Java Lab Programmes

Lab 1:

Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

```
import java.lang.*;
import java.util.*;
class Lab1
{
   public static void main(String args[]) {
      int i,count=0,n;
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter Any Number : ");
      n=sc.nextInt();

     for(i=1;i<=n;i++) {
        if(n%i==0) {
              count++;
        }
      }

      if(count==2)
        System.out.println(n+" is prime");
      else
        System.out.println(n+" is not prime");
    }
}
```

Output: Enter Any Number : 4
4 is not prime
Enter Any Number : 5
5 is prime

Lab 2:

Write a Java program to demonstrate the OOP principles. [i.e., Encapsulation, Inheritance, Polymorphism and Abstraction]

```
class Person {
   private String name;
   private int age;

  public Person(String name, int age) {
      this.name = name;
      this.age = age;
   }

   public void setName(String name) {
      this.name = name;
   }
```

```
  public String getName() {
        return name;
    }

  public void setAge(int age) {
        this.age = age;
    }

 public int getAge() {
        return age;
    }

   public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

class Employee extends Person {
    private double salary;

    public Employee(String name, int age, double salary) {
        super(name, age);
        this.salary = salary;
    }

  public double getSalary() {
        return salary;
    }

 public void setSalary(double salary) {
        this.salary = salary;
    }

 public void displayInfo() {
        super.displayInfo();
        System.out.println("Salary: " + salary);
    }
}

public class Lab2 {
    public static void main(String[] args) {
        Person person = new Person("Rahul",14);
        System.out.println("Person Info:");
        person.displayInfo();
        System.out.println("======================");
        Employee employee = new Employee("Priyanka", 24, 50000);
        System.out.println("Employee Info:");
        employee.displayInfo();
    }
}
```

Output:

```
Person Info:
Name: Rahul
Age: 14
====================
Employee Info:
Name: Priyanka
Age: 24
Salary: 50000.0
```

Lab 3:

Write a Java program to handle checked and unchecked exceptions. Also, demonstrate the usage of custom exceptions in real time scenario.

```
import java.io.File;
import java.io.FileReader;
import java.io.FileNotFoundException;

class InvalidAgeException extends Exception {
   public InvalidAgeException(String message) {
      super(message);
   }
}

public class Lab3 {
    public static void register(String name, int age) throws InvalidAgeException {
      if (age < 18) {
         throw new InvalidAgeException("User must be at least 18 years old.");
      } else {
         System.out.println("Registration successful for user: " + name);
      }
   }

 public static void main(String[] args) {
      try {
         File file = new File("myfile.txt");
         FileReader fr = new FileReader(file);
      }
     catch (FileNotFoundException e) {
         System.out.println("File not found: " + e.getMessage());
      }

      try {
         int[] arr = {1, 2, 3};
          System.out.println(arr[5]);
      }
       catch (ArrayIndexOutOfBoundsException e) {
         System.out.println("Array index out of bounds: " + e.getMessage());
      }

      finally {
         System.out.println("Cleanup operations can be performed here.");
      }
```

```
        System.out.println("Demonstrating Custom Exception:");

        try {
            register("Rahul", 14);
        }
        catch (InvalidAgeException e) {
            System.out.println("Custom Exception Caught: " + e.getMessage());
        }
    }
}
```

Output:

File not found: myfile.txt (The system cannot find the file specified)
Array index out of bounds: Index 5 out of bounds for length 3
Cleanup operations can be performed here.
Demonstrating Custom Exception:
Custom Exception Caught: User must be at least 18 years old.

Lab 4:

Write a Java program on Random Access File class to perform different read and write operations.

```
import java.io.*;
public class Lab4 {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("data.txt", "rw");
            String data1 = "Hello";
            String data2 = "World";

            file.writeUTF(data1);
            file.writeUTF(data2);
            file.seek(0);

            String readData1 = file.readUTF();
            String readData2 = file.readUTF();
            System.out.println("Data read from file:");
            System.out.println(readData1);
            System.out.println(readData2);

            file.seek(file.length());
            String newData = "Java!";
            file.writeUTF(newData);
            file.seek(0);

            readData1 = file.readUTF();
            readData2 = file.readUTF();
            String readData3 = file.readUTF();

            System.out.println("Data read from file after appending:");
            System.out.println(readData1);
            System.out.println(readData2);
            System.out.println(readData3);
```

```
            file.close();
        } catch (IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Output:

```
Data read from file:
Hello
World
Data read from file after appending:
Hello
World
Java!
```

Lab 5:

Write a Java program to demonstrate the working of different collection classes. [Use package structure to store multiple classes].

ListExample.java

```
import java.util.ArrayList;
public class ListExample {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.add("Orange");
        System.out.println("List Example:");
        for (String fruit : list) {
            System.out.println(fruit);
        }
    }
}
```

SetExample.java

```
import java.util.HashSet;
public class SetExample {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Apple");
        set.add("Banana");
        set.add("Orange");
        set.add("Apple"); // This won't be added since sets don't allow duplicates
        System.out.println("Set Example:");
        for (String fruit : set) {
            System.out.println(fruit);
        }
    }
}
```

MapExample.java

```
import java.util.*;
public class MapExample {
   public static void main(String[] args) {
      HashMap<Integer, String> map = new HashMap<>();
      map.put(1,"Apple");
      map.put(2,"Banana");
      map.put(3,"Orange");
      System.out.println("Map Example:");
      for (Map.Entry<Integer, String> entry : map.entrySet()) {
         System.out.println(entry.getKey()+ ":"+ entry.getValue());
      }
   }
}
```

Lab5.java

```
public class Lab5
{
   public static void main(String[] args) {
      ListExample.main(args);
      SetExample.main(args);
      MapExample.main(args);
   }
}
```

Output:

```
List Example:
Apple
Banana
Orange

Set Example:
Apple
Orange
Banana

Map Example:
1:Apple
2:Banana
3:Orange
```

Lab 6:

Write a program to synchronize the threads acting on the same object. [Consider the example of any reservations like railway, bus, movie ticket booking, etc.]

```
class Reserve extends Thread{
    //Available berths are
    int available = 1;
    int wanted;
```

```
    //accept wanted berths at run time
    public Reserve(int i){
        wanted = i;
    }
     public void run(){
        //Display available berths
        System.out.println("Available berths= "+available);
        //available berths are more than wanted berths
        if(available>=wanted){
            //get the name of the person
            String name = Thread.currentThread().getName();
             //Allot the berth to him
            System.out.println(wanted +" Berth(s) reserved for "+name);
            try{
                Thread.sleep(1500);//wait for printing the ticket
                available = available-wanted;
                //update the no. of available berths
            }catch (InterruptedException ie) {
                ie.printStackTrace();
            }
        }else{
            //if available berths are less, display sorry
            System.out.println("Sorry! No berths available");
        }
    }
}
public class Lab6 {
    public static void main(String[] args) {
        //Tell that one berth is needed
        Reserve obj = new Reserve(2);
        //Attach first thread to the object
        Thread t1 = new Thread(obj);
        //Attach second thread to the same object
        Thread t2 = new Thread(obj);
        //take the thread names as persons names
        t1.setName("First Person");
        t2.setName("Second Person");
        //Send the request for berths
        t1.start();
        t2.start();
    }
}
```

Output:

```
Available berths= 1
Sorry! No berths available
Available berths= 1
Sorry! No berths available
```

Lab 7:

Write a program to perform CRUD operations on the student table in a database using JDBC.
MySQL 8.0

InsertData.java

```
import java.sql.*;

import java.util.Scanner;

public class InsertData {


   public static void main(String[] args) {
      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbcdb","root","password");
         Scanner sc = new Scanner(System.in);
         System.out.println("Inserting Data into student table : ");
         System.out.println("_____");
         System.out.print("Enter student id : ");
         int sid = sc.nextInt();
         System.out.print("Enter student name : ");
         String sname = sc.next();
         String sql="insert into student values("+sid+",'"+sname+"')";
         PreparedStatement ps=con.prepareStatement(sql);
         int r=ps.executeUpdate();
         if(r>0)
         System.out.println("Data inserted successfully into student table");
         sc.close();
         con.close();
      } catch (SQLException err) {
         System.out.println("ERROR: " + err);
      } catch (Exception err) {
         System.out.println("ERROR: " + err);
      }
   }
}
```

Output:

Inserting Data into student table :

_____

Enter student id : 2

Enter student name : ramesh

Data inserted successfully into student table

Inserting Data into student table :

_____

Enter student id : 3

Enter student name : rahul

Data inserted successfully into student table

UpdateData.java

```
import java.sql.*;
import java.util.Scanner;
public class UpdateData {
   public static void main(String[] args) {
      try {
         // to create connection with database
         Class.forName("com.mysql.cj.jdbc.Driver");
         Connection con = DriverManager.getConnection("jdbc:mysql://localhost/jdbcdb", "root", "password");
         Statement s = con.createStatement();
         Scanner sc = new Scanner(System.in);
         System.out.println("Update Data in student table : ");
         System.out.println("_____");
         System.out.print("Enter student Rno to Update: ");
         int sid = sc.nextInt();
         System.out.print("Enter student name : ");
         String sname = sc.next();
         s.execute("update student set name='"+sname+"'where rno = "+sid);
         System.out.println("Data updated successfully");
         s.close();
```

```
            sc.close();
            con.close();
        } catch (SQLException err) {
            System.out.println("ERROR: " + err);
        } catch (Exception err) {
            System.out.println("ERROR: " + err);
        }
    }
}
```

Output:

Update Data in student table :

Enter student Rno to Update: 3

Enter student name : priyanka

Data updated successfully

DeleteData.java

```
import java.sql.*;
import java.util.Scanner;
public class DeleteData {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbcdb", "root",
"password");
            Statement s = con.createStatement();

            Scanner sc = new Scanner(System.in);
            System.out.println("Delete Data from student table : ");
            System.out.println("_____");
            System.out.print("Enter student Rno To Delete: ");
            int sid = sc.nextInt();
            // to execute delete query
            s.execute("delete from student where rno = "+sid);
```

```
            System.out.println("Data deleted successfully");
            s.close();
            sc.close();
            con.close();
        } catch (SQLException err) {
            System.out.println("ERROR: " + err);
        } catch (Exception err) {
            System.out.println("ERROR: " + err);
        }
    }
}
```

Output:

Delete Data from student table :

_____

Enter student Rno To Delete: 2

Data deleted successfully

DisplayData.java

```
import java.sql.*;
public class DisplayData {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost/jdbcdb", "root", "password");
            Statement s = con.createStatement();
           ResultSet rs = s.executeQuery("select * from student");
            if (rs != null) {
            System.out.println("SID \t STU_NAME ");
            System.out.println("_____");
            while (rs.next())
            {
                System.out.println(rs.getInt("rno") +" \t "+ rs.getString("name"));
                System.out.println("_____");
```

```
            }
            s.close();
            con.close();
            }
        } catch (SQLException err) {
            System.out.println("ERROR: " + err);
        } catch (Exception err) {
            System.out.println("ERROR: " + err);
        }
    }
}
```

Output:

SID STU_NAME

---

3 priyanka

Lab 8:

Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Lab8 implements ActionListener
{
int c,n;
String s1,s2,s3,s4,s5;
JFrame f;
JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
JPanel p;
JTextField tf;
GridLayout g;
Lab8()
{
        f = new JFrame("My calculator");
        p = new JPanel();
        tf = new JTextField(20);

        b1 = new JButton("0");
        b2 = new JButton("1");
        b3 = new JButton("2");
        b4 = new JButton("3");
```

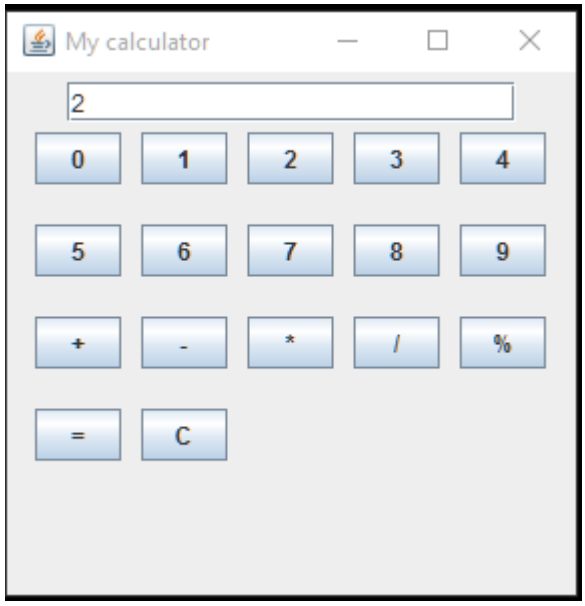
```
        b5 = new JButton("4");
        b6 = new JButton("5");
        b7 = new JButton("6");
        b8 = new JButton("7");
        b9 = new JButton("8");
        b10 = new JButton("9");
        b11 = new JButton("+");
        b12 = new JButton("-");
        b13 = new JButton("*");
        b14 = new JButton("/");
        b15 = new JButton("%");
        b16 = new JButton("=");
        b17 = new JButton("C");

        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);
        b17.addActionListener(this);

        g = new GridLayout(4,4,10,20);

        p.setLayout(g);

        p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.add(b9);
        p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);

        f.add(tf);
        f.add(p);
        f.setSize(300,300);
        f.setVisible(true);
        f.setLayout(new FlowLayout());
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent e)
{
        if(e.getSource()==b1)
        {
                s3 = tf.getText();
                s4 = "0";
                s5 = s3+s4;
                tf.setText(s5);
        }
```

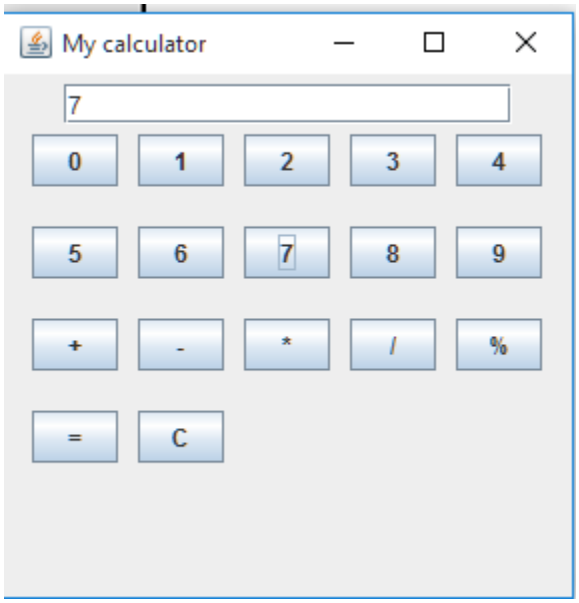
```
if(e.getSource()==b2)
{
        s3 = tf.getText();
        s4 = "1";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b3)
{
        s3 = tf.getText();
        s4 = "2";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b4)
{
        s3 = tf.getText();
        s4 = "3";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b5)
{
        s3 = tf.getText();
        s4 = "4";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b6)
{
        s3 = tf.getText();
        s4 = "5";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b7)
{
        s3 = tf.getText();
        s4 = "6";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b8)
{
        s3 = tf.getText();
        s4 = "7";
        s5 = s3+s4;
        tf.setText(s5);
}
if(e.getSource()==b9)
{
        s3 = tf.getText();
        s4 = "8";
        s5 = s3+s4;
        tf.setText(s5);
```

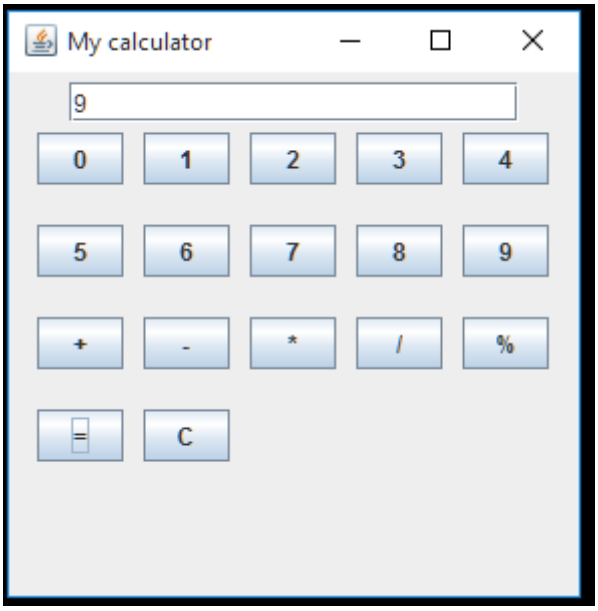
```
}
if(e.getSource()==b10)
{
	s3 = tf.getText();
	s4 = "9";
	s5 = s3+s4;
	tf.setText(s5);
}
if(e.getSource()==b11)
{
	s1 = tf.getText();
	tf.setText("");
	c=1;
}
if(e.getSource()==b12)
{
	s1 = tf.getText();
	tf.setText("");
	c=2;
}
if(e.getSource()==b13)
{
	s1 = tf.getText();
	tf.setText("");
	c=3;
}
if(e.getSource()==b14)
{
	s1 = tf.getText();
	tf.setText("");
	c=4;
}
if(e.getSource()==b15)
{
	s1 = tf.getText();
	tf.setText("");
	c=5;
}
if(e.getSource()==b16)
{
	s2 = tf.getText();
	if(c==1)
	{
		n = Integer.parseInt(s1)+Integer.parseInt(s2);
		tf.setText(String.valueOf(n));
	}
	else if(c==2)
	{
		n = Integer.parseInt(s1)-Integer.parseInt(s2);
		tf.setText(String.valueOf(n));
	}
	else if(c==3)
	{
		n = Integer.parseInt(s1)*Integer.parseInt(s2);
		tf.setText(String.valueOf(n));
```

```
                }
                if(c==4)
                {
                        try
                        {
                                n = Integer.parseInt(s1)/Integer.parseInt(s2);
                                tf.setText(String.valueOf(n));
                        }
                        catch(Exception i)
                        {
                                tf.setText("Arithmetic Exception");
                        }
                }
                if(c==5)
                {
                        n = Integer.parseInt(s1)%Integer.parseInt(s2);
                        tf.setText(String.valueOf(n));
                }
        }
        if(e.getSource()==b17)
        {
                tf.setText("");
        }
}
public static void main(String[] abc)
{
  new Lab8();
}
}
```

Output:







Lab 9

Write a Java program that handles all mouse events and shows the event name at the centre of the window when a mouse event is fired. [Use Adapter classes]

```
import javax.swing.*;

import java.awt.*;

import javax.swing.event.*;

import java.awt.event.*;

class Lab9 extends JFrame implements MouseListener

{

   JLabel l1;

   public Lab9()
```

```
{
   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   setSize(300,300);
   setLayout(new FlowLayout(FlowLayout.CENTER));
   l1 = new JLabel();
   Font f = new Font("Verdana", Font.BOLD, 20);
   l1.setFont(f);
   l1.setForeground(Color.BLUE);
   add(l1);
   addMouseListener(this);
   setVisible(true);
}
public void mouseExited(MouseEvent m)
{
   l1.setText("Mouse Exited");
}
public void mouseEntered(MouseEvent m)
{
   l1.setText("Mouse Entered");
}
public void mouseReleased(MouseEvent m)
{
   l1.setText("Mouse Released");
}
public void mousePressed(MouseEvent m)
{
   l1.setText("Mouse Pressed");
}
public void mouseClicked(MouseEvent m)
{
   l1.setText("Mouse Clicked");
}
public static void main(String[] args) {
```

```
        Lab9 mep = new Lab9();
    }
}
```

Output: