println(num);
                primeCount++;
            }
            num++;
        }
    }
    private boolean isPrime(int number) {
        if (number < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
public class Main {
    public static void main(String[] args) {
        new MultiplicationTableThread().start();
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac Main.java
sjcet@Z238-UL:~/jom/java/cycle4$ java Main

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
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
```

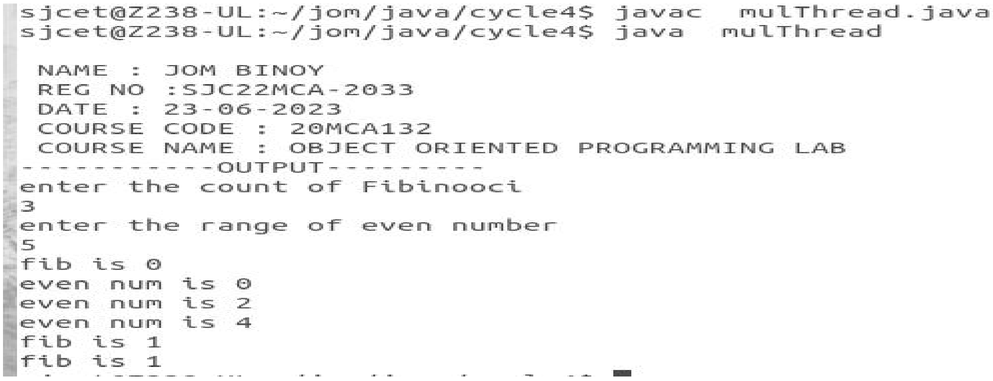
PROGRAM 22 :

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 Fib extends Thread{
int f,n1=0,n2=1,n3;
Fib(int c){
this.f=c;
}
public void run(){
System.out.println("fib is "+n1);
System.out.println("fib is "+n2);
for(int i=2;i<this.f;++i) {
n3=n1+n2;
System.out.println("fib is "+n3);
n1=n2;
n2=n3;
}
}
}
class even extends Thread{
int range;
even(int range){
this.range=range;
}
public void run(){
for(int i=0;i<this.range;i++){
if(i%2==0){
System.out.println("even num is "+i);
}
}
}
}
public class mulThread {
public static void main(String [] args){
int c,range;
Scanner sc=new Scanner(System.in);
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
System.out.println("enter the count of Fibinooci");
c=sc.nextInt();
Fib fi=new Fib(c);
System.out.println("enter the range of even number");
range=sc.nextInt();
even ev = new even(range);
fi.start();
ev.start();
}
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac  mulThread.java
sjcet@Z238-UL:~/jom/java/cycle4$ java  mulThread

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
enter the count of Fibinooci
3
enter the range of even number
5
fib is 0
even num is 0
even num is 2
even num is 4
fib is 1
fib is 1
```

PROGRAM 23 :

Producer/Consumer using ITC

CODE :

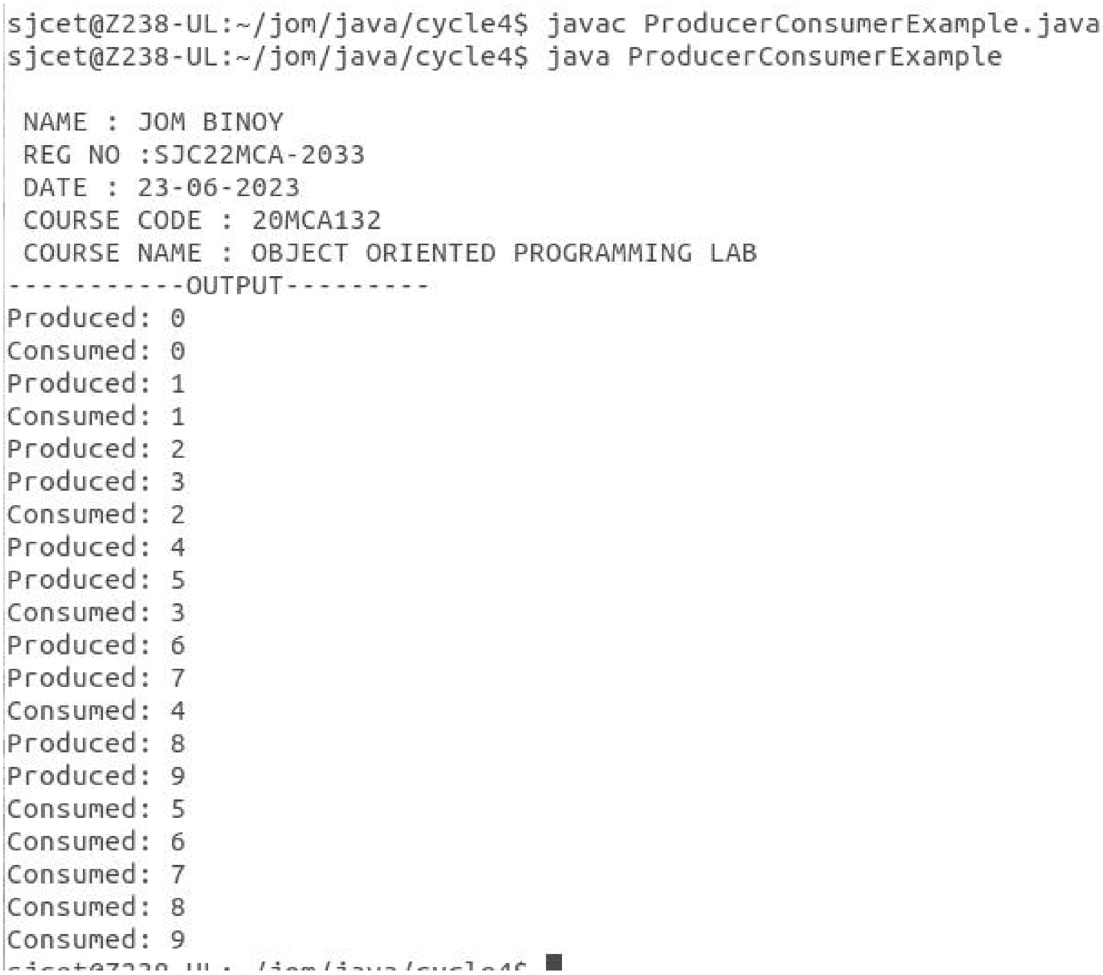
```
import java.util.LinkedList;
class Buffer {
    private LinkedList<Integer> buffer;
    private int capacity;
    public Buffer(int capacity) {
        this.buffer = new LinkedList<>();
        this.capacity = capacity;
    }
    public void produce(int value) throws InterruptedException {
        synchronized (this) {
            while (buffer.size() == capacity) {
                wait();
            }
            buffer.add(value);
            System.out.println("Produced: " + value);
            notifyAll();
        }
    }
    public void consume() throws InterruptedException {
        synchronized (this) {
            while (buffer.isEmpty()) {
                wait();
            }
            int value = buffer.removeFirst();
            System.out.println("Consumed: " + value);
            notifyAll();
        }
    }
}

class Producer implements Runnable {
    private Buffer buffer;
    private int numProductions;

    public Producer(Buffer buffer, int numProductions) {
        this.buffer = buffer;
        this.numProductions = numProductions;
    }
    @Override
    public void run() {
        for (int i = 0; i < numProductions; i++) {
            try {
                buffer.produce(i);
                Thread.sleep(1000); // Simulate production time
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
class Consumer implements Runnable {
    private Buffer buffer;
    private int numConsumptions;

    public Consumer(Buffer buffer, int numConsumptions) {
        this.buffer = buffer;
        this.numConsumptions = numConsumptions;
    }
    @Override
    public void run() {
        for (int i = 0; i < numConsumptions; i++) {
            try {
                buffer.consume();
                Thread.sleep(2000); // Simulate consumption time
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
public class ProducerConsumerExample {
    public static void main(String[] args) {
    System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        Buffer buffer = new Buffer(5);
        int numProductions = 10;
        int numConsumptions = 10;
        Producer producer = new Producer(buffer, numProductions);
        Consumer consumer = new Consumer(buffer, numConsumptions);
        Thread producerThread = new Thread(producer);
        Thread consumerThread = new Thread(consumer);
        producerThread.start();
        consumerThread.start();
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac ProducerConsumerExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java ProducerConsumerExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Produced: 0
Consumed: 0
Produced: 1
Consumed: 1
Produced: 2
Produced: 3
Consumed: 2
Produced: 4
Produced: 5
Consumed: 3
Produced: 6
Produced: 7
Consumed: 4
Produced: 8
Produced: 9
Consumed: 5
Consumed: 6
Consumed: 7
Consumed: 8
Consumed: 9
sjcet@Z238-UL:~/jom/java/cycle4$
```

PROGRAM 24 :

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

CODE :

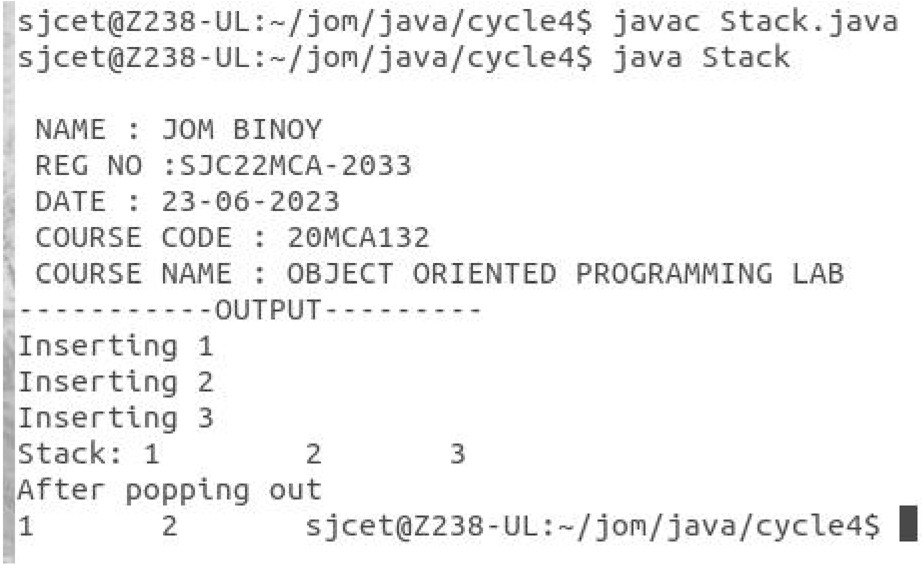
```
class Stack {
private int arr[];
private int top;
private int capacity;
Stack(int size) {
arr = new int[size];
capacity = size;
top = -1;
}
public void push(int x) {
if (isFull()) {

System.out.println("Stack OverFlow");
System.exit(1);
}

System.out.println("Inserting " + x);
arr[++top] = x;
}
public int pop() {
if (isEmpty()) {
System.out.println("STACK EMPTY");
System.exit(1);
}
return arr[top--];
}
public int getSize() {
```

```
return top + 1;
}
public Boolean isEmpty() {
return top == -1;
}
public Boolean isFull() {
return top == capacity - 1;
}
public void printStack() {
for (int i = 0; i <= top; i++) {
System.out.print(arr[i] + "\t");
}
}
public static void main(String[] args) {
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
Stack stack = new Stack(5);
stack.push(1);
stack.push(2);
stack.push(3);
System.out.print("Stack: ");
stack.printStack();
stack.pop();
System.out.println("\nAfter popping out");
stack.printStack();
}
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac Stack.java
sjcet@Z238-UL:~/jom/java/cycle4$ java Stack

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Inserting 1
Inserting 2
Inserting 3
Stack: 1        2       3
After popping out
1       2       sjcet@Z238-UL:~/jom/java/cycle4$ 
```

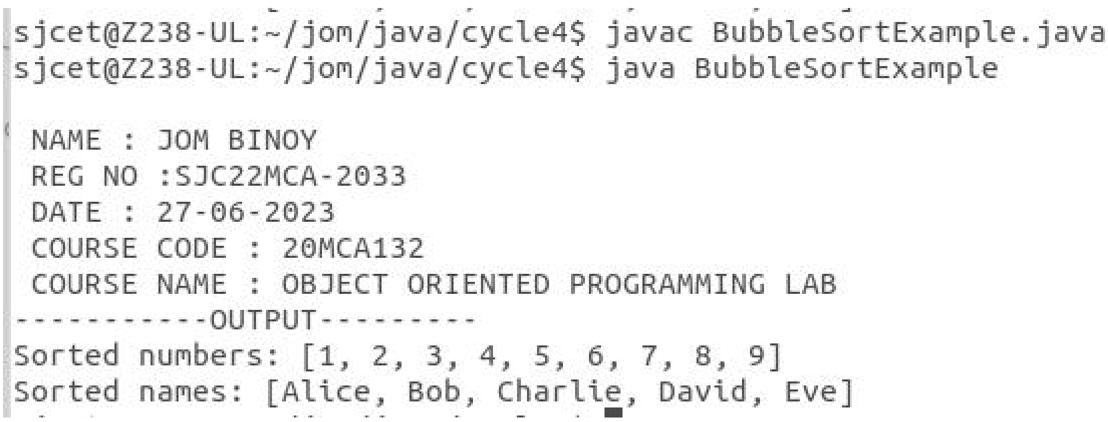
PROGRAM 25 :

Using generic method perform Bubble sort.

CODE :

```
import java.util.Arrays;
public class BubbleSortExample {
   public static <T extends Comparable<T>> void bubbleSort(T[] array) {
       int n = array.length;
       for (int i = 0; i < n - 1; i++) {
          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;
             }
          }
       }
   }
   public static void main(String[] args) {
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 27-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
       Integer[] numbers = {4, 2, 6, 1, 9, 3, 8, 5, 7};
       bubbleSort(numbers);
       System.out.println("Sorted numbers: " + Arrays.toString(numbers));
       String[] names = {"Alice", "Bob", "Charlie", "David", "Eve"};
       bubbleSort(names);
       System.out.println("Sorted names: " + Arrays.toString(names));
   }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac BubbleSortExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java BubbleSortExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 27-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Sorted numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9]
Sorted names: [Alice, Bob, Charlie, David, Eve]
```

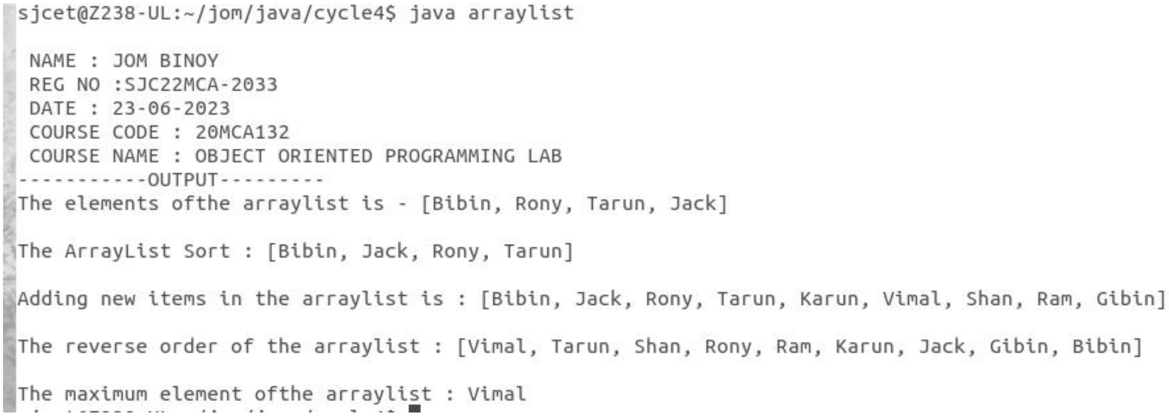
PROGRAM 26 :

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

CODE :

```
import java.util.*;
public class arraylist{
public static void main(String[] args) {
ArrayList<String> arrayList= new ArrayList<>();
arrayList.add("Bibin");
arrayList.add("Rony");
arrayList.add("Tarun");
arrayList.add("Jack");
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
System.out.println("The elements ofthe arraylist is - "+arrayList);
Collections.sort(arrayList);
System.out.println("\nThe ArrayList Sort : "+arrayList);
Collections.addAll(arrayList,"Karun","Vimal","Shan","Ram","Gibin");
System.out.println("\nAdding new items in the arraylist is : "+arrayList);
Collections.sort(arrayList, Collections.reverseOrder());
System.out.println("\nThe reverse order of the arraylist : "+arrayList);
System.out.println("\nThe maximum element ofthe arraylist :
"+Collections.max(arrayList));
}
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ java arraylist

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
The elements ofthe arraylist is - [Bibin, Rony, Tarun, Jack]

The ArrayList Sort : [Bibin, Jack, Rony, Tarun]

Adding new items in the arraylist is : [Bibin, Jack, Rony, Tarun, Karun, Vimal, Shan, Ram, Gibin]

The reverse order of the arraylist : [Vimal, Tarun, Shan, Rony, Ram, Karun, Jack, Gibin, Bibin]

The maximum element ofthe arraylist : Vimal
```

PROGRAM 27 :

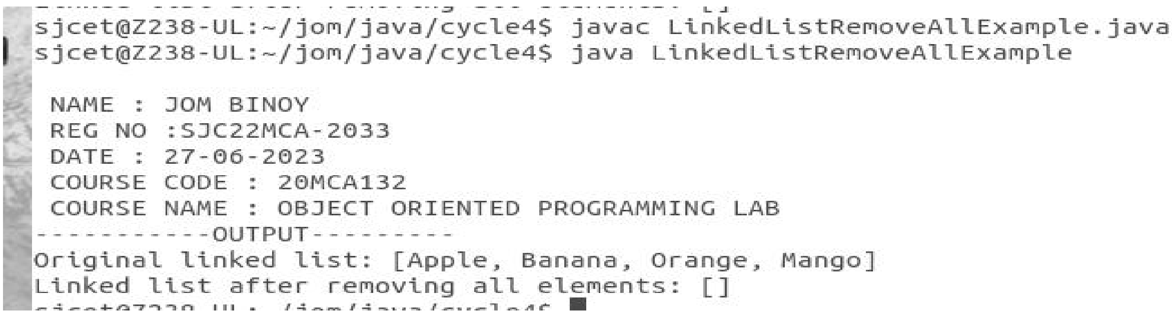
Program to remove all the elements from a linked list

CODE :

```
import java.util.LinkedList;

public class LinkedListRemoveAllExample {
   public static void main(String[] args) {
       LinkedList<String> linkedList = new LinkedList<>();
       linkedList.add("Apple");
       linkedList.add("Banana");
       linkedList.add("Orange");
       linkedList.add("Mango");
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 27-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
       System.out.println("Original linked list: " + linkedList);
            linkedList.clear();
       System.out.println("Linked list after removing all elements: " + linkedList);
   }
}
```

OUTPUT :



```
Linked list after removing all elements: []
sjcet@Z238-UL:~/jom/java/cycle4$ javac LinkedListRemoveAllExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java LinkedListRemoveAllExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 27-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Original linked list: [Apple, Banana, Orange, Mango]
Linked list after removing all elements: []
sjcet@Z238-UL:~/jom/java/cycle4$ █
```

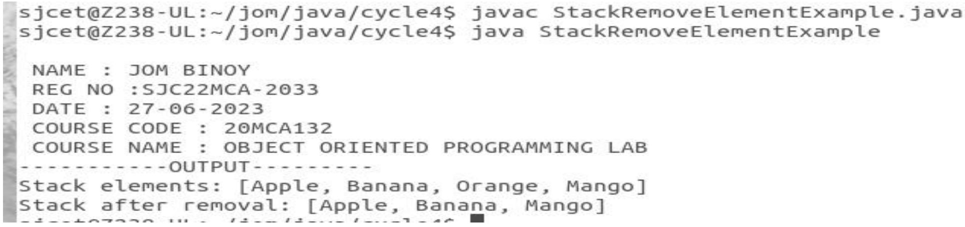
PROGRAM 28 :

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

CODE :

```
import java.util.Stack;
public class StackRemoveElementExample {
    public static void main(String[] args) {
     Stack<String> stack = new Stack<>();
        stack.push("Apple");
        stack.push("Banana");
        stack.push("Orange");
        stack.push("Mango");
        System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 27-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        System.out.println("Stack elements: " + stack);
        int positionToRemove = 2; // Position starts from 1
        removeElement(stack, positionToRemove);
        System.out.println("Stack after removal: " + stack);
    }
    public static void removeElement(Stack<String> stack, int position) {
        if (stack.isEmpty() || position <= 0 || position > stack.size()) {
            System.out.println("Invalid position or stack is empty.");
            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 :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac StackRemoveElementExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java StackRemoveElementExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 27-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Stack elements: [Apple, Banana, Orange, Mango]
Stack after removal: [Apple, Banana, Mango]
sjcet@Z238-UL:~/jom/java/cycle4$
```

PROGRAM 29 :

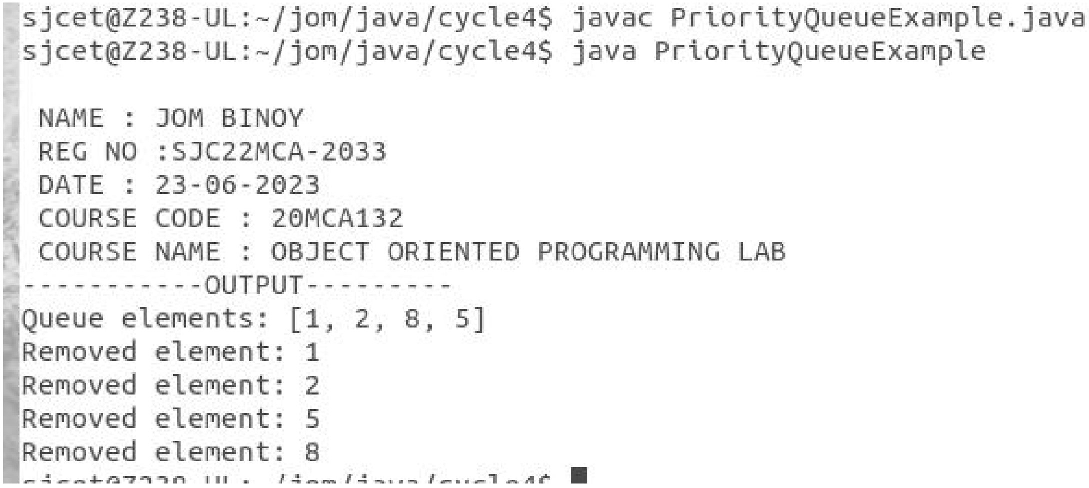
Program to demonstrate the creation of queue object using the PriorityQueue class

CODE :

```
import java.util.PriorityQueue;
import java.util.Queue;

public class PriorityQueueExample {
    public static void main(String[] args) {
        Queue<Integer> queue = new PriorityQueue<>();
        queue.offer(5);
        queue.offer(2);
        queue.offer(8);
        queue.offer(1);
 System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        System.out.println("Queue elements: " + queue);
        while (!queue.isEmpty()) {
            int element = queue.poll();
            System.out.println("Removed element: " + element);
        }
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac PriorityQueueExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java PriorityQueueExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Queue elements: [1, 2, 8, 5]
Removed element: 1
Removed element: 2
Removed element: 5
Removed element: 8
sjcet@Z238-UL:~/jom/java/cycle4$ █
```

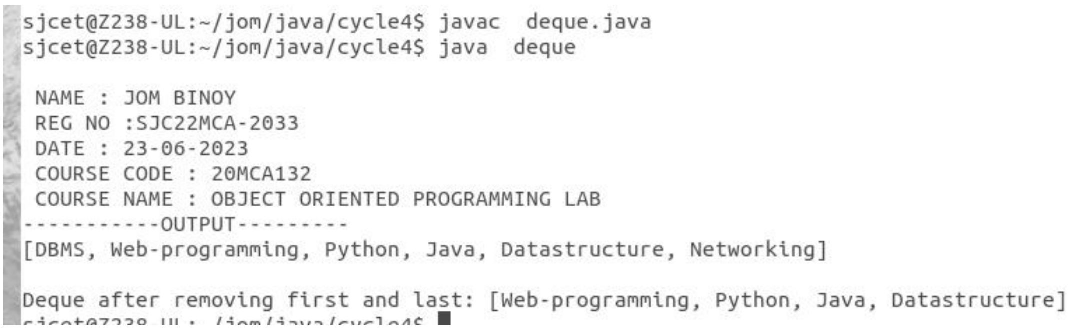
PROGRAM 30 :

Program to demonstrate the addition and deletion of elements in deque

CODE :

```
import java.util.*;
class deque
{
public static void main(String[] args)
{
Deque<String> deque = new LinkedList<String>();
deque.add("Java");
deque.addFirst("Python");
deque.addLast("Datastructure");
deque.push("Web-programming");
deque.offer("Networking");
deque.offerFirst("DBMS");
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033
\n DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME :
OBJECT ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
System.out.println(deque + "\n");
deque.removeFirst();
deque.removeLast();
System.out.println("Deque after removing " + "first and last: " + deque);
}
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac  deque.java
sjcet@Z238-UL:~/jom/java/cycle4$ java  deque

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
[DBMS, Web-programming, Python, Java, Datastructure, Networking]

Deque after removing first and last: [Web-programming, Python, Java, Datastructure]
sjcet@Z238-UL:~/jom/java/cycle4$
```

PROGRAM 31 :

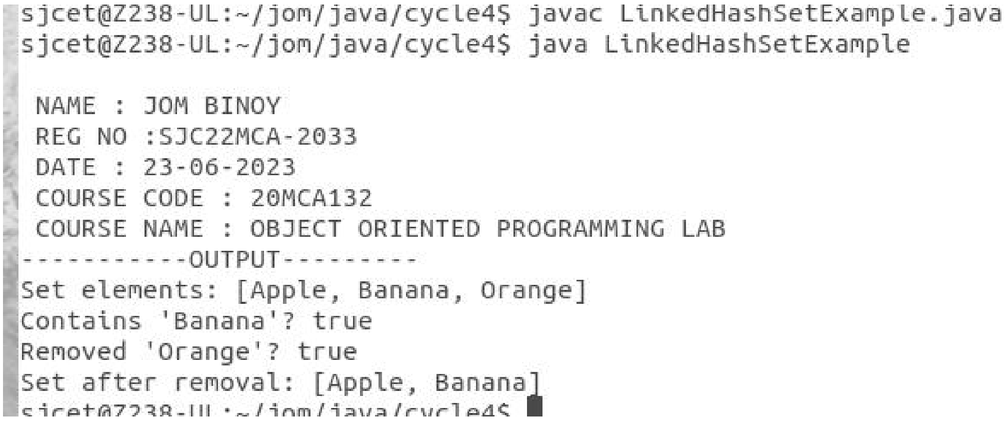
Program to demonstrate the creation of Set object using the LinkedHashset class.

CODE :

```
import java.util.LinkedHashSet;
import java.util.Set;

public class LinkedHashSetExample {
  public static void main(String[] args) {
       Set<String> set = new LinkedHashSet<>();
    set.add("Apple");
    set.add("Banana");
    set.add("Orange");
    set.add("Apple"); // Adding a duplicate element
    System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
    System.out.println("Set elements: " + set);
    boolean containsBanana = set.contains("Banana");
    System.out.println("Contains 'Banana'? " + containsBanana);
    boolean removedOrange = set.remove("Orange");
    System.out.println("Removed 'Orange'? " + removedOrange);
    System.out.println("Set after removal: " + set);
  }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac LinkedHashSetExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java LinkedHashSetExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Set elements: [Apple, Banana, Orange]
Contains 'Banana'? true
Removed 'Orange'? true
Set after removal: [Apple, Banana]
sjcet@Z238-UL:~/jom/java/cycle4$
```

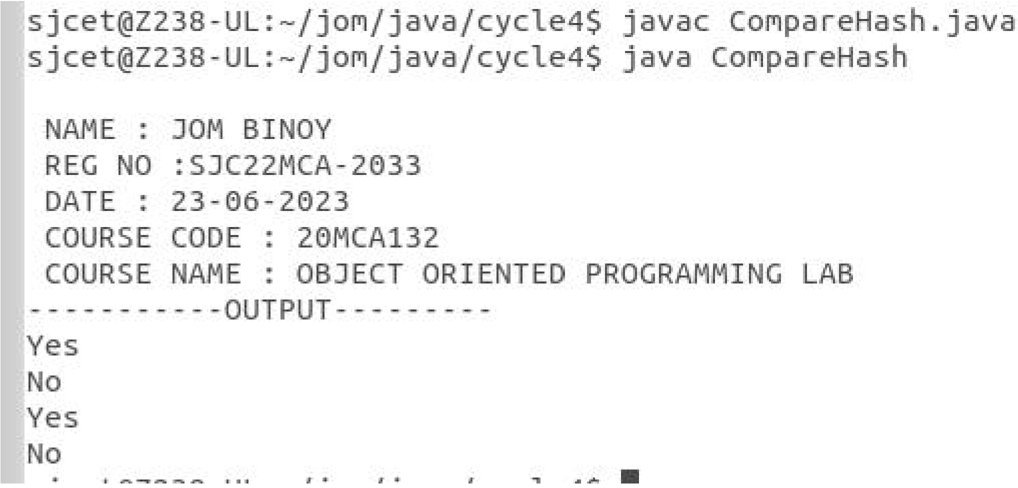
PROGRAM 32 :

Write a Java program to compare two hash set

CODE :

```
import java.util.*;
public class CompareHash {
public static void main(String[] args) {
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n
DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT
ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
HashSet<String> h_set = new HashSet<String>();
h_set.add("Red");
h_set.add("Green");
h_set.add("Black");
h_set.add("White");
HashSet<String>h_set2 = new HashSet<String>();
h_set2.add("Red");
h_set2.add("Pink");
h_set2.add("Black");
h_set2.add("Orange");
HashSet<String>result_set = new HashSet<String>();
for (String element : h_set){
System.out.println(h_set2.contains(element) ? "Yes" : "No");
}
}
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac CompareHash.java
sjcet@Z238-UL:~/jom/java/cycle4$ java CompareHash

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Yes
No
Yes
No
```

PROGRAM 33 :

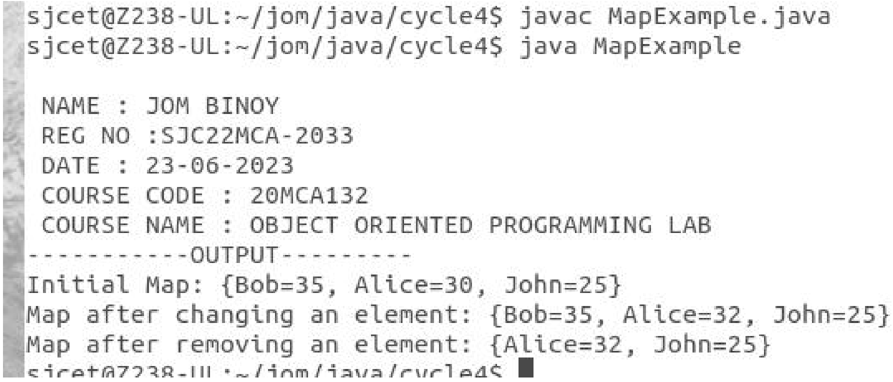
Program to demonstrate the working of Map interface by adding, changing and removing elements.

CODE :

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

public class MapExample {
   public static void main(String[] args) {
            Map<String, Integer> map = new HashMap<>();
        map.put("John", 25);
        map.put("Alice", 30);
        map.put("Bob", 35);
        System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-
2033 \n DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE
NAME : OBJECT ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        System.out.println("Initial Map: " + map);
        map.put("Alice", 32);
        System.out.println("Map after changing an element: " + map);
        map.remove("Bob");
        System.out.println("Map after removing an element: " + map);
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac MapExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java MapExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
Initial Map: {Bob=35, Alice=30, John=25}
Map after changing an element: {Bob=35, Alice=32, John=25}
Map after removing an element: {Alice=32, John=25}
sjcet@Z238-UL:~/jom/java/cycle4$ 
```

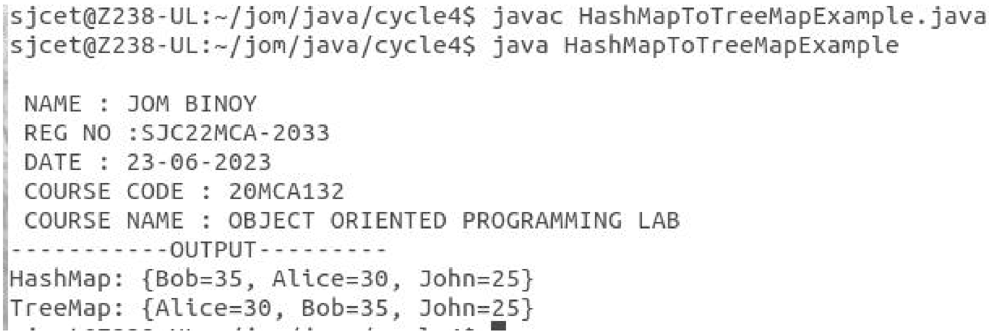
PROGRAM 34 :

Program to Convert HashMap to TreeMap

CODE :

```
import java.util.HashMap;
import java.util.Map;
import java.util.TreeMap;
public class HashMapToTreeMapExample {
    public static void main(String[] args) {
        Map<String, Integer> hashMap = new HashMap<>();
        hashMap.put("John", 25);
        hashMap.put("Alice", 30);
        hashMap.put("Bob", 35);
        Map<String, Integer> treeMap = new TreeMap<>(hashMap);
 System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033
\n DATE : 23-06-2023 \n COURSE CODE : 20MCA132\n COURSE NAME :
OBJECT ORIENTED PROGRAMMING LAB");
System.out.println("-----------OUTPUT---------");
        System.out.println("HashMap: " + hashMap);
        System.out.println("TreeMap: " + treeMap);
    }
}
```

OUTPUT :



```
sjcet@Z238-UL:~/jom/java/cycle4$ javac HashMapToTreeMapExample.java
sjcet@Z238-UL:~/jom/java/cycle4$ java HashMapToTreeMapExample

 NAME : JOM BINOY
 REG NO :SJC22MCA-2033
 DATE : 23-06-2023
 COURSE CODE : 20MCA132
 COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----------OUTPUT---------
HashMap: {Bob=35, Alice=30, John=25}
TreeMap: {Alice=30, Bob=35, John=25}
```