Java Codes By Rajan Sir

1. **Constructor and Methods**

```
class Demo {
    int num;
    Demo() {
        num = 10;

    }
void display()
{ System.out.println("Number: " + num);
    }
public static void main(String[] args) {
Demo obj = new Demo(); obj.display();
    }
}
```

2. **Constructor Overloading**

```
class Demo {
    int num;
    Demo() {
        num = 10;

    }
    Demo(int n) {
        num = n;

    }
    void display() {
        System.out.println("Number: " + num);
    }
    public static void main(String[] args) {
        Demo obj1 = new Demo();
        Demo obj2 = new Demo(20);
        obj1.display();
        obj2.display();}}
```

Java Codes By Rajan Sir

3. **Method Overloading**

```
class Demo {
    void show(int a) {
        System.out.println("Integer: " + a);
    }
    void show(int a, int b) {
        System.out.println("Sum: " + (a + b));
    }
    public static void main(String[] args) {
        Demo obj = new Demo();
        obj.show(10);
        obj.show(10, 20);
    }
}
```

4. **Method Overriding**

```
class Parent {
    void show() {
        System.out.println("Parent class method");
    }
}
class Child extends Parent {
    @Override
    void show() {
        System.out.println("Child class method");
    }
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.show();
```

```
    }
}
```

5. **Command Line Arguments**

```
class CommandLineDemo {
    public static void main(String[] args) {
        if (args.length > 0) {
            System.out.println("Command Line Argument: " + args[0]);
        } else {
            System.out.println("No arguments provided!");
        }
    }
}
```

6. **Logical Operators**

```
class LogicalOperators {
    public static void main(String[] args) {
        boolean a = true, b = false;
        System.out.println("AND (a && b): " + (a && b));
        System.out.println("OR (a || b): " + (a || b));
        System.out.println("NOT (!a): " + (!a));
    }
}
```

7. **Looping Statements**

a) For Loop

```
class ForLoop {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Iteration: " + i);}}}
```

b) While Loop

```
class WhileLoop {
    public static void main(String[] args) {
        int i = 1;
        while (i <= 5) {
            System.out.println("Iteration: " + i);
            i++;
        }
    }
}
```

c) Do-While Loop

```
class DoWhileLoop {
    public static void main(String[] args) {
        int i = 1;
        do {
            System.out.println("Iteration: " + i);
            i++;
        } while (i <= 5);
    }
```

8. **Decision-Making Statements**

If-Else Statement

```
class IfElseDemo {
    public static void main(String[] args) {
        int num = 10;


        if (num > 0) {
            System.out.println("Positive Number");
        } else {

            System.out.println("Negative  Number");
```

```
    }
  }
}
```

Switch Statement

```
class SwitchDemo {
  public static void main(String[] args) {
    int day = 2;
    switch (day) {
      case 1:
        System.out.println("Monday");
        break;
      case 2:
        System.out.println("Tuesday");
        break;
      default:
        System.out.println("Other Day");
    }
  }
}
```

9. Single Inheritance

```
class Parent {
  void display() {
    System.out.println("This is Parent class");
  }
}
class Child extends Parent {

  public static void main(String[] args) {
    Child obj = new Child();
    obj.display();
```

```
    }
}
```

10. Multilevel Inheritance

```
class Grandparent {
    void show() {
        System.out.println("Grandparent class");
    }
}
class Parent extends Grandparent {
    void display() {
        System.out.println("Parent class");
    }
}
class Child extends Parent {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.show();
        obj.display();
    }
}
```

11. Hierarchical Inheritance

```
class Parent {
    void show() {
        System.out.println("Parent class");
    }
}
class Child1 extends Parent {
    void child1Method() {
```

```
        System.out.println("Child1 class");
    }
}
class Child2 extends Parent {
    void child2Method() {
        System.out.println("Child2 class");

    }}
public class Hierarchical {

    public static void main(String[] args) {
        Child1 obj1 = new Child1();
        obj1.show();
        obj1.child1Method();
        Child2 obj2 = new Child2();
        obj2.show();
        obj2.child2Method();
    }}
```

12. Multiple Inheritance (Using Interface)

```
interface A {

    void methodA();

}
interface B {

    void methodB();

}
class C implements A, B {
    public void methodA() {
        System.out.println("Method from Interface A");

    }
    public void methodB() {
```

```
        System.out.println("Method from Interface B");
    }
    public static void main(String[] args) {
        C obj = new C();
        obj.methodA();
        obj.methodB();
    }}
```

13. Hybrid Inheritance (Combination of Multiple and Hierarchical, achieved using Interfaces)

```
interface A {
    void methodA();
}
interface B extends A {
    void methodB();
}
class C implements A {
    public void methodA() {
        System.out.println("Method from Interface A");
    }
}
class D extends C implements B {
    public void methodB() {
        System.out.println("Method from Interface B");
    }
    public static void main(String[] args) {
        D obj = new D();
        obj.methodA();
        obj.methodB();
    }
}
```

14. Try catch and finally

Java Codes By Rajan Sir

```
class TryCatchFinallyDemo {

    public static void main(String[] args) {
        try {
            int a = 10, b = 0;
            int result = a / b;
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
          System.out.println("Exception Caught: Division by zero is not allowed.");
        } finally {
          System.out.println("Finally block executed.");
    }}}
```

15. Creating thread using Thread class

```
class MyThread extends Thread {
    public void run() {
          System.out.println("Thread running using Thread class");
    }
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        t1.start();
    }}
```

16. Creating thread using runnable interface

```
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Thread running using Runnable interface");

    }
    public static void main(String[] args) {
        MyRunnable obj = new MyRunnable();
        Thread t2 = new Thread(obj);
```

```
        t2.start();
    }
}
```

17. this keyword

```
class Demo {
    int num;
    Demo(int num) {
        this.num = num;
    }
    void display() {
        System.out.println("Number: " + this.num);
    }
    public static void main(String[] args) {
        Demo obj = new Demo(10);
        obj.display();
    }
}
```

18. Dot operator

```
class Demo {
    int num = 10;
    void show() {
        System.out.println("Using dot operator: " + num);
    }
    public static void main(String[] args) {
        Demo obj = new Demo();
        obj.show();
    }
```

```
}
```

19. instance of operator

```
class Parent {}
class Child extends Parent {
    public static void main(String[] args) {
        Child obj = new Child();
        System.out.println("Is obj an instance of Child? " + (obj instanceof Child));
        System.out.println("Is obj an instance of Parent? " + (obj instanceof Parent));
    }
}
```

20. Methods of StringBuffer

```
class StringBufferDemo {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Hello");
        sb.append(" World");
         System.out.println(sb);
        sb.insert(5, " Java");
        System.out.println(sb);
        sb.replace(6, 10, "C++");
        System.out.println(sb);
        sb.delete(6, 10);
        System.out.println(sb);
        sb.reverse();
        System.out.println(sb);
    }
}
```

21. Vector Class

Java Codes By Rajan Sir

```
import java.util.Vector;
class VectorDemo {

    public static void main(String[] args) {
        Vector<Integer> vec = new Vector<>();
        vec.add(10);

        vec.add(20);

        vec.add(30);
        System.out.println("Vector elements: " + vec);

    }

}
```

22. Wrapper Classes

```
class WrapperDemo {

    public static void main(String[] args) {
        int num = 10;
        Integer obj = Integer.valueOf(num);
        int val = obj.intValue();
        System.out.println("Boxed: " + obj);
        System.out.println("Unboxed: " + val);
    }

}
```

23. Garbage Collection

```
public class GC {
    protected void finalize() throws Throwable {
        System.out.println("Garbage collector called. Object is deleted.");
    }
    public static void main(String[] args) {
```

```
        GC a = new GC();
        GC b = new GC();
        a = null;
        b = null;
        System.gc();
    }}
```

24. **Java Program to Demonstrate Declaration and Importing of User-Defined Packages.**

Step 1. Declare package

```
// File: MyPackage.java
package myPackage;  // Declaring the package
public class MyClass {
    public void displayMessage() {
        System.out.println("Hello from MyClass in myPackage!");
    }
    }
```

Step 2. Import and Use the User-Defined Package

```
// File: Main.java
import myPackage.MyClass;  // Importing the class from user-defined package
public class Main {
    public static void main(String[] args) {
        MyClass myObject = new MyClass();  // Creating an object of MyClass
        myObject.displayMessage();        // Calling the method
    }
}
```

25. Final variable and method.

```
class A {
    final int SIZE = 100; // Final variable
    final void meth() {   // Final method
        System.out.println("This is a final method.");
```

Java Codes By Rajan Sir

```
        System.out.println("Final variable SIZE = " + SIZE)
 } }
class B extends A {
    // This will cause an error because final methods cannot be overridden
    /*
    void meth() {
        System.out.println("Illegal!");
    }
}
class Test {
    public static void main(String[] args) {
        B obj = new B();
        obj.meth();
    }
}
```

26. Super keyword

```
class Person {
    void message() {
        System.out.println("This is person class");
    }
}
class Student extends Person {
    int mobileno;
     void message() {
        System.out.println("This is student class");
    }
    void display() {
        message();
        super.message();
    }
}
```

```
class Test {
    public static void main(String args[]) {
        Student s = new Student();
        s.display();
    }
}
```

27. Write a simple program to create a GUI application using Frame class.

```
import java.awt.*;
import java.awt.event.*;

public class SimpleFrameExample {
    public static void main(String[] args) {
        Frame f = new Frame("Simple Frame Example");
        Button b = new Button("Click Me");
        b.setBounds(100, 100, 100, 50);
        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Button was clicked!");
            }
        });
        f.add(b);
        f.setLayout(null);
        f.setSize(300, 200);
        f.setVisible(true);
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }
}
```

Java Codes By Rajan Sir

28. Write a program to use class CheckBoxGroup to create Radiobutton

```
import java.awt.*;
public class RadioButtonExample {
    public static void main(String[] args) {
        Frame frame = new Frame("Radio Button Example");
        CheckboxGroup group = new CheckboxGroup();
        Checkbox rb1 = new Checkbox("Option 1", group, true);
        Checkbox rb2 = new Checkbox("Option 2", group, false);
        rb1.setBounds(50, 80, 100, 30);
        rb2.setBounds(50, 120, 100, 30);
        frame.add(rb1);
        frame.add(rb2);
        frame.setSize(200, 200);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

29. Write a program to demonstrate constructors of AWT class TextField and TextArea.

```
import java.awt.*;
public class TextFieldTextAreaExample {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT TextField and TextArea Example");
        TextField tf1 = new TextField();
        TextField tf2 = new TextField(20);
        TextField tf3 = new TextField("Hello, World!");
        tf1.setBounds(50, 50, 200, 30);
        tf2.setBounds(50, 100, 200, 30);
```

```
        tf3.setBounds(50, 150, 200, 30);
        TextArea ta1 = new TextArea();
        TextArea ta2 = new TextArea("Initial Text", 5, 20);
        ta1.setBounds(50, 200, 200, 100);
        ta2.setBounds(50, 320, 200, 100);
        frame.add(tf1);
        frame.add(tf2);
        frame.add(tf3);
        frame.add(ta1);
        frame.add(ta2);
        frame.setSize(300, 500);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

30. Write a program to demonstrate BorderLayout.

```
import java.awt.*;
public class BorderLayoutExample {
    public static void main(String[] args) {
        Frame frame = new Frame("BorderLayout Example");
        frame.setLayout(new BorderLayout());
        frame.add(new Button("North"), BorderLayout.NORTH);
        frame.add(new Button("South"), BorderLayout.SOUTH);
        frame.add(new Button("East"), BorderLayout.EAST);
        frame.add(new Button("West"), BorderLayout.WEST);
        frame.add(new Button("Center"), BorderLayout.CENTER);
        frame.setSize(300, 200);
```

```
        frame.setVisible(true);
        frame.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

31. Write a program to demonstrate GridLayout.

```
import java.awt.*;
public class GridLayoutExample {
    public static void main(String[] args) {
        Frame frame = new Frame("GridLayout Example");
        frame.setLayout(new GridLayout(3, 2));
        frame.add(new Button("Button 1"));
        frame.add(new Button("Button 2"));
        frame.add(new Button("Button 3"));
        frame.add(new Button("Button 4"));
        frame.add(new Button("Button 5"));
        frame.add(new Button("Button 6"));
        frame.setSize(300, 200);
        frame.setVisible(true);
        frame.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

Java Codes By Rajan Sir

32. Write a program to demonstrate swing class JComboBox

```
import javax.swing.*;
import java.awt.*;
public class JComboBoxExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JComboBox Example");
        String[] items = {"Apple", "Banana", "Cherry", "Date", "Grape"};
        JComboBox<String> comboBox = new JComboBox<>(items);
        comboBox.setSelectedIndex(0);
        JLabel label = new JLabel("Selected Item: " +
comboBox.getSelectedItem());
        comboBox.addActionListener(e ->
            label.setText("Selected Item: " + comboBox.getSelectedItem())
        );
        frame.setLayout(new FlowLayout());
        frame.add(comboBox);
        frame.add(label);
        frame.setSize(300, 150);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

33. JRadioButton Example Using Swing

```
import javax.swing.*;
import java.awt.*;
public class JRadioButtonExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JRadioButton Example");
        JRadioButton radioButton1 = new JRadioButton("Option 1");
        JRadioButton radioButton2 = new JRadioButton("Option 2");
        JRadioButton radioButton3 = new JRadioButton("Option 3", true);
       ButtonGroup group = new ButtonGroup();
        group.add(radioButton1);
```

```
        group.add(radioButton2);


        group.add(radioButton3);
        JLabel label = new JLabel("Selected Option: Option 3");
        radioButton1.addActionListener(e -> label.setText("Selected Option:
Option 1"));
        radioButton2.addActionListener(e -> label.setText("Selected Option:
Option 2"));
        radioButton3.addActionListener(e -> label.setText("Selected Option:
Option 3"));
        frame.setLayout(new FlowLayout());
        frame.add(radioButton1);
        frame.add(radioButton2);
        frame.add(radioButton3);
        frame.add(label);
        frame.setSize(300, 150);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

34. Write a Example of use of setToolTipNext() method to set Tooltip text for swing component.

```
import javax.swing.*;
import java.awt.*;
public class ToolTipExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("ToolTip Example");
        JButton button = new JButton("Click Me");
        button.setToolTipText("This button performs an action when clicked.");
        JLabel label = new JLabel("Hover over the button to see the tooltip.");
        frame.setLayout(new FlowLayout());
        frame.add(label);
```

```
        frame.add(button);
        frame.setSize(300, 150);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

35. Write a program to design a form using the components List and Choice.

```
import java.awt.*;
public class Practical2 {
    public static void main(String[] args) {
        Frame f = new Frame("Form with List and Choice Components");
        f.setSize(400, 400);
        f.setLayout(null);
        f.setVisible(true);
        Label c = new Label("Select Country");
        Choice c1 = new Choice();
        c1.add("India");
        c1.add("America");
        c1.add("England");
        c1.add("Russia");
        Label w = new Label("Weekdays");
        List l = new List(7, true);
        l.add("Sunday");
        l.add("Monday");
        l.add("Tuesday");
        l.add("Wednesday");
        l.add("Thursday");
        l.add("Friday");
        l.add("Saturday");
        c.setBounds(100, 100, 100, 20);
        c1.setBounds(100, 130, 100, 100);
        w.setBounds(100, 200, 100, 20);
        l.setBounds(100, 250, 100, 100);
```

```
        f.add(c);
        f.add(c1);
        f.add(w);
        f.add(l);
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }
}
```

36. Write a program using AWT to create a menubar where menubar contains menu items such as File, Edit, View and create a submenu under the File menu: New and Open.

```
import java.awt.*;
public class practical5 extends Frame {
    MenuBar mb;
    Menu m,m1,m2;
    MenuItem mi,mi2;
    practical5()
    {
        Frame f = new Frame();
        setVisible(true);
        setSize(400,400);
        setLayout(null);
        MenuBar mb =new MenuBar();
        setMenuBar(mb);
        Menu m = new Menu("File");
        Menu m1 = new Menu("Edit");
        Menu m2 = new Menu("View");
        MenuItem mi = new MenuItem("New");
        MenuItem mi2 = new MenuItem("Open");
        mb.add(m);
```

```
        mb.add(m1);
        mb.add(m2);
        m.add(mi);
        m.add(mi2);
    }
    public static void main(String args[])
    {
        practical5 p = new practical5();
        p.setVisible(true);
    }
}
```

37. Write a program to create a JTable.

```
import java.awt.*;
import javax.swing.*;
public class JTableExample {
    JFrame f;
    JTableExample()
    {
        String rows[][]={   {"Dnyaneshwari","2001160374","IF5I"},
                            {"Hariprasad","2001160372","IF5I"},
                            {"Soham","2001160376","IF5I"},
                            {"Vaishnavi","2001160368","IF5I"}
                        };
        String col[]={"NAME","ENROLLMENT NO","DEPARTMENT"};
        JTable jt = new JTable(rows,col);
        jt.setBounds(30,40,200,300);
        JScrollPane sp = new JScrollPane(jt);
        JFrame f = new JFrame();
        f.add(sp);
        f.setSize(300,400);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
```

```
        new JTableExample();
    }
}
```

38. Write a program to demonstrate the use of JTextField and JPasswordField using Listener Interface.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Pra12_1 extends JFrame implements ActionListener {
    JTextField t;
    JPasswordField p;
    JButton b;
    public Pra12_1() {
        JLabel l1 = new JLabel("Username:");
        JLabel l2 = new JLabel("Password:");
        t = new JTextField(20);
        p = new JPasswordField(20);
        p.setEchoChar('#');
        b = new JButton("Login");
        b.addActionListener(this);
        setLayout(new FlowLayout());
        add(l1);
        add(t);
        add(l2);
        add(p);
        add(b);
        setTitle("Login Form");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        String username = t.getText();
```

```
        String password = new String(p.getPassword());
        if (username.equals("admin") && password.equals("1234")) {
            JOptionPane.showMessageDialog(this, "Login Successful!");
        } else {
            JOptionPane.showMessageDialog(this, "Invalid Credentials!");
        }
    }
    public static void main(String[] args) {
        new Pra12_1();
    }
}
```

39. Write a program to demonstrate the use of AWT components like Label, Textfield, TextArea, Button, Checkbox, RadioButton etc.

```
import java.awt.*;
import java.awt.event.*;
public class Practical1 {
    public static void main(String args[]) {
        Frame f = new Frame("AWT Components Demo");
        f.setSize(500, 600);
        f.setLayout(null);
        Label n = new Label("Enter your Name:");
        TextField t = new TextField();
        Label a = new Label("Enter Your Address:");
        TextArea ta = new TextArea();
        Label g = new Label("Select Gender:");
        CheckboxGroup cbg = new CheckboxGroup();
        Checkbox c1 = new Checkbox("Male", cbg, true);
        Checkbox c2 = new Checkbox("Female", cbg, false);
        Label s = new Label("Select Subject:");
        Checkbox c3 = new Checkbox("AJP");
        Checkbox c4 = new Checkbox("CSS");
        Checkbox c5 = new Checkbox("OSY");
```

```
        Button b = new Button("Submit");
        n.setBounds(100, 50, 150, 20);
        t.setBounds(100, 75, 250, 25);
        a.setBounds(100, 110, 150, 20);
        ta.setBounds(100, 135, 250, 60);
        g.setBounds(100, 210, 150, 20);
        c1.setBounds(100, 235, 100, 20);
        c2.setBounds(200, 235, 100, 20);
        s.setBounds(100, 270, 150, 20);
        c3.setBounds(100, 295, 100, 20);
        c4.setBounds(100, 320, 100, 20);
        c5.setBounds(100, 345, 100, 20);
        b.setBounds(100, 390, 100, 30);
        f.add(n); f.add(t);
        f.add(a); f.add(ta);
        f.add(g); f.add(c1); f.add(c2);
        f.add(s); f.add(c3); f.add(c4); f.add(c5);
        f.add(b);
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                f.dispose();
            }
        });
        f.setVisible(true);
    }
}
```

40. Write a program to demonstrate the use of InetAddress class and its factory methods.

```
package practicals;
import java.net.*;
import java.io.*;
public class inet_address {
    public static void main(String[] args) {
```

```
        try {
            InetAddress ip = InetAddress.getByName("www.google.com");
            System.out.println("Host Name:"+ip.getHostName());
            System.out.println("IP Address:"+ip.getHostAddress());
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

41. Write a program for use of URL class method.

```
import java.net.*;
public class URLExample {
    public static void main(String[] args) throws MalformedURLException {
        URL url = new URL("https://www.example.com:8080/index.html");
        System.out.println("Host: " + url.getHost());
        System.out.println("Protocol: " + url.getProtocol());
        System.out.println("Path: " + url.getPath
        System.out.println("Port: " + url.getPort());
    }
}
```

42. Write a program using URL class to retrieve the host,protocol,port and file of URL http://www.msbte.org.in

```
import java.net.*;
public class URLDetails {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.msbte.org.in");
            System.out.println("Protocol: " + url.getProtocol());
            System.out.println("Host: " + url.getHost());
            System.out.println("Port: " + url.getPort());
            System.out.println("File: " + url.getFile());
```

```
        } catch (MalformedURLException e) {
            System.out.println("Invalid URL: " + e.getMessage());
        }
    }
}
```

43. Develop a program using InetAddress class to retrieve IP address of computer when hostname is entered by the user.

```
import java.net.*;
import java.util.Scanner;
public class HostnameToIP {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter hostname: ");
        String hostname = scanner.nextLine();
        try {
            InetAddress address = InetAddress.getByName(hostname);
            System.out.println("IP Address: " + address.getHostAddress());
        } catch (UnknownHostException e) {
            System.out.println("Host not found: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

44. Write a program for URL Class with constructor.

```
import java.io.IOException;
import java.net.URL;
public class URLExample {
    public static void main(String[] args) {
        try {
            URL url = new
```

```
URL("https://www.tutorialspoint.com/index.htm?language=en#j2se");
            System.out.println("URL: " + url.toString());
            System.out.println("Protocol: " + url.getProtocol());
            System.out.println("Authority: " + url.getAuthority());
            System.out.println("File: " + url.getFile());
            System.out.println("Host: " + url.getHost());
            System.out.println("Path: " + url.getPath());
            System.out.println("Port: " + url.getPort());
            System.out.println("Default Port: " + url.getDefaultPort());
            System.out.println("Query: " + url.getQuery());
            System.out.println("Reference (Fragment): " + url.getRef());
         } catch (IOException e) {
            e.printStackTrace();
         }
      }
   }
```

45. Write a program for URL connection class method.

```
import java.io.*;
import java.net.*;
public class URLConnectionExample {
   public static void main(String[] args) {
      try {
         URL url = new URL("https://www.v2vclass.com");
         URLConnection urlcon = url.openConnection();
         InputStream stream = urlcon.getInputStream();
         int i;
         while ((i = stream.read()) != -1) {
            System.out.print((char) i);
         }
      } catch (Exception e) {
         System.out.println("Error: " + e);
      }
} }
```

46. Write a program using Socket and ServerSocket to create an chat application.

It consists of two programs:

1. Server Program (Receives and sends messages)

2. Client Program (Connects to the server and communicates)

Server program:

```
import java.io.*;
import java.net.*;
public class ChatServer {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(12345);
            System.out.println("Server started. Waiting for client...");
            Socket socket = serverSocket.accept();
            System.out.println("Client connected!");
            BufferedReader input = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter output = new PrintWriter(socket.getOutputStream(),
true);
            BufferedReader consoleInput = new BufferedReader(new
InputStreamReader(System.in));
            String message;
            while (true) {
                message = input.readLine();
                if (message.equalsIgnoreCase("exit")) break;
                System.out.println("Client: " + message);
                System.out.print("You: ");
                output.println(consoleInput.readLine());
            }
            socket.close();
            serverSocket.close();
```

Java Codes By Rajan Sir

```
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client program:

```
import java.io.*;
import java.net.*;

public class ChatClient {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 12345);
            System.out.println("Connected to server!");
            BufferedReader input = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter output = new PrintWriter(socket.getOutputStream(),
true);
            BufferedReader consoleInput = new BufferedReader(new
InputStreamReader(System.in));
            String message;
            while (true) {
                System.out.print("You: ");
                message = consoleInput.readLine();
                output.println(message);
                if (message.equalsIgnoreCase("exit")) break;
                System.out.println("Server: " + input.readLine());
            }
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    } }
```

Java Codes By Rajan Sir

47. Write a program to insert a record in table.

```
import java.io.*;
import java.net.*;
public class ChatClient {
   public static void main(String[] args) {
      try {
         Socket socket = new Socket("localhost", 12345);
         System.out.println("Connected to server!");
         BufferedReader input = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
         PrintWriter output = new PrintWriter(socket.getOutputStream(),
true);
         BufferedReader consoleInput = new BufferedReader(new
InputStreamReader(System.in));
         String message;
         while (true) {
            System.out.print("You: ");
            message = consoleInput.readLine();
            output.println(message);
            if (message.equalsIgnoreCase("exit")) break;
            System.out.println("Server: " + input.readLine());
         }
         socket.close();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

48. Write a program to update record in a database.

```
import java.sql.*;
public class UpdateData {
   public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/student";
        String user = "root";
        String password = "";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user,
password);
            System.out.println("Connected to database!");
            String updateSQL = "UPDATE login SET emailid = ? WHERE
emailid = ?";
            PreparedStatement pstmt = conn.prepareStatement(updateSQL);
            pstmt.setString(1, "rajan@accunityservices.com");
            pstmt.setString(2, "rajan@gmail.com");
            int rowsUpdated = pstmt.executeUpdate();
            if (rowsUpdated > 0) {
                System.out.println("Student record updated successfully!");
            } else {
                System.out.println("No matching record found.");
            }
            pstmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

49. Write a program to implement delete statement.

```
import java.sql.*;
public class DeleteData {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/student";
        String user = "root";
        String password = "";
```

```
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user,
password);
            String deleteSQL = "DELETE FROM login WHERE username = ?";
            PreparedStatement pstmt = conn.prepareStatement(deleteSQL);
            pstmt.setString(1, "John Doe");
            int rowsDeleted = pstmt.executeUpdate();
            System.out.println(rowsDeleted > 0 ? "Record deleted successfully!"
: "No matching record found.");
            pstmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

50. Develop JDBC program to retrieve data from table using ResultSet interface.

```
import java.sql.*;
public class RetrieveStudentRecords {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/school"; // Replace 'school'
with your database name
        String user = "root";
        String password = "";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user,
password);
            String selectSQL = "SELECT * FROM student";
            Statement stmt = conn.createStatement();
```

Java Codes By Rajan Sir

```
            ResultSet rs = stmt.executeQuery(selectSQL);
            System.out.println("Student Records:");
            while (rs.next()) {
                int id = rs.getInt("id");
                String name = rs.getString("name");
                int age = rs.getInt("age");
                String grade = rs.getString("grade");
                System.out.println(id + " | " + name + " | " + age + " | " + grade);
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Java Codes By Rajan Sir

```
// Set frame properties
```