Sree Narayana Gurukulam College of Engineering, Kadayiruppu,

Kolencherry 682311
(Affliliated to APJ Abdul Kalam Technological University)

Department of Computer Science Engineering

CSL 203 OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)
Lab Manual

ACADEMIC YEAR 2020

CYCLE

Part A

<u>Basic programs using datatypes, operators, and control statements in Java.</u>

1. Write a Java program that checks whether a given string is a palindrome or not.

Program

```
public class Palindrome {
  public static void main(String[] args) {
    String str = "Malayalam";
    StringBuffer newStr =new StringBuffer();
    for(int i = str.length()-1; i >= 0 ; i--) {
      newStr = newStr.append(str.charAt(i));
    }
    if(str.equalsIgnoreCase(newStr.toString())) {
      System.out.println("String is palindrome");
    } else {
      System.out.println("String is not palindrome");
    }
  }
}
```

Output

user@user-System-Product-Name:~/Desktop/javalab$javac Palindrome.java

user@user-System-Product-Name:~/Desktop/javalab$java Palindrome

String is palindrome

2. Write a Java Program to find the frequency of a given character in a string. **

Program

```
import java.util.Scanner;
public class FrequencyOfACharacter {
  public static void main(String args[]){
    System.out.println("Enter a string value ::");
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();

    System.out.println("Enter a particular character ::");
    char character = sc.nextLine().charAt(0);
    int count = 0;

    for (int i=0; i<str.length(); i++){
      if(character == str.charAt(i)){
        count++;
      }
    }
    System.out.println("Frequency of the give character:: "+count);
  }
}
```

Output

user@user-System-Product-Name:~/Desktop/javalab$javac CharFrequency.java

user@user-System-Product-Name:~/Desktop/javalab$java CharFrequency

Enter a string value ::

Malayalam

Enter a particular character ::

a

Frequency of the give character:: 4

3. Write a Java program to multiply two given matrices.

Program

```
public class MatrixMul {

	public static void main(String[] args) {
		int[][] x = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
		int[][] y = {{ 2, 2, 2}, {1, 1,1}, {3,3, 3}};

		int[][] multi = new int[3][3];
		int i, j,k;

		for(i = 0; i < x.length; i++)
		{
			for(j = 0; j < x[0].length; j++)
			{
			  multi[i][j]=0;

			  for(k=0;k< x[0].length;k++)
				multi[i][j] = multi[i][j] + x[i][k] * y[k][j];
			}
		}
		System.out.println("Product Matrix ");

		for(i = 0; i < x.length; i++)
		{
			for(j = 0; j < x[0].length; j++)
			{
				System.out.format("%d \t", multi[i][j]);
			}
			System.out.println("");
		}
	}
}
```

Output

user@user-System-Product-Name:~/Desktop/javalab$javac MatMul.java

user@user-System-Product-Name:~/Desktop/javalab$java MatMul

Product Matrix

2 4 6

4 5 6

21 24 27

Part B Object Oriented Programming Concepts

4. Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. (inheritance). **

Program

```
import java.util.*;
class Employee
{
String name;
int age;
int phone;
String address;
int salary;
void printSalary()
{
System.out.println("SALARY DETAILS");
System.out.println("---------------");
System.out.println("NAME :  "+name);
System.out.println("AGE    :  "+age);
System.out.println("PHONE NO     :  "+phone);
System.out.println("ADDRESS      :  "+address);
System.out.println("SALARY       :  "+salary);
}
}
class Officer extends Employee
{
String specialization;
}
class Managers extends Employee
{
String department;
}

public class Manager
{
public static void main(String args[])
{
Managers m1=new Managers();
m1.name="Geena";
m1.age=20;
m1.phone=897654;
m1.address="Sangeetham";
m1.salary=10000;
m1.printSalary();
Officer o1=new Officer();
```

```
o1.name="Teena";
o1.age=25;
o1.phone=78652;
o1.address="Karunyam";
o1.salary=20000;
o1.printSalary();
}
}
```

Output

user@user-System-Product-Name:~/Desktop/javalab$javac Manager.java

user@user-System-Product-Name:~/Desktop/javalab$java Manager

SALARY DETAILS

NAME : Geena

AGE : 20

PHONE NO : 897654

ADDRESS : Sangeetham

SALARY : 10000

SALARY DETAILS

NAME : Teena

AGE : 25

PHONE NO : 78652

ADDRESS : Karunyam

SALARY : 20000

5. Write a java program to create an abstract class named Shape that contains an emptymethod named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures. (polymorphism). **

Program

```
import java.util.*;
abstract class Shape
{
void numberOfSides(){}
}
class Rectangle extends Shape
{
void numberOfSides()
{
System.out.println("calling method of class Rectangle");
System.out.println("FOUR SIDES");
}
}
class Triangle extends Shape
{
void numberOfSides()
{
System.out.println("calling method of class Triangle");
System.out.println("THREE SIDES");
}
}
class Hexagon extends Shape
{
void numberOfSides()
{
System.out.println("calling method of class Hexagon");
System.out.println("SIX SIDES");
}
}
public class Abstractclass
{
public static void main(String args[])
{
Rectangle r1=new Rectangle();
r1.numberOfSides();
Triangle t1=new Triangle();
t1.numberOfSides();
Hexagon h1=new Hexagon ();
h1.numberOfSides();
}
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Abstractclass.java
user@user-System-Product-Name:~/Desktop/javalab$java Abstractclass
calling method of class Rectangle
FOUR SIDES
calling method of class Triangle
THREE SIDES
calling method of class Hexagon
SIX SIDES
```

Part C File Handling s as well as input and output management methods

6. Write a file handling program in Java with reader/writer.

Program

```
// Creating a text File using FileWriter
import java.io.FileWriter;
import java.io.FileReader;
import java.io.IOException;
class File1
{
public static void main(String[] args) throws IOException
{
// Accept a string
String str = "File Handling in Java using "+
" FileWriter and FileReader\n";
// attach a file to FileWriter
FileWriter fw=new FileWriter("output.txt");

// read characterwise from string and write
// into FileWriter
for (int i = 0; i < str.length(); i++)
fw.write(str.charAt(i));

System.out.println("Data written successfully");
//close the file
fw.close();
FileReader fr=null;

     int ch;
          fr = new FileReader("output.txt");
       // read from FileReader till the end of file
       while ((ch=fr.read())!=-1)
          System.out.print((char)ch);
  System.out.println("Data Read successfully");
       // close the file
       fr.close();
}
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac File1.java

user@user-System-Product-Name:~/Desktop/javalab$ java File1

Data written successfully

File Handling in Java using  FileWriter and FileReader

Data Read successfully
```

7. Write a Java program that read from a file and write to file by handling all file related exceptions.

Program

```
import java.io.*;
public class Pgm7
{
   public static void main(String[] args)
{
      try {
         Writer w = new FileWriter("output.txt");
         String content = "I love my country";
         w.write(content);
         w.close();
         System.out.println("Done");
         }
         catch (IOException e)
      {
         e.printStackTrace();
      }
 try {
         Reader reader = new FileReader("output.txt");
         int data = reader.read();
         while (data != -1) {
            System.out.print((char) data);
            data = reader.read();
         }
         reader.close();
      } catch (Exception ex) {
         System.out.println(ex.getMessage());
      }

   }
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Pgm7.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm7
Done
I love my country
```

8. Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util).

Program

```
/Program to print the read numbers and find sum
import java.util.*;
class Sumtoken
{
public static void main(String args[])
{
Scanner scr=new Scanner(System.in);
System.out.println("\nEnter sequence of integers with space b/w them and press enter : ");
String digit=scr.nextLine();
 StringTokenizer token=new StringTokenizer(digit);
 int dig=0,sum=0;
 System.out.println("\nEntered digits are : ");
 while(token.hasMoreTokens())
 {
String s=token.nextToken();
 dig=Integer.parseInt(s);
 System.out.print(dig+" ");
 sum=sum+dig;
 }
      System.out.println();
System.out.println("Sum is : "+sum);
}
}
```

Output:

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Sumtoken.java
user@user-System-Product-Name:~/Desktop/javalab$ javac Sumtoke Sumtoken
Enter sequence of integers with space b/w them and press enter :
1 2 3 4 5 6 7 8 9 10

Entered digits are :
1 2 3 4 5 6 7 8 9 10

Sum is : 55
```

Part D Exception handling and multi-threading applications:

9. Write a Java program that shows the usage of try, catch, throws and finally.

Program

```
import java.util.*;
class AgeVerify
{
void vote(int age) throws IllegalAccessException
{
  try
  {
  if (age < 18)
  {throw new IllegalAccessException("You must be at least 18 years old to
  vote.");}
  else
  System.out.println("You can vote");
  }
  catch(Exception e)
  {
  System.out.println("EXCEPTION OCCURED: "+e);
  }
  finally
  {
  System.out.println("Verification is completed");
  }
}
}

public class Pgm9
{
public static void main(String args[]) throws IllegalAccessException
{
Scanner sc=new Scanner(System.in);
int age;
//System.out.println("Enter the age of voter");
System.out.println("Enter the age of the voter");
age=sc.nextInt();
 AgeVerify av=new AgeVerify();
 av.vote(age);
System.out.println("Completed successfully");
}
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Pgm9.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm9
```

```
Enter the age of the voter
12
EXCEPTION OCCURED: java.lang.IllegalAccessException: You must be at least 18 years old to vote.
Verification is completed
Completed successfully
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm9
Enter the age of the voter
36
You can vote
Verification is completed
Completed successfully
user@user-System-Product-Name:~/Desktop/javalab$
```

10. Write a Java program that implements a multi-threaded program which has three threads. First thread generates a random integer every 1 second. If the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.

Program

```
import java.util.Random;
class RandomNumberThread extends Thread {
  public void run() {
        Random random = new Random();
        for (int i = 0; i < 10; i++) {
              int randomInteger = random.nextInt(100);
              System.out.println("Random Integer generated : " +
randomInteger);
              if((randomInteger%2) == 0) {
                    SquareThread sThread = new
SquareThread(randomInteger);
                    sThread.start();
              }
              else {
                    CubeThread cThread = new CubeThread(randomInteger);
                    cThread.start();
              }
              try {
                    Thread.sleep(1000);
              }
              catch (InterruptedException ex) {
                    System.out.println(ex);
              }
        }
  }
}

class SquareThread extends Thread {
  int number;

  SquareThread(int randomNumbern) {
        number = randomNumbern;
  }

  public void run() {
        System.out.println("Square of " + number + " = " + (number * number));
  }
}
```

```
class CubeThread extends Thread {
  int number;

  CubeThread(int randomNumber) {
        number = randomNumber;
  }

  public void run() {
        System.out.println("Cube of " + number + " = " + number * number *
number);
  }
}

public class Pgm10 {
  public static void main(String args[]) {
        RandomNumberThread rnThread = new RandomNumberThread();
        rnThread.start();
  }
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Pgm10.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm10
Random Integer generated : 97
Cube of 97 = 912673
Random Integer generated : 42
Square of 42 = 1764
Random Integer generated : 65
Cube of 65 = 274625
Random Integer generated : 85
Cube of 85 = 614125
Random Integer generated : 9
Cube of 9 = 729
Random Integer generated : 44
Square of 44 = 1936
Random Integer generated : 68
Square of 68 = 4624
Random Integer generated : 44
```

Square of 44 = 1936

Random Integer generated : 40

Square of 40 = 1600

Random Integer generated : 99

Cube of 99 = 970299

11. Write a Java program that shows thread synchronization.

Program

```
//example of java synchronized method
class Table{
 synchronized void printTable(int n){//synchronized method
   for(int i=1;i<=5;i++){
     System.out.println(n*i);
     try{
      Thread.sleep(400);
     }catch(Exception e){System.out.println(e);}
   }

 }
}

class MyThread1 extends Thread{
Table t;
MyThread1(Table t){
this.t=t;
}
public void run(){
t.printTable(5);
}
}
class MyThread2 extends Thread{
Table t;
MyThread2(Table t){
this.t=t;
}
public void run(){
t.printTable(100);
}
}
```

```
public class TestSynchronization2{
public static void main(String args[]){
Table obj = new Table();//only one object
MyThread1 t1=new MyThread1(obj);
MyThread2 t2=new MyThread2(obj);
t1.start();
t2.start();
}
}
```

Output

```
5
10
15
20
25
100
200
300
400
500
```

Part E Graphics Programming:

12. Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing.

Program

```
import javax.swing.*;
import java.awt.event.*;
class Calculator extends JFrame implements ActionListener{
        private JTextField t1;
        private JButton b1;
        private JButton b2;
        private JButton b3;
        private JButton b4;
        private JButton b5;
        private JButton b6;
        private JButton b7;
        private JButton b8;
        private JButton b9;
        private JButton b10;
        private JButton b11;
        private JButton b12;
        private JButton b13;
        private JButton b14;
        private JButton b15;
        private JButton b16;
        private Integer res;
        private String operation;
        public Calculator(){
                setLayout(null);
                setSize(640,480);
                t1 = new JTextField();
                t1.setBounds(100,100,200,30);
                b1 = new JButton("1");
                b1.setBounds(100,140,50,30);
                b2 = new JButton("2");
                b2.setBounds(150,140,50,30);
                b3 = new JButton("3");
                b3.setBounds(200,140,50,30);
                b4 = new JButton("+");
                b4.setBounds(250,140,50,30);
                // Third Row
                b5 = new JButton("4");
                b5.setBounds(100,170,50,30);
                b6 = new JButton("5");
                b6.setBounds(150,170,50,30);
```

```
        b7 = new JButton("6");
        b7.setBounds(200,170,50,30);
        b8 = new JButton("-");
        b8.setBounds(250,170,50,30);
        // Fourth Row
        b9 = new JButton("7");
        b9.setBounds(100,200,50,30);
        b10 = new JButton("8");
        b10.setBounds(150,200,50,30);
        b11 = new JButton("9");
        b11.setBounds(200,200,50,30);
        b12 = new JButton("*");
        b12.setBounds(250,200,50,30);
        // Fifth Row
        b13 = new JButton("/");
        b13.setBounds(100,230,50,30);
        b14 = new JButton("%");
        b14.setBounds(150,230,50,30);
        b15 = new JButton("=");
        b15.setBounds(200,230,50,30);
        b16 = new JButton("C");
        b16.setBounds(250,230,50,30);
        add(t1);add(b1);add(b2);
        add(b3);add(b4);add(b5);
        add(b6);add(b7);add(b8);
        add(b9);add(b10);add(b11);
        add(b12);add(b13);add(b14);
        add(b15);add(b16);

        b1.addActionListener(this);b2.addActionListener(this);
        b3.addActionListener(this);b4.addActionListener(this);
        b5.addActionListener(this);b6.addActionListener(this);
        b7.addActionListener(this);b8.addActionListener(this);
        b9.addActionListener(this);b10.addActionListener(this);
        b11.addActionListener(this);b12.addActionListener(this);
        b13.addActionListener(this);b14.addActionListener(this);
        b15.addActionListener(this);b16.addActionListener(this);
}
public void doAction(String op){
        if(operation == null){
                operation = op;
                res = Integer.parseInt(t1.getText());
                t1.setText("");
        }
        else{
                switch(operation){
                        case "+": res = res + Integer.parseInt(t1.getText());
                                        break;
                        case "-": res = res - Integer.parseInt(t1.getText());
```

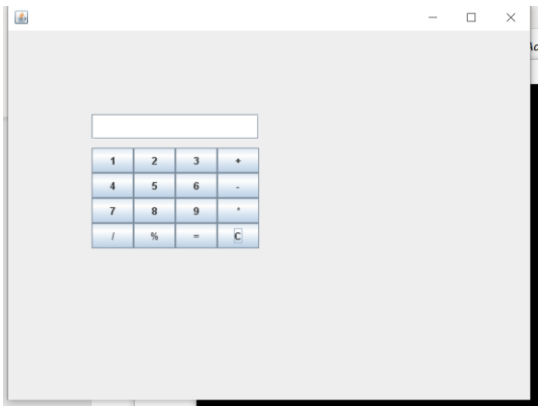
```
                                        break;
                                case "/": try{
                                        if(t1.getText().equals("0"){
                                        throw new ArithmeticException("Divide by Zero");
                                        }
                                                res = res / Integer.parseInt(t1.getText());
                                                 }
                                                 catch(ArithmeticException e){
                                                        t1.setText(e.getMessage());
                                                        operation = null;
                                                        res = 0;
                                                 }
                                                 break;
                                case "*": res = res * Integer.parseInt(t1.getText());
                                                break;
                                case "%": res = res % Integer.parseInt(t1.getText());
                                                break;
                        }
                        if(op.equals("=")){
                                t1.setText(res.toString());
                                res = 0;
                                operation = null;
                        }
                        else{
                                operation = op;
                                t1.setText("");
                        }
                }

        }
        public void actionPerformed(ActionEvent e){
                if(e.getSource()== b1)
                        t1.setText(t1.getText()+"1");
                else if(e.getSource()== b2)
                        t1.setText(t1.getText()+"2");
                else if(e.getSource()==  b3)
                        t1.setText(t1.getText()+"3");
                else if(e.getSource()==  b5)
                        t1.setText(t1.getText()+"4");
                else if(e.getSource()==  b6)
                        t1.setText(t1.getText()+"5");
                else if(e.getSource()==  b7)
                        t1.setText(t1.getText()+"6");
                else if(e.getSource()==  b9)
                        t1.setText(t1.getText()+"7");
                else if(e.getSource()==  b10)
                        t1.setText(t1.getText()+"8");
                else if(e.getSource()==  b11)
                        t1.setText(t1.getText()+"9");
```

```
                else if(e.getSource()==  b16){
                        t1.setText("");
                        res =0;
                        operation = null;
                }
                else if(e.getSource()==  b4){
                        doAction("+");
                }
                else if(e.getSource()==  b8)
                        doAction("-");
                else if(e.getSource()==  b12)
                        doAction("*");
                else if(e.getSource()==  b13)
                        doAction("/");
                else if(e.getSource()==  b14)
                        doAction("%");
                else if(e.getSource()==  b15)
                        doAction("=");

        }
        public static void main(String args[]){
                new Calculator().setVisible(true);
        }

}
```

Output

javac Calculator.java

java Calculator



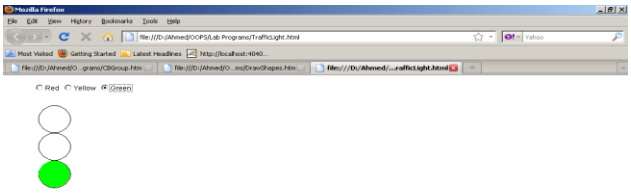
13. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts. **

Program

```
import java.awt.event.*;
import java.applet.*;
import java.awt.*;
/*  <applet code="TrafficLight" width=250 height=200>
 </applet>*/
public class TrafficLight extends Applet implements ItemListener {
 String msg = "";
 Checkbox red,green,yellow;
 CheckboxGroup cbg;
  public void init() {
   cbg = new CheckboxGroup();
   red = new Checkbox("Red", cbg, false);
   green = new Checkbox("Green", cbg, false);
   yellow = new Checkbox("Yellow", cbg, false);
   add(red);
   add(yellow);
   add(green);
    red.addItemListener(this);
   yellow.addItemListener(this);
   green.addItemListener(this);
 }
  public void itemStateChanged(ItemEvent ie) {
   repaint();
 }
  // Display current state of the check boxes.
 public void paint(Graphics g) {
    Color color;
     color=Color.BLACK;
```

```
g.setColor(color);
g.drawOval(50, 50, 52, 52);
g.drawOval(50, 103, 52, 52);
g.drawOval(50, 156, 52, 52);
 String col = cbg.getSelectedCheckbox().getLabel();
System.out.println(col);
 if(col.equalsIgnoreCase("Green"))
{
color= Color.GREEN;
g.setColor(color);
g.fillOval(50, 156, 52, 52);
}
if(col.equalsIgnoreCase("Red"))
{
color=Color.RED;
g.setColor(color);
g.fillOval(51, 51, 51, 51);
  }
if(col.equalsIgnoreCase("Yellow"))
{
color=Color.YELLOW;
g.setColor(color);
g.fillOval(50, 103, 51, 51);
}
 }
}
```

Output



14. Write a Java program to display all records from a table using Java Database Connectivity (JDBC).

ALGORITHM

The following steps are required to create a new Database using JDBC application −

Import the packages: Requires that you include the packages containing the JDBC classes needed for database programming. Most often, using import java.sql.* will suffice.

Register the JDBC driver: Requires that you initialize a driver so you can open a communications channel with the database.

Open a connection: Requires using the DriverManager.getConnection() method to create a Connection object, which represents a physical connection with a database server.

Execute a query: Requires using an object of type Statement for building and submitting an SQL statement to select (i.e. fetch) records from a table.

Extract Data: Once SQL query is executed, you can fetch records from the table.

Clean up the environment: Requires explicitly closing all database resources versus relying on the JVM's garbage collection.

Program

```
//STEP 1. Import required packages
import java.sql.*;

public class JDBCExample {
  // JDBC driver name and database URL
  static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
  static final String DB_URL = "jdbc:mysql://localhost/STUDENTS";
```

```
//  Database credentials
static final String USER = "username";
static final String PASS = "password";

public static void main(String[] args) {
Connection conn = null;
Statement stmt = null;
try{
   //STEP 2: Register JDBC driver
   Class.forName("com.mysql.jdbc.Driver");

   //STEP 3: Open a connection
   System.out.println("Connecting to a selected database...");
   conn = DriverManager.getConnection(DB_URL, USER, PASS);
   System.out.println("Connected database successfully...");

   //STEP 4: Execute a query
   System.out.println("Creating statement...");
   stmt = conn.createStatement();

   String sql = "SELECT id, first, last, age FROM Registration";
   ResultSet rs = stmt.executeQuery(sql);
   //STEP 5: Extract data from result set
   while(rs.next()){
      //Retrieve by column name
      int id  = rs.getInt("id");
      int age = rs.getInt("age");
      String first = rs.getString("first");
      String last = rs.getString("last");

      //Display values
      System.out.print("ID: " + id);
      System.out.print(", Age: " + age);
```

```
        System.out.print(", First: " + first);
        System.out.println(", Last: " + last);
      }
      rs.close();
    }catch(SQLException se){
      //Handle errors for JDBC
      se.printStackTrace();
    }catch(Exception e){
      //Handle errors for Class.forName
      e.printStackTrace();
    }finally{
      //finally block used to close resources
      try{
        if(stmt!=null)
           conn.close();
      }catch(SQLException se){
      }// do nothing
      try{
        if(conn!=null)
           conn.close();
      }catch(SQLException se){
        se.printStackTrace();
      }//end finally try
    }//end try
    System.out.println("Goodbye!");
  }//end main
  }//end JDBCExample
```

Output

```
C:\>java JDBCExample
Connecting to a selected database...
Connected database successfully...
Creating statement...
ID: 100, Age: 18, First: Zara, Last: Ali
ID: 101, Age: 25, First: Mahnaz, Last: Fatma
ID: 102, Age: 30, First: Zaid, Last: Khan
ID: 103, Age: 28, First: Sumit, Last: Mittal
Goodbye!
```

Part F Standard Searching and Sorting Algorithms

15. Write a Java program to Create a doubly linked list

Program

```
class DoublyLinkedList {
   //A node class for doubly linked list
   class Node{
      int item;
      Node previous;
      Node next;
         public Node(int item) {
          this.item = item;
      }
   }
   //Initially, heade and tail is set to null
   Node head, tail = null;

   //add a node to the list
   public void addNode(int item) {
      //Create a new node
      Node newNode = new Node(item);

      //if list is empty, head and tail points to newNode
      if(head == null) {
         head = tail = newNode;
         //head's previous will be null
         head.previous = null;
         //tail's next will be null
         tail.next = null;
      }
      else {
         //add newNode to the end of list. tail->next set to newNode
         tail.next = newNode;
```

```
            //newNode->previous set to tail
            newNode.previous = tail;
            //newNode becomes new tail
            tail = newNode;
            //tail's next point to null
            tail.next = null;
        }
    }

//print all the nodes of doubly linked list
    public void printNodes() {
        //Node current will point to head
        Node current = head;
        if(head == null) {
            System.out.println("Doubly linked list is empty");
            return;
        }
        System.out.println("Nodes of doubly linked list: ");
        while(current != null) {
            //Print each node and then go to next.
            System.out.print(current.item + " ");
            current = current.next;
        }
    }
}
class Main{
    public static void main(String[] args) {
        //create a DoublyLinkedList object
        DoublyLinkedList dl_List = new DoublyLinkedList();
        //Add nodes to the list
        dl_List.addNode(10);
        dl_List.addNode(20);
        dl_List.addNode(30);
```

```
        dl_List.addNode(40);
        dl_List.addNode(50);

        //print the nodes of DoublyLinkedList
        dl_List.printNodes();
    }
}
```

Output

```
Nodes of doubly linked list:
10 20 30 40 50
```

16. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.

Program

```
public class Pgm16 {

  String names[];
int length;
  public static void main(String[] args) {
     Pgm16 obj = new Pgm16();
     String stringsList[] = {"Reeja", "Amrutha", "Teena", "Ramya", "Honey",
   "Keerthy", "Tintu"};
     obj.sort(stringsList);

     for (String i : stringsList) {
        System.out.println(i);
              }
   }
   void sort(String array[]) {
     if (array == null || array.length == 0) {
   return;
     }
     this.names = array;
     this.length = array.length;
     quickSort(0, length - 1);
   }

   void quickSort(int lowerIndex, int higherIndex) {
     int i = lowerIndex;
    int j = higherIndex;
    String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];

     while (i <= j) {
       while (this.names[i].compareToIgnoreCase(pivot) < 0) {
         i++;
       }
       while (this.names[j].compareToIgnoreCase(pivot) > 0) {
         j--;
       }

       if (i <= j) {
          exchangeNames(i, j);
          i++;
          j--;
       }
}
       if (lowerIndex < j) {
```

```
            quickSort(lowerIndex, j);
        }
        if (i < higherIndex) {
            quickSort(i, higherIndex);
        }
    }

  void exchangeNames(int i, int j) {
    String temp = this.names[i];
            this.names[i] = this.names[j];
    this.names[j] = temp;
        }
      }
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Pgm16.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm16
Amrutha Honey Keerthy Ramya Reeja Teena Tintu user@user-System-Product-
Name:~/Desktop/javalab$ javac Pgm16.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm16
Amrutha
Honey
Keerthy
Ramya
Reeja
Teena
Tintu
user@user-System-Product-Name:~/Desktop/javalab$
```

17. Write a Java program that implements the binary search algorithm

Program

```
import java.util.Scanner;
// Binary Search in Java
class Pgm17 {
 int binarySearch(int array[], int element, int low, int high) {

  // Repeat until the pointers low and high meet each other
  while (low <= high) {
   // get index of mid element
   int mid = low + (high - low) / 2;

   // if element to be searched is the mid element
   if (array[mid] == element)
    return mid;
   // if element is less than mid element
   // search only the left side of mid
   if (array[mid] < element)
    low = mid + 1;
   // if element is greater than mid element
   // search only the right side of mid
   else
    high = mid - 1;
  }
  return -1;
 }
 public static void main(String args[]) {
  // create an object of Main class
  Pgm17 obj = new Pgm17();
  // create a sorted array
  int[] array = { 10, 20, 30, 40, 50, 60, 70,80,90,100 };
  int n = array.length;
```

```
    // get input from user for element to be searched
    Scanner input = new Scanner(System.in);
     System.out.println("Array Elements");
for(int i=0;i<10;i++)
System.out.print(array[i]+"\t");
    System.out.println("\nEnter element to be searched:");
    // element to be searched
    int element = input.nextInt();
    input.close();
    // call the binary search method
    // pass arguments: array, element, index of first and last element
    int result = obj.binarySearch(array, element, 0, n - 1);
    if (result == -1)
      System.out.println("Not found");
    else
      System.out.println("Element found at index " + result);
  }
}
```

Output

```
user@user-System-Product-Name:~/Desktop/javalab$ javac Pgm17.java
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm17
Array Elements
10   20      30      40      50      60      70      80      90      100
Enter element to be searched:
40
Element found at index 3
user@user-System-Product-Name:~/Desktop/javalab$ java Pgm17
Array Elements
10   20      30      40      50      60      70      80      90      100
Enter element to be searched:
78
Not found
```