CS-22 Programming with Java

BCA Semester – 4

Journal

H & H Kotak Institute of Science, Rajkot

BCA Department

H & H B Kotak Institute of Science, Rajkot

BCA Department

Laboratory Certificate

This is to certify that Smt./Shri ____

has satisfactory completed **BCA Semester-4** practical experiments of subject **CS-22 Programming with Java** during the academic year __________. Her/His enrollment number is ______________________ registered at Saurashtra University, Rajkot.

Date: _____________

Subject In-Charge

Head of the Department

Index

Index

Unit – 1

History, Introduction and Language Basics, Classes and Objects

1. Hello World Program

```
class HelloJava {
      public static void main(String arg[]) {
            System.out.println("Hello Java");
            System.out.print("Java is an OOP");
      }
}
```

2. Java Variables

```
//Java Variables
class VariableDemo {
      public static void main(String[] arg) {
            int i=10;
            String n="Java";
            float f=5.5f;
            System.out.println("Value of i: "+i);
            System.out.println("Value of n: "+n);
            System.out.println("Value of f: "+f);
      }
}
```

3. Leap Year

```
//Leap year example using if...else

public class LeapYearExample {
      public static void main(String[] args) {
          int year=2021;
          if(((year % 4==0) && (year % 100!=0)) || (year % 400==0)){
            System.out.println("LEAP YEAR");
          }
          else{
            System.out.println("COMMON YEAR");
          }
      }
}
```

4. Find vowels

```
//Vowels using switch...case

public class SwitchExample {
      public static void main(String[] args) {
          char ch='L';
          switch(ch)
          {
              case 'a':
                  System.out.println("Vowel");
                  break;
              case 'e':
                  System.out.println("Vowel");
                break;
              case 'i':
                System.out.println("Vowel");
                  break;
              case 'o':
                  System.out.println("Vowel");
                  break;
              case 'u':
```

```
         System.out.println("Vowel");
         break;
      case 'A':
         System.out.println("Vowel");
         break;
      case 'E':
         System.out.println("Vowel");
         break;
       case 'I':
         System.out.println("Vowel");
         break;
       case 'O':
         System.out.println("Vowel");
         break;
       case 'U':
         System.out.println("Vowel");
         break;
       default:
        System.out.println("Consonant");
     }
   }
}
```

5. Passing an array to function

```
//Java Program to demonstrate the way of passing an array

class FindMin{
   static void min(int arr[]){
      int min=arr[0];
      for(int i=1;i<arr.length;i++)
         if(min>arr[i]) min=arr[i];
      System.out.println(min);
   }
   public static void main(String args[]){
      int a[]={33,3,1,5};//declaring and initializing an array
      min(a);//passing array to method
   }
}
```

6. Class and Objects

```
//Oop Example
class Student{
   int id;
   String name;
}

class TestStudent{
   public static void main(String args[]){
    Student s1=new Student();
    Student s2=new Student();
    s1.id=101;
    s1.name="Ritul";
    s2.id=102;
    s2.name="Amit";
    System.out.println(s1.id+" "+s1.name);
    System.out.println(s2.id+" "+s2.name);
  }
}
```

7. Class with Method

```java
//Class with method

class Employee{
    int id;
    String name;
    float salary;
    void setData(int i, String n, float s) {
        id=i;
        name=n;
        salary=s;
    }
    void getData() {
      System.out.println(id+" "+name+" "+salary);
    }
}
public class TestEmployee {
  public static void main(String[] args) {
    Employee e1=new Employee();
    Employee e2=new Employee();
    e1.setData(101,"Ravi",45000);
    e2.setData(102,"Mohit",25000);
    e1.getData();
    e2.getData();
  }
}
```

8. Parameterized constructor

```java
//Java Program to demonstrate the use of the parameterized constructor.

class Student4{
    int id;
    String name;
    //creating a parameterized constructor
    Student4(int i,String n){
      id = i;
      name = n;
    }
    //method to display the values
    void display(){
      System.out.println(id+" "+name);
    }
    public static void main(String args[]){
      //creating objects and passing values
      Student4 s1 = new Student4(111,"Ritul");
      Student4 s2 = new Student4(222,"Ravi");
      //calling method to display the values of object
      s1.display();
      s2.display();
   }
}
```

9. Constructor Overloading

```java
//Java program to overload constructors
class Student5{
    int id;
    String name;
    int age;
    //creating two arg constructor
    Student5(int i,String n){
           id = i;
           name = n;
    }
    //creating three arg constructor
    Student5(int i,String n,int a){
        id = i;
        name = n;
        age=a;
    }
    void display(){System.out.println(id+" "+name+" "+age);}

    public static void main(String args[]){
       Student5 s1 = new Student5(111,"Mohit");
       Student5 s2 = new Student5(222,"Priyanshu",25);
       s1.display();
       s2.display();
   }
}
```

10. Jagged Array

```
//Program to Jagged Array.

class Test
{
    public static void main(String[] args)
    {
        int[][] arr = new int[2][];// Declare the array

        arr[0] = new int[] { 11, 21, 56, 78 };// Initialize the array
        arr[1] = new int[] { 42, 61, 37, 41, 59, 63 };

        // Traverse array elements
        for (int i = 0; i < arr.length; i++)
        {
            for (int j = 0; j < arr[i].length; j++)
            {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

11. Copy constructor

```
//Copy constructor...

class Student6{
    int id;
    String name;
    //constructor to initialize integer and string
    Student6(int i,String n){
        id = i;
        name = n;
    }
    //constructor to initialize another object
    Student6(Student6 s){
        id = s.id;
        name =s.name;
    }
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
        Student6 s1 = new Student6(111,"Krupa");
        Student6 s2 = new Student6(s1);
        s1.display();
        s2.display();
   }
}
```

12. Java Inheritance

```
//Java Inheritance Demo

class Animal{
    void eat(){
        System.out.println("eating....");
        }
}
class Dog extends Animal{
        void bark(){
                System.out.println("barking...");
        }
}
class BabyDog extends Dog{
        void weep(){
                System.out.println("weeping...");
        }
}
class TestInheritance{
        public static void main(String args[]){
                BabyDog d=new BabyDog();
                d.weep();
                d.bark();
                d.eat();
        }
}
```

13. Method Overloading

```
//Method Overloading Demo...
class Adder{
        static int add(int a, int b) {
                return a+b;
        }
        static double add(double a, double b) {
                return a+b;
        }
}
class TestOverloading{
        public static void main(String[] args){
                System.out.println(Adder.add(11,11));
                System.out.println(Adder.add(12.3,12.6));
        }
}
```

Unit – 2

Inheritance, Java Packages

14. Constructor in Inheritance

```
//Constructor in Inheritance
class Animal{
      Animal() {
            System.out.println("From animal constructor");
      }
      void eat(){
            System.out.println("eating....");
      }
      protected void finalize() {
            System.out.println("End of animal");
      }
}
class Dog extends Animal{
      Dog() {
            System.out.println("From dog constructor");
      }
      void bark(){
            System.out.println("barking...");
      }
      protected void finalize() {
            System.out.println("End of dog");
      }

}
class BabyDog extends Dog{
      BabyDog() {
            System.out.println("From babydog constructor");
      }
      void weep(){
            System.out.println("weeping...");
      }
      protected void finalize() {
            System.out.println("End of babydog");
      }

}
class TestInheritance2{
      public static void main(String args[]){
            BabyDog d=new BabyDog();
            d.weep();
            d.bark();
            d.eat();
            d=null;
            System.gc();
      }
}
```

15. Abstract Class

```
//abstract class demo.

abstract class Shape{
      abstract void draw();
}

class Rectangle extends Shape{
      void draw(){System.out.println("drawing rectangle");}
}
class Circle extends Shape{
      void draw(){System.out.println("drawing circle");}
}

class TestAbstraction{
      public static void main(String args[]){
            Shape s1=new Circle();
            Shape s2=new Rectangle();
            s1.draw();
            s2.draw();
      }
}
```

16. Final Class

```
//Final Class

final class ParentClass
{
    void showData()
    {
        System.out.println("This is a method of final Parent class");
    }
}

//It will throw compilation error
class ChildClass extends ParentClass
{
    void showData()
    {
        System.out.println("This is a method of Child class");
    }
}
class MainClass
{
    public static void main(String arg[])
    {
        ParentClass obj = new ChildClass();
        obj.showData();
    }
}
```

17. Java Interface

```
//Interface Demo...
interface Animal {
   public void eat();
   public void travel();
}

class MammalInt implements Animal {

   public void eat() {
      System.out.println("Mammal eats");
   }

   public void travel() {
      System.out.println("Mammal travels");
   }

   public int noOfLegs() {
      return 0;
   }

}

```

```
public class Main {
   public static void main(String args[]) {
      MammalInt m = new MammalInt();
      m.eat();
      m.travel();
   }
}
```

18. Inner Class

```
//Inner class demo.

class Main {
    private int data=30;
    class Inner{
        void msg(){System.out.println("data is "+data);}
    }
    public static void main(String args[]){
        Main obj=new Main();
        Main.Inner in=obj.new Inner();
        in.msg();
    }
}
```

19. util.Date class

```
import java.util.Date;

public class Main {

   public static void main(String args[]) {

      Date date = new Date();

      System.out.println(date.toString());
   }
}
```

20. Java Wrapper Classes

```
//wrapper classes objects and vice-versa

public class Main {
    public static void main(String args[]){
        byte b=10;
        short s=20;
        int i=30;
        long l=40;
        float f=50.0F;
        double d=60.0D;
        char c='a';
        boolean b2=true;

        //Autoboxing: Converting primitives into objects
        Byte byteobj=b;
        Short shortobj=s;
        Integer intobj=i;
        Long longobj=l;
        Float floatobj=f;
        Double doubleobj=d;
        Character charobj=c;
        Boolean boolobj=b2;

        //Printing objects
        System.out.println("---Printing object values---");
        System.out.println("Byte object: "+byteobj);
        System.out.println("Short object: "+shortobj);
```

```java
        System.out.println("Integer object: "+intobj);
        System.out.println("Long object: "+longobj);
        System.out.println("Float object: "+floatobj);
        System.out.println("Double object: "+doubleobj);
        System.out.println("Character object: "+charobj);
        System.out.println("Boolean object: "+boolobj);

        //Unboxing: Converting Objects to Primitives
        byte bytevalue=byteobj;
        short shortvalue=shortobj;
        int intvalue=intobj;
        long longvalue=longobj;
        float floatvalue=floatobj;
        double doublevalue=doubleobj;
        char charvalue=charobj;
        boolean boolvalue=boolobj;

        //Printing primitives
        System.out.println("---Printing primitive values---");
        System.out.println("byte value: "+bytevalue);
        System.out.println("short value: "+shortvalue);
        System.out.println("int value: "+intvalue);
        System.out.println("long value: "+longvalue);
        System.out.println("float value: "+floatvalue);
        System.out.println("double value: "+doublevalue);
        System.out.println("char value: "+charvalue);
        System.out.println("boolean value: "+boolvalue);
    }
}
```

21. Creating user defined package

```java
//Creating user-defined package..

package mypack;

public class Simple{
 public static void main(String args[]){
    System.out.println("Welcome to package");
   }
}
```

22. Java StringTokenizer

```java
import java.util.StringTokenizer;

public class Simple {
 public static void main(String args[]){
   StringTokenizer st = new StringTokenizer("Java OOP Programing Language"," ");
     while (st.hasMoreTokens()) {
         System.out.println(st.nextToken());
     }
   }
}
```

Unit – 3

Exception Handling, Threading and Streams (Input and Output)

23. Exception Handling

```
//Exception Handling Demonstration
public class Main
{
       public static void main(String[] args) {
              int a=10,b=0,c=0;
              System.out.println("Start of main()");
              try{
                  c=a/b;
              }catch(ArithmeticException ae) {
                  System.out.println(ae);
              }finally {
                  System.out.println("I am always there...");
              }
              System.out.println("Value of C:"+c);
              System.out.println("End of main()");
       }
}
```

24. Multiple catch statements

```
//multiple catch statements
public class Main {

    public static void main(String[] args) {

           try{
                int a[]=new int[5];
                a[5]=30/0;
               }
               catch(ArithmeticException e)
                  {
                   System.out.println("Arithmetic Exception occurs");
                  }
               catch(ArrayIndexOutOfBoundsException e)
                  {
                   System.out.println("ArrayIndexOutOfBounds Exception occurs");
                  }
               catch(Exception e)
                  {
                   System.out.println("Parent Exception occurs");
                  }
               System.out.println("rest of the code");
    }
}
```

24. Custom exception

```
//Custom exception example...
class InvalidAgeException extends Exception{
 InvalidAgeException(String s){
  super(s);
 }
}
class Main {

   static void validate(int age)throws InvalidAgeException{
     if(age<18)
      throw new InvalidAgeException("not valid");
     else
      System.out.println("welcome to vote");
   }

   public static void main(String args[]){
      try{
      validate(13);
      }catch(Exception m){System.out.println("Exception occured: "+m);}

      System.out.println("rest of the code...");
  }
}
```

25. Multithreading using Thread Class

```
public class ThreadDemo1 {

    public static void main(String[] args) {
        System.out.println("Start of main");
        MyThread1 mt1 = new MyThread1();
        MyThread2 mt2 = new MyThread2();
        mt1.start();
        mt2.start();
        System.out.println("End of main");
    }
}

class MyThread1 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-1."+i);
        }
    }
}

class MyThread2 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-2."+i);
        }
    }
}
```

26. Multithreading using Runnable interface

```
public class ThreadDemo2 {
    public static void main(String[] args) {
        System.out.println("Start of main");
        MyThread mt = new MyThread();
        Thread t1 = new Thread(mt,"Thread-1");
        Thread t2 = new Thread(mt,"Thread-2");
        t1.start();
        t2.start();
        System.out.println("End of main");
    }
}

class MyThread implements Runnable {
    public void run() {
        for(int i=1;i<=10;i++) {
            System.out.println(Thread.currentThread().getName()+"."+i);
        }
    }
}
```

27. Thread Scheduling

```
public class ThreadDemo3 {

    public static void main(String[] args) {
        System.out.println("Start of main");
        MyThread1 mt1 = new MyThread1();
        MyThread2 mt2 = new MyThread2();
        mt1.start();
        mt2.start();
        System.out.println("End of main");
    }

}

```

```
class MyThread1 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-1."+i);
            Thread.yield();
        }
    }
}

class MyThread2 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-2."+i);
            Thread.yield();
        }
    }
}
```

28. Thread Joins

```
public class ThreadDemo3 {

    public static void main(String[] args) {
        try {
        System.out.println("Start of main");
        MyThread1 mt1 = new MyThread1();
        MyThread2 mt2 = new MyThread2();
        mt1.start();
        mt1.join();
        mt2.start();
        mt2.join();
        System.out.println("End of main");
        }catch(Exception e){}
    }

}

class MyThread1 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-1."+i);
            try {
                 sleep(100);
            }catch(Exception e){ }
        }
    }
}

class MyThread2 extends Thread{
    public void run(){
        for(int i=1;i<=10;i++) {
            System.out.println("MyThread-2."+i);
            try {
                 sleep(200);
            }catch(Exception e){ }
        }
    }
}
```

29. Thread Priorates

```
public class ThreadDemo4 {

    public static void main(String[] args) {
        System.out.println("Start main");
        MyThread mt = new MyThread();
        Thread t1 = new Thread(mt,"Thread-1");
        Thread t2 = new Thread(mt,"Thread-2");
        t1.start();
        t2.start();
        t2.setPriority(t1.getPriority()+5);
        System.out.println("End main");
    }

}

```

```
class MyThread implements Runnable {
    public void run() {
        for(int i = 1;i<=10;i++) {
            System.out.println(Thread.currentThread().getName());
        }
    }
}
```

30. File Class

```
import java.io.*;

public class IODemo1 {

    public static void main(String[] args) {

        try {
            File f = new File("abc.txt");
            if(f.createNewFile()) {
                System.out.println("File Sucessfully created");
            }
            else {
                System.out.println("File already exist");
            }
            System.out.println("File name : "+f.getName());
            System.out.println("Path: "+f.getPath());
            System.out.println("Absolute path: " +f.getAbsolutePath());
            System.out.println("Parent: "+f.getParent());
            System.out.println("Exists : "+f.exists());
            System.out.println("Is writeable: "+f.canWrite());
            System.out.println("Is readable: "+f.canRead());
            System.out.println("Is a directory: "+f.isDirectory());
            System.out.println("File Size in bytes: "+f.length());
        }catch(Exception e){
               System.out.println(e);
        }
    }
}
```

31. Bytestream Class to read file

```
import java.io.*;

public class IODemo3 {
    public static void main(String[] args) {
        System.out.println("Content of output.txt file:\n");
        try{
            FileInputStream fin = new FileInputStream("output.txt");
            int c;

            while((c=fin.read())!= -1 ){
                System.out.print((char)c);
            }
        }catch(Exception e) { }
    }
}
```

32. Bytestream Class to create file

```
import java.io.*;

public class IODemo2 {

    public static void main(String[] args) {
        try{
            //DataInputStream out = new DataInputStream(System.in);
            BufferedInputStream out = new BufferedInputStream(System.in);
            FileOutputStream fout = new FileOutputStream("output.txt");
            System.out.println("Enter text (enter & to end): &");
            int ch;
            while ((ch = (char) out.read()) != '&')
                fout.write((char)ch);
            fout.close();
        }catch(Exception e){}
    }
}
```

33. Character stream Class to read and write file

```java
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class IOStreamsExample {
   public static void main(String args[]) throws IOException {
      //Creating FileReader object
      File file = new File("D:/myFile.txt");
      FileReader reader = new FileReader(file);
      char chars[] = new char[(int) file.length()];
      //Reading data from the file
      reader.read(chars);
      //Writing data to another file
      File out = new File("D:/CopyOfmyFile.txt");
      FileWriter writer = new FileWriter(out);
      //Writing data to the file
      writer.write(chars);
      writer.flush();
      System.out.println("Data successfully written in the specified file");
   }
}
```

Unit – 4

JavaFx Basics and Event-driven programming and animations

34. JavaFX HelloWorld

```java
//HelloWorld
package Unit4;

import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;


public class FirstApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);
        Scene scene = new Scene(root, 300, 250);
        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

35. Text and Font Example

```java
package Unit4;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class TextExample extends Application {

    @Override
    public void start(Stage primaryStage) {

        Text text = new Text();
        text.setText("Hello !! Welcome to JavaFX");
        text.setFill(Color.BLUE);
        text.setStrokeWidth(2);
        text.setStroke(Color.RED);
        text.setFont(Font.font("Comic Sans
MS",FontWeight.BOLD,FontPosture.REGULAR,30));
        StackPane root = new StackPane();
        root.getChildren().add(text);

        Scene scene = new Scene(root, 600, 400);

        primaryStage.setTitle("Text Example");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

36. 2D Shape Example

```
package Unit4;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;

public class ShapeExample extends Application {
    @Override
    public void start(Stage primaryStage) {
        Group group = new Group();
        Rectangle rect = new Rectangle();
        rect.setX(10);
        rect.setY(10);
        rect.setWidth(100);
        rect.setHeight(100);
        rect.setArcHeight(35);
        rect.setArcWidth(35);
        rect.setFill(Color.RED);

        Circle circle = new Circle();
        circle.setCenterX(300);
        circle.setCenterY(200);
        circle.setRadius(100);
        circle.setFill(Color.GREEN);

        group.getChildren().addAll(rect,circle);
        Scene scene = new Scene(group, 600, 400);
        primaryStage.setTitle("2D Shape Example");
        primaryStage.setScene(scene);
        primaryStage.show();
     }

    public static void main(String[] args) {
        launch(args);
    }

}
```

37. 3D Shape Example

```java
package Unit4;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.PerspectiveCamera;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Box;
import javafx.scene.shape.Cylinder;
import javafx.stage.Stage;

public class Shapes3DExample extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        //Creating Boxes
        Box box1 = new Box();

         //Setting properties for first box
        box1.setHeight(100);
        box1.setWidth(100);
        box1.setDepth(400);
        box1.setTranslateX(200);
        box1.setTranslateY(200);
        box1.setTranslateZ(200);
        Cylinder cyn = new Cylinder();

        //setting the radius and height for the cylinder
        cyn.setRadius(80);
        cyn.setHeight(200);
        cyn.setTranslateX(400);
        cyn.setTranslateY(250);

        //Setting the perspective camera
        PerspectiveCamera camera = new PerspectiveCamera();
        camera.setTranslateX(100);
        camera.setTranslateY(100);
        camera.setTranslateZ(50);

        //Configuring Group, Scene and Stage
        Group root = new Group();
        root.getChildren().addAll(box1, cyn);
        Scene scene = new Scene(root, 450, 350, Color.LIMEGREEN);
        scene.setCamera(camera);
        primaryStage.setScene(scene);
        primaryStage.setTitle("3DShape Example");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

38. Grid Layout Example

```java
package Unit4;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class GridPaneExample extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        Label first_name=new Label("First Name");
        Label last_name=new Label("Last Name");
        TextField tf1=new TextField();
        TextField tf2=new TextField();
        Button Submit=new Button ("Submit");
        GridPane root=new GridPane();
        Scene scene = new Scene(root,400,200);
        root.addRow(0, first_name,tf1);
        root.addRow(1, last_name,tf2);
        root.addRow(2, Submit);
        primaryStage.setTitle("GridPane Layout");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

39. Image Input Example

```java
package Unit4;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.effect.ImageInput;
import javafx.scene.image.Image;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;

public class ImageInputExample extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        Image img = new Image("https://upload.wikimedia.org/wikipedia/commons/e/e3/Animhorse.gif");
        ImageInput imginput = new ImageInput();
        Rectangle rect = new Rectangle();
        imginput.setSource(img);
        imginput.setX(20);
        imginput.setY(100);
        Group root = new Group();
        rect.setEffect(imginput);
        root.getChildren().add(rect);
        Scene scene = new Scene(root, 530, 500, Color.BLACK);
        primaryStage.setScene(scene);
        primaryStage.setTitle("ImageInput Example");
        primaryStage.show();

    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

40. Animation Example

```java
package Unit4;
import javafx.animation.FadeTransition;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
import javafx.util.Duration;
public class AnimationExample extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        Circle cir = new Circle(250,120,80);
        cir.setFill(Color.RED);
        cir.setStroke(Color.BLACK);
        FadeTransition fade = new FadeTransition();
        fade.setDuration(Duration.millis(5000));
        fade.setFromValue(10);
        fade.setToValue(0.1);
        fade.setCycleCount(1000);
        fade.setAutoReverse(true);
        fade.setNode(cir);
        fade.play();
        Group root = new Group();
        root.getChildren().addAll(cir);
        Scene scene = new Scene(root,500,250,Color.WHEAT);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Translate Transition example");
        primaryStage.show();

    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Unit – 5

GUI using SWING Event Handling

41. Label Example

```java
package Unit5;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.stage.Stage;
public class JavafxLabel extends Application {
   public void start(Stage stage) {
      Label label = new Label("Sample label");
      Font font = Font.font("Brush Script MT", FontWeight.BOLD,
FontPosture.REGULAR, 25);
      label.setFont(font);
      label.setTextFill(Color.BROWN);
      label.setTranslateX(10);
      label.setTranslateY(10);
      Group root = new Group();
      root.getChildren().add(label);
      Scene scene = new Scene(root, 400, 300, Color.BEIGE);
      stage.setTitle("Label Example");
      stage.setScene(scene);
      stage.show();
   }
   public static void main(String args[]){
      launch(args);
   }
}
```

42. Button and Textfield Example

```java
package Unit5;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class JavafxTextfield extends Application {
    public void start(Stage stage) {
        TextField textField1 = new TextField();
        TextField textField2 = new TextField();
        Label label1 = new Label("Enter Text: ");
        Label label2 = new Label("Entered Text: ");
        Button btn = new Button("Click");
        btn.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent arg0) {
                // TODO Auto-generated method stub
                textField2.setText(textField1.getText());
            }
        });
        HBox box = new HBox(5);
        box.setPadding(new Insets(25, 5, 5, 50));
        box.getChildren().addAll(label1, textField1, label2, textField2,
btn);
        //Setting the stage
        Scene scene = new Scene(box, 595, 150, Color.BEIGE);
        stage.setTitle("Text Field Example");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String args[]) {
        launch(args);
    }
}
```

43. ListView Example

```
package Unit5;

import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.ListView;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.stage.Stage;

public class JavafxListview extends Application {
    public void start(Stage stage) {
        Label label = new Label("Educational qualification:");
        Label label2 = new Label();
        Font font = Font.font("verdana", FontWeight.BOLD,
FontPosture.REGULAR, 12);
        label.setFont(font);
        ObservableList<String> names =
FXCollections.observableArrayList("Engineering", "MCA", "MBA",
"Graduation", "MTECH", "Mphil", "Phd");
        ListView<String> listView = new ListView<String>(names);
        listView.setMaxSize(200, 160);

        Button btn = new Button("Select Course");
        btn.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent arg0) {
                // TODO Auto-generated method stub
                label2.setText(listView.getItems().toString());
            }
        });
        VBox layout = new VBox(10);
        layout.setPadding(new Insets(5, 5, 5, 50));
        layout.getChildren().addAll(label, listView, btn, label2);
        layout.setStyle("-fx-background-color: BEIGE");
        //Setting the stage
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("List View Example");
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String args[]) {
        launch(args);
    }
}
```

44. Slider Example

```
package Unit5;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Slider;
import javafx.stage.Stage;

public class SliderExample extends Application {
      public void start(Stage stage)
      {
            Group root = new Group();
            Scene scene = new Scene(root, 600, 400);
            stage.setScene(scene);
            stage.setTitle("Slider Sample");
            Slider slider = new Slider();
            root.getChildren().add(slider);
            stage.show();
      }
      public static void main(String[] args)
      {
            launch(args);
      }
}
```

45. Play Audio Example

```
package Unit5;

import java.io.File;
import javafx.application.Application;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.stage.Stage;

public class PlayAudio extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        String path = "C:\\Users\\malay\\Downloads\\test.mp3";
        Media media = new Media(new File(path).toURI().toString());
        MediaPlayer mediaPlayer = new MediaPlayer(media);
        mediaPlayer.setAutoPlay(true);
        primaryStage.setTitle("Playing Audio");
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

45. Play Video Example

```java
package Unit5;
import java.io.File;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.scene.media.MediaView;
import javafx.stage.Stage;

public class PlayVideo extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        String path = "C:\\Users\\malay\\Downloads\\test.mp4";
        Media media = new Media(new File(path).toURI().toString());
        MediaPlayer mediaPlayer = new MediaPlayer(media);
        MediaView mediaView = new MediaView(mediaPlayer);
        mediaPlayer.setAutoPlay(true);
        Group root = new Group();
        root.getChildren().add(mediaView);
        Scene scene = new Scene(root, 500, 400);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Playing video");
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```