

INCOMPLETE DOCUMENT

CODE & TITLES are correct
rest are unfinished, some progs have
their outputs in the code

Rexiel Scarlet

2/10/2024

Contents

1	Fibonacci Series	4
1.1	Aim	4
1.2	Code	4
1.3	Output	4
2	Operations On Complex Numbers	5
2.1	Aim	5
2.2	Code	5
2.3	Output	6
3	Package Implementation	7
4	Class Implementation	8
4.1	Aim	8
4.2	Code	8
4.3	Output	9
5	Abstract Class Implementation	10
5.1	Aim	10
5.2	Code	10
5.3	Output	11
6	Inheritance In Java	12
6.1	Aim	12
6.2	Code	12
6.3	Output	13
7	Abstract Class With Inheritance	14
7.1	Aim	14
7.2	Code	14
7.3	Output	14
8	Interface Implementation	15
8.1	Aim	15
8.2	Code	15
8.3	Output	16
9	Multithreading In Java	17
9.1	Aim	17
9.2	Code	17
9.3	Output	18
10	Exception Handling	19
10.1	Aim	19
10.2	Code	19
10.3	Output	19
11	Validation using Swing	20
11.1	Aim	20
11.2	Code	20
11.3	Output	21
12	Text Field in Swing	22
12.1	Aim	22
12.2	Code	22
12.3	Output	23

13 Traffic Light Simulation	24
13.1 Aim	24
13.2 Code	24
13.3 Output	25
14 Applet In Java	26
14.1 Aim	26
14.2 Code	26
14.3 Output	26
15 Savings Account	27
15.1 Aim	27
15.2 Code	27
15.3 Output	27
16 Integer Divisions	28
16.1 Aim	28
16.2 Code	28
16.3 Output	29
17 Simple Interest	30
17.1 Aim	30
17.2 Code	30
17.3 Output	30
18 Mouse Coordinates	31
18.1 Aim	31
18.2 Code	31
18.3 Output	31
19 Simple Banner	32
19.1 Aim	32
19.2 Code	32
19.3 Output	32
20 Grid Layout Manager	33
20.1 Aim	33
20.2 Code	33
20.3 Output	33
21 Priority Threads	34
21.1 Aim	34
21.2 Code	34
21.3 Output	34
22 Employee Details	35
22.1 Aim	35
22.2 Code	35
22.3 Output	35
23 Update In JDBC	36
23.1 Aim	36
23.2 Code	36
23.3 Output	37
24 Product Details	38
24.1 Aim	38
24.2 Code	38
24.3 Output	39

1 Fibonacci Series

1.1 Aim

- Write a program to print the first n terms of the Fibonacci series

1.2 Code

```
package org.projects.prog1;

import java.util.Scanner;

public class Fibo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of terms");
        int limit = sc.nextInt();

        for (int i = 0; i < limit; i++) {
            System.out.print(fibo(i) + " ");
        }
    }

    static int fibo(int num) {
        return (num <= 1) ? num : fibo(num - 1) + fibo(num - 2);
    }
}
```

1.3 Output

2 Operations On Complex Numbers

2.1 Aim

- Write a program to create a class Complex to store complex numbers
- Complex has two fields to store the real and imaginary parts
- create static functions to add and multiply two complex numbers
- create a member function to display the complex number

2.2 Code

```
package org.projects.prog2;

public class Complex {
    int real, img;

    Complex(int real, int img) {
        this.real = real;
        this.img = img;
    }

    void display() {
        // some formatting to correctly display complex numbers
        String fmt =
            String.format("%d %c %di", real, (img > 0) ? '+' : '-', (img > 0) ? img : (-1
                ↪ * img));

        System.out.println(fmt);
    }

    static Complex add(Complex n1, Complex n2) {
        return new Complex(n1.real + n2.real, n1.img + n2.img);
    }

    static Complex multiply(Complex n1, Complex n2) {
        // alg
        // (a + bi) * (c + di)
        // = ac + adi + bci - bd =
        // = (ac - bd) + (ad + bc)i
        int real = n1.real * n2.real - (n1.img * n2.img);
        int img = (n1.real * n2.img) + (n1.img * n2.real);
        return new Complex(real, img);
    }

    static void cmp(Complex n1, Complex n2) {
        if (n1.real == n2.real && n1.img == n2.img) {
            System.out.println("The complex numbers are equal");
        } else {
            System.out.println("The complex numbers are not equal");
        }
    }

    public static void main(String args[]) {
        Complex c1 = new Complex(2, 3),
            c2 = new Complex(1, -2),
            c3 = add(c1, c2),
```

```
        c4 = multiply(c1, c2);

    c1.display();
    c2.display();
    c3.display();
    c4.display();

    cmp(c1, c2);
    cmp(c1, c1);
}
}
```

2.3 Output

3 Package Implementation

4 Class Implementation

4.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

4.2 Code

```
package org.projects.prog4;

public class Solid {
    int length, breadth, height;

    public Solid(int side) {
        length = side;
        breadth = side;
        height = side;
    }

    public Solid(int side, int height) {
        length = side;
        breadth = side;
        this.height = height;
    }

    public Solid(int length, int breadth, int height) {
        // its really easy to misspell length/breadth/height and get into odd errors
        ⇨ ideally during
        // exams I recommend shortening these to just l, b, h;
        this.length = length;
        this.breadth = breadth;
        this.height = height;
    }

    int area() {
        return 2 * ((length * breadth) + (breadth * height) + (height * length));
    }

    int volume() {
        return length * breadth * height;
    }

    void display() {
        System.out.println(String.format("Area: %d\nVolume: %d", area(), volume()));
    }

    public static void main(String args[]) {
        Solid cube = new Solid(2), sPrism = new Solid(2, 3), cuboid = new Solid(2, 3, 4);

        cube.display();
        sPrism.display();
        cuboid.display();
    }
}

// OUTPUT
```



```
/*  
Area: 24  
Volume: 8  
Area: 32  
Volume: 12  
Area: 52  
Volume: 24  
*/
```

4.3 Output

5 Abstract Class Implementation

5.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

5.2 Code

```
package org.projects.prog5;

abstract class Figure {
    int length, breadth, height;

    abstract int area();

    void display() {
        // similiar to printf in C <stdio.h>
        System.out.printf(
            "Length: %d, Breadth: %d, Height: %d\nArea: %d\n", length, breadth, height,
            ↪ area());
    }
}

class Rectangle extends Figure {
    public Rectangle(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
        height = 0;
    }

    int area() {
        return length * breadth;
    }
}

class Triangle extends Figure {
    public Triangle(int breadth, int height) {
        this.breadth = breadth;
        this.height = height;
        length = 0;
    }

    int area() {
        return (breadth * height) / 2;
    }
}

public class Main {
    public static void main(String args[]) {
        Rectangle rec = new Rectangle(2, 3);
        Triangle tri = new Triangle(2, 4);

        rec.display();
        tri.display();
    }
}
```

```
// OUTPUT
/*
Length: 2, Breadth: 3, Height: 0
Area: 6
Length: 0, Breadth: 2, Height: 4
Area: 4
*/
```

5.3 Output

6 Inheritance In Java

6.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

6.2 Code

```
package org.projects.prog6;

class Rectangle {
    int length, breadth;

    public Rectangle(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public int area() {
        return length * breadth;
    }
}

public class Box extends Rectangle {
    int height;

    public Box(int length, int breadth, int height) {
        super(length, breadth);
        this.height = height;
    }

    public int area() {
        return 2 * ((length * breadth) + (breadth * height) + (height * length));
    }

    public int volume() {
        return length * breadth * height;
    }

    public static void main(String args[]) {
        Rectangle rec = new Rectangle(2, 3);
        System.out.println("Area of Rectangle: " + rec.area());

        Box box = new Box(2, 3, 2);
        System.out.printf("Area of Box: %d\nVolume of Box: %d\n", box.area(),
            ↵ box.volume());
    }
}

// OUTPUT
/*
Area of Rectangle: 6
Area of Box: 32
Volume of Box: 12
*/
```

6.3 Output

7 Abstract Class With Inheritance

7.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

7.2 Code

```
package org.projects.prog7;

// this program does not require a main function
// since abstract classes can not be instantiated
public abstract class Account {
    String holder_name;
    long number;
    float balance;

    Account(String name, long num) {
        balance = 0;
        holder_name = name;
        number = num;
    }

    public void deposit(int amount) {
        balance += amount;
    }

    abstract void withdraw(int amount);
}
```

7.3 Output

8 Interface Implementation

8.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

8.2 Code

```
package org.projects.prog8;

class Employee {
    String name, code;
    public int basicPay;

    public Employee(String name, String code, int basicPay) {
        this.name = name;
        this.code = code;
        this.basicPay = basicPay;
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Code: " + code);
        System.out.println("Basic Pay: " + basicPay);
    }
}

interface Salary {
    public abstract int calculate();
}

public class SallarySlip extends Employee implements Salary {
    int allowances, bonuses, deductions;

    public SallarySlip(
        String name, String code, int basicPay, int allowances, int bonuses, int
        ↪ deductions) {
        super(name, code, basicPay);
        this.allowances = allowances;
        this.bonuses = bonuses;
        this.deductions = deductions;
    }

    public int calculate() {
        return basicPay + allowances + bonuses - deductions;
    }

    public void display() {
        System.out.printf("Employee: %s\nCode: %s\nFinal Sallary: %d\n", name, code,
        ↪ calculate());
    }

    public static void main(String args[]) {
        SallarySlip slip = new SallarySlip("Bob Rose", "001", 10000, 4000, 3000, 1500);
        slip.display();
    }
}
```

```
}
```

```
// OUTPUT
```

```
/*
```

```
Employee: Bob Rose
```

```
Code: 001
```

```
Final Sallary: 15500
```

```
*/
```

8.3 Output

9 Multithreading In Java

9.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

9.2 Code

```
package org.projects.prog9;

public class Multithread {

    public static boolean even = false;
    public static boolean odd = false;
    public static boolean terminate = false;

    public static int random = 0;

    public static Runnable threadMain =
        new Runnable() {
            public void run() {
                int i = 1, max = 4;
                while (i++ < max) {
                    try {
                        Thread.sleep(1000);
                    } catch (Exception e) {
                    }
                    random = (int) ((Math.random() * 18) % 10);
                    if (random % 2 == 0) {
                        even = true;
                    } else {
                        odd = true;
                    }
                    try {
                        notify(); // Notify the waiting threads
                    } catch (Exception e) {
                    }
                }
                terminate = true;
                try {
                    notify();
                } catch (Exception e) {
                }
            }
        };

    public static Runnable threadEven =
        new Runnable() {
            public void run() {
                while (!terminate) {
                    while (!even && !terminate) {
                        try {
                            wait(); // Wait until the even flag is set
                        } catch (Exception e) {
                        }
                    }
                }
            }
        }
```

```

        if (!terminate) {
            System.out.println("Square: " + random * random);
            even = false;
            try {
                notify(); // Notify the main thread
            } catch (Exception e) {
            }
        }
    }
}

};

public static Runnable threadOdd =
    new Runnable() {
        public void run() {
            while (!terminate) {
                while (!odd && !terminate) {
                    try {
                        wait(); // Wait until the odd flag is set
                    } catch (Exception e) {
                    }
                }
                if (!terminate) {
                    System.out.println("Cube: " + random * random * random);
                    odd = false;
                    try {
                        notify(); // Notify the main thread
                    } catch (Exception e) {
                    }
                }
            }
        }
    };

public static void main(String[] args) {
    // prolly the most complex program of the rooster ideally wrap everything
    // inside a try catch block and hope everything works :)
    Thread t1 = new Thread(threadMain);
    Thread t2 = new Thread(threadEven);
    Thread t3 = new Thread(threadOdd);

    t1.start();
    t2.start();
    t3.start();
}
}

```

9.3 Output

10 Exception Handling

10.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

10.2 Code

```
package org.projects.prog10;

public class Errors {
    static void division() throws InterruptedException {
        Thread.sleep(100);
        throw new ArithmeticException("division by zero");
    }

    public static void main(String args[]) {
        try {
            division();
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        } catch (ArithmeticException e) {
            System.out.println(e.getMessage());
        } finally {
            System.out.println("Hello Mars!");
        }
    }
}

// output
/*
division by zero
Hello Mars!
*/
```

10.3 Output

11 Validation using Swing

11.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

11.2 Code

```
package org.projects.prog11;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Swing {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Swing Example");
        frame.setLayout(new GridLayout(3, 2));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 130);

        JTextField phoneNum = new JTextField();
        frame.add(new JLabel("Phone"));
        frame.add(phoneNum);

        JTextField email = new JTextField();
        frame.add(new JLabel("Email"));
        frame.add(email);

        JButton submit = new JButton("Submit");
        JLabel result = new JLabel();
        result.setHorizontalAlignment(JLabel.CENTER);

        submit.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String em = email.getText();
                    String phone = phoneNum.getText();
                    result.setText("invalid");

                    if (em.isEmpty() || phone.isEmpty()) {
                        return;
                    }

                    try {
                        Long.parseLong(phone);
                        if (phone.length() < 10) {
                            return;
                        }
                    } catch (Exception error) {
                        return;
                    }

                    if (!em.contains("@")) {
                        return;
                    }
                }
            }
        );
    }
}
```

```
        }  
        result.setText("Valid");  
    }  
});  
  
frame.add(submit);  
frame.add(result);  
frame.setVisible(true);  
}  
}
```

11.3 Output

12 Text Field in Swing

12.1 Aim

- Create a Swing program that accepts a number
- if the number is even add it to list1
- else add it to list 2

12.2 Code

```
package org.projects.prog12;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Swing {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Swing Example");
        frame.setLayout(new GridLayout(4, 2));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 130);

        JTextField number = new JTextField();
        frame.add(new JLabel("Number: "));
        frame.add(number);

        JLabel list1 = new JLabel();
        list1.setHorizontalAlignment(JLabel.CENTER);
        frame.add(new JLabel("List 1"));
        frame.add(list1);

        JLabel list2 = new JLabel();
        list2.setHorizontalAlignment(JLabel.CENTER);
        frame.add(new JLabel("List 2"));
        frame.add(list2);

        JButton submit = new JButton("Submit");

        submit.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) throws NumberFormatException {
                    int num = Integer.parseInt(number.getText());
                    number.setText("");
                    if (num % 2 == 0) {
                        list1.setText(list1.getText() + " " + num);
                    } else {
                        list2.setText(list2.getText() + " " + num);
                    }
                }
            }
        );

        frame.add(submit);
        frame.setVisible(true);
    }
}
```

}

12.3 Output

13 Traffic Light Simulation

13.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

13.2 Code

```
package org.projects.prog13;

import java.awt.Color;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Traffic {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Traffic");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new GridLayout(2, 1));
        frame.setSize(300, 130);

        JLabel sign = new JLabel();
        sign.setHorizontalAlignment(JLabel.CENTER);
        frame.add(sign);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(1, 3));

        JButton red = new JButton("STOP");
        red.setBackground(Color.RED);
        panel.add(red);

        JButton yellow = new JButton("READY");
        yellow.setBackground(Color.YELLOW);
        panel.add(yellow);

        JButton green = new JButton("GO");
        green.setBackground(Color.GREEN);
        panel.add(green);

        ActionListener on_click =
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    sign.setText(((JButton) e.getSource()).getText());
                }
            };

        red.addActionListener(on_click);
        yellow.addActionListener(on_click);
        green.addActionListener(on_click);

        frame.add(panel);
        frame.setVisible(true);
    }
}
```


}

13.3 Output

14 Applet In Java

14.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

14.2 Code

```
package org.projects.prog14;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Tables extends Applet implements ActionListener {
    TextField t1;
    Button t2;
    String[] result;

    public void init() {
        t1 = new TextField(5);
        t2 = new Button("Submit");
        t2.addActionListener(this);

        result = new String[10];

        add(new Label("Enter a number"));
        add(t1);
        add(t2);
    }

    public void paint(Graphics g) {
        for (int i = 0; i < 10; i++) {
            g.drawString(result[i], 20, 40 + 20 * i);
        }
    }

    public void actionPerformed(ActionEvent e) {
        int number = Integer.parseInt(t1.getText());
        for (int i = 1; i <= 10; i++) {
            result[i - 1] = number + " x " + i + " = " + i * number;
        }
    }
}
```

14.3 Output

15 Savings Account

15.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

15.2 Code

```
package org.projects.prog15;

class Account {
    public String accName;
    public long accNumber;

    Account(String name, long number) {
        accName = name;
        accNumber = number;
    }
}

public class SavingsAccount extends Account {
    private int accBalance;

    SavingsAccount(String name, long number) {
        super(name, number);
        accBalance = 0;
    }

    public void display() {
        System.out.println("Account: " + accName);
        System.out.println("No: " + accNumber);
        System.out.println("Balance: " + accBalance);
    }

    public void deposit(int ammount) {
        accBalance += ammount;
    }

    public static void main(String args[]) {
        SavingsAccount account = new SavingsAccount("BiJo", 1239812398);
        account.deposit(1000);
        account.display();
    }
}
```

15.3 Output

16 Integer Divisions

16.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

16.2 Code

```
package org.projects.prog16;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Dialog {
    public static void main(String args[]) {
        JFrame frame = new JFrame("cool frame");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setLayout(new GridLayout(2, 1));

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2, 2));

        panel.add(new JLabel("Num 1: "));
        JTextField num1 = new JTextField();
        panel.add(num1);

        panel.add(new JLabel("Num 2: "));
        JTextField num2 = new JTextField();
        panel.add(num2);

        frame.add(panel);

        JButton submit = new JButton("Submit");
        submit.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    JDialog dialog = new JDialog();
                    dialog.setSize(300, 50);

                    JLabel prompt = new JLabel();
                    prompt.setHorizontalAlignment(JLabel.CENTER);
                    try {
                        int n1 = Integer.parseInt(num1.getText());
                        int n2 = Integer.parseInt(num2.getText());
                        int result = n1 / n2;

                        prompt.setText(n1 + "/" + n2 + " = " + result);
                    } catch (NumberFormatException _e) {
                        prompt.setText("Invalid Number");
                    } catch (ArithmeticException _e) {
                        prompt.setText("Division by Zero");
                    }

                    dialog.add(prompt);
```

```
        dialog.setVisible(true);
    }
});

frame.add(submit);
frame.setVisible(true);
}
}
```

16.3 Output

17 Simple Interest

17.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

17.2 Code

```
package org.projects.prog17;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Interest extends Applet implements ActionListener {
    TextField principle, rate, time;
    Label result;
    Button submit;

    public void init() {
        principle = new TextField("principle");
        rate = new TextField("rate%");
        time = new TextField("time");

        add(new Label("Fill the following"));
        add(principle);
        add(rate);
        add(time);

        submit = new Button("Submit");
        submit.addActionListener(this);
        add(submit);

        result = new Label();
        add(result);
    }

    public void actionPerformed(ActionEvent e) {
        int p = Integer.parseInt(principle.getText());
        int r = Integer.parseInt(rate.getText());
        int n = Integer.parseInt(time.getText());

        float si = (p * r * n) / 100;
        result.setText("" + si);
    }
}
```

17.3 Output

18 Mouse Coordinates

18.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

18.2 Code

```
package org.projects.prog18;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class MouseProg extends Applet implements MouseMotionListener {
    int x, y;

    public void init() {
        x = 0;
        y = 0;

        addMouseMotionListener(this);
    }

    public void paint(Graphics g) {
        g.drawString("x = " + x + " y = " + y, 20, 20);
    }

    public void mouseMoved(MouseEvent e) {
        x = e.getX();
        y = e.getY();
        repaint();
    }

    public void mouseDragged(MouseEvent e) {}
}
```

18.3 Output

19 Simple Banner

19.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

19.2 Code

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

public class Banner extends Applet implements Runnable {
    String bannerText;
    Thread scroll;

    public void init() {
        bannerText = "REALLY COOL BANNER ";
        scroll = new Thread(this);
    }

    public void start() {
        scroll.start();
    }

    public void paint(Graphics g) {
        g.drawString(bannerText, 100, 100);
    }

    public void run() {
        try {
            while (true) {
                Thread.sleep(500);
                bannerText = bannerText.substring(1) + bannerText.charAt(0);
                repaint();
            }
        } catch (Exception e) {
        }
    }
}
```

19.3 Output

20 Grid Layout Manager

20.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

20.2 Code

```
package org.projects.prog20;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GridLay {
    public static void main(String args[]) {
        JFrame frame = new JFrame("Simple Grid");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setLayout(new GridLayout(3, 1));

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(1, 2));
        panel.add(new JLabel("Enter Name"));

        JTextField name = new JTextField();
        panel.add(name);

        frame.add(panel);

        JButton submit = new JButton("Submit");
        JLabel hello = new JLabel();

        submit.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    hello.setText("Hello " + name.getText() + "!!");
                }
            }
        );
        frame.add(submit);

        frame.add(hello);

        frame.setVisible(true);
    }
}
```

20.3 Output

21 Priority Threads

21.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

21.2 Code

```
package org.projects.prog21;

public class Threading {
    public static Runnable simple_runnable =
        new Runnable() {
            public void run() {
                String name = Thread.currentThread().getName();
                for (int i = 0; i < 10; i++) {
                    System.out.println(name + " " + i);
                }
            }
        };

    public static void main(String args[]) {
        Thread th0 = new Thread(simple_runnable), th1 = new Thread(simple_runnable);

        th0.setPriority(Thread.MIN_PRIORITY);
        th1.setPriority(Thread.MAX_PRIORITY);

        th0.start();
        th1.start();
    }
}
```

21.3 Output

22 Employee Details

22.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

22.2 Code

```
package org.projects.prog22;

import java.sql.*;

public class Employee {
    public static void main(String args[]) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String username = "root", password = "coolPass", dbname = "dbone";
            Connection cn =
                DriverManager.getConnection(
                    // "jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
                    "jdbc:mysql://localhost:3306/" + dbname, username, password);

            Statement state = cn.createStatement();
            ResultSet rs = state.executeQuery("select * from employee");

            while (rs.next()) {
                System.out.println("Id: " + rs.getInt(1));
                System.out.println("Name: " + rs.getString(2));
                System.out.println("Designation: " + rs.getString(3));
                System.out.println();
            }

            cn.close();
        } catch (Exception e) {
        }
    }
}
```

22.3 Output

23 Update In JDBC

23.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

23.2 Code

```
package org.projects.prog23;

import java.sql.*;
import java.util.Scanner;

public class Student {
    public static Scanner sc = new Scanner(System.in);

    public static void main(String args[]) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String username = "root", password = "coolPass", dbname = "dbone";
            Connection cn =
                DriverManager.getConnection(
                    // "jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
                    "jdbc:mysql://localhost:3306/" + dbname, username, password);
            System.out.println("1 - add Student");
            System.out.println("2 - update Student");

            switch (sc.nextInt()) {
                case 1:
                    addStudent(cn);
                    break;
                case 2:
                    updateStudent(cn);
                    break;
                default:
                    System.out.println("Invalid Choice");
                    break;
            }

            displayStudents(cn);
            cn.close();
        } catch (Exception e) {
        }
    }

    public static void addStudent(Connection cn) {
        sc.nextLine(); // ignore garbage
        System.out.println("Enter name");
        String name = sc.nextLine();

        System.out.println("Enter roll number, age");
        int roll = sc.nextInt(), age = sc.nextInt();

        System.out.println("Enter Grade");
        char grade = sc.next().charAt(0);
    }
}
```

```

// to create the table use the following mysql code
// create table students (
//   roll int unique not null,
//   name varchar(20) not null,
//   age int not null,
//   grade char(1) not null
// );

try {
    cn.createStatement()
        .execute(
            String.format(
                "INSERT INTO students VALUES (%d, '%s', %d, '%c')", roll, name,
                ↵ age, grade));
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}

public static void updateStudent(Connection cn) {
    System.out.println("Enter rol number to update");
    int roll = sc.nextInt();

    System.out.println("Enter new age");
    int age = sc.nextInt();

    System.out.println("Enter new grade");
    char grade = sc.next().charAt(0);

    try {
        cn.createStatement()
            .executeUpdate(
                String.format(
                    "UPDATE students SET age = %d, grade = '%c' WHERE roll = %d", age,
                    ↵ grade, roll));
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

public static void displayStudents(Connection cn) {
    try {
        ResultSet set = cn.createStatement().executeQuery("select * from students");
        while (set.next()) {
            System.out.println(
                String.format(
                    "roll: %d, name: %s, age: %d, grade: %s",
                    set.getInt(1), set.getString(2), set.getInt(3), set.getString(4)));
        }
    } catch (Exception e) {
    }
}
}

```

23.3 Output

24 Product Details

24.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

24.2 Code

```
package org.projects.prog24;

import java.sql.*;
import java.util.Scanner;

public class Product {
    public static Scanner sc = new Scanner(System.in);

    public static void main(String args[]) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String username = "root", password = "coolPass", dbname = "dbone";
            Connection cn =
                DriverManager.getConnection(
                    // "jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
                    "jdbc:mysql://localhost:3306/" + dbname, username, password);

            boolean done = false;
            while (!done) {
                System.out.println("1 - add Product");
                System.out.println("2 - delete Product");
                System.out.println("3 - display");

                switch (sc.nextInt()) {
                    case 1:
                        addProduct(cn);
                        break;
                    case 2:
                        deleteProduct(cn);
                        break;
                    case 3:
                        displayProducts(cn);
                        break;
                    default:
                        System.out.println("Invalid Choice");
                        done = true;
                        break;
                }
            }
            cn.close();
        } catch (Exception e) {
        }
    }

    public static void addProduct(Connection cn) {
        sc.nextLine(); // ignore garbage
        System.out.println("Enter Product name");
        String name = sc.nextLine();
    }
}
```

```

System.out.println("Enter product price");
float price = sc.nextFloat();

// to create the table use the following mysql code
// create table products (
//   id int unique AUTO_INCREMENT,
//   name varchar(20) not null,
//   price decimal(8, 2) not null,
// );

try {
    cn.createStatement()
        .execute(
            String.format("INSERT INTO products (name, price) VALUES ('%s', %f)",
                ↪ name, price));
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}

public static void deleteProduct(Connection cn) {
    System.out.println("Enter and id to delete");
    int id = sc.nextInt();

    try {
        cn.createStatement().execute(String.format("DELETE FROM products WHERE id =
            ↪ %d", id));
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

public static void displayProducts(Connection cn) {
    try {
        ResultSet set = cn.createStatement().executeQuery("select * from products");
        while (set.next()) {
            System.out.println(
                String.format(
                    "id: %d, name: %s, cost: %f", set.getInt(1), set.getString(2),
                    ↪ set.getFloat(3)));
        }
    } catch (Exception e) {
    }
}
}

```

24.3 Output