

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

Sr.	Name of Experiments	Page No.	Date of Experiment	Date of Supervision	Remarks
	Unit-1				
1	Hello World Program	6			
2	Java Variables	6			
3	Leap Year	6			
4	Find vowels	6			
5	Passing an array to function	7			
6	Class and Objects	7			
7	Class with Method	8			
8	Parameterized constructor	8			
9	Constructor Overloading	8			
10	Jagged Array	9			
11	Copy constructor	9			
12	Java Inheritance	9			
13	Method Overloading	10			
	Unit-2				
14	Constructor in Inheritance	12			
15	Abstract Class	12			
16	Final Class	13			
17	Java Interface	13			
18	Inner Class	14			
19	util.Date class	14			
20	Java Wrapper Classes	14			
21	Creating user defined package	15			
22	Java StringTokenizer	15			
	Unit-3				
23	Exception Handling	17			
24	Multiple catch statements	17			
25	Multithreading using Thread Class	18			
26	Multithreading using Runnable interface	18			
27	Thread Scheduling	18			
28	Thread Joins	19			
29	Thread Priorities	19			
30	File Class	20			
31	Bytestream Class to read file	20			
32	Bytestream Class to create file	20			
33	Character stream Class to read and write file	21			
	Unit-4				
34	JavaFX HelloWorld	23			

Index

Sr.	Name of Experiments	Page No.	Date of Experiment	Date of Supervision	Remarks
35	Text and Font Example	23			
36	2D Shape Example	24			
37	3D Shape Example	25			
38	Grid Layout Example	25			
39	Image Input Example	26			
40	Animation Example	27			
	Unit-5				
41	Label Example	29			
42	Button and Textfield Example	29			
43	ListView and Combobox Example	30			
44	Slider Example	31			
45	Play Audio Example	31			
46	Play Video Example	32			

Unit – 1

History, Introduction and

Language Basics, Classes and

Objects

1. Hello World Program

```
1 class HelloJava {  
2     public static void main(String arg[]) {  
3         System.out.println("Hello Java");  
4         System.out.print("Java is an OOP");  
5     }  
6 }
```

2. Java Variables

```
1 //Java Variables  
2 class VariableDemo {  
3     public static void main(String[] arg) {  
4         int i=10;  
5         String n="Java";  
6         float f=5.5f;  
7         System.out.println("Value of i: "+i);  
8         System.out.println("Value of n: "+n);  
9         System.out.println("Value of f: "+f);  
10    }  
11 }
```

3. Leap Year

```
1 //Leap year example using if...else  
2  
3 public class LeapYearExample {  
4     public static void main(String[] args) {  
5         int year=2021;  
6         if(((year % 4==0) && (year % 100!=0)) || (year % 400==0)){  
7             System.out.println("LEAP YEAR");  
8         }  
9         else{  
10             System.out.println("COMMON YEAR");  
11         }  
12     }  
13 }
```

4. Find vowels

```
1 //Vowels using switch...case  
2  
3 public class SwitchExample {  
4     public static void main(String[] args) {  
5         char ch='L';  
6         switch(ch)  
7         {  
8             case 'a':  
9                 System.out.println("Vowel");  
10                break;  
11             case 'e':  
12                 System.out.println("Vowel");  
13                break;  
14             case 'i':  
15                 System.out.println("Vowel");  
16                break;  
17             case 'o':  
18                 System.out.println("Vowel");  
19                break;  
20             case 'u':  
21                 System.out.println("Vowel");  
22                break;
```

```

21         System.out.println("Vowel");
22         break;
23     case 'A':
24         System.out.println("Vowel");
25         break;
26     case 'E':
27         System.out.println("Vowel");
28         break;
29     case 'I':
30         System.out.println("Vowel");
31         break;
32     case 'O':
33         System.out.println("Vowel");
34         break;
35     case 'U':
36         System.out.println("Vowel");
37         break;
38     default:
39         System.out.println("Consonant");
40     }
41 }
42 }
```

5. Passing an array to function

```

1 //Java Program to demonstrate the way of passing an array
2
3 class FindMin{
4     static void min(int arr[]){
5         int min=arr[0];
6         for(int i=1;i<arr.length;i++)
7             if(min>arr[i]) min=arr[i];
8         System.out.println(min);
9     }
10    public static void main(String args[]){
11        int a[]={33,3,1,5}//declaring and initializing an array
12        min(a);//passing array to method
13    }
14 }
```

6. Class and Objects

```

1 //Oop Example
2 class Student{
3     int id;
4     String name;
5 }
6
7 class TestStudent{
8     public static void main(String args[]){
9         Student s1=new Student();
10        Student s2=new Student();
11        s1.id=101;
12        s1.name="Ritul";
13        s2.id=102;
14        s2.name="Amit";
15        System.out.println(s1.id+" "+s1.name);
16        System.out.println(s2.id+" "+s2.name);
17    }
18 }
```

7. Class with Method

```
1 //Class with method
2
3 class Employee{
4     int id;
5     String name;
6     float salary;
7     void setData(int i, String n, float s) {
8         id=i;
9         name=n;
10        salary=s;
11    }
12    void getData() {
13        System.out.println(id+" "+name+" "+salary);
14    }
15 }
16 public class TestEmployee {
17     public static void main(String[] args) {
18         Employee e1=new Employee();
19         Employee e2=new Employee();
20         e1.setData(101,"Ravi",45000);
21         e2.setData(102,"Mohit",25000);
22         e1.getData();
23         e2.getData();
24     }
25 }
```

8. Parameterized constructor

```
1 //Java Program to demonstrate the use of the parameterized constructor.
2
3 class Student4{
4     int id;
5     String name;
6     //creating a parameterized constructor
7     Student4(int i,String n){
8         id = i;
9         name = n;
10    }
11    //method to display the values
12    void display(){
13        System.out.println(id+" "+name);
14    }
15    public static void main(String args[]){
16        //creating objects and passing values
17        Student4 s1 = new Student4(111,"Ritul");
18        Student4 s2 = new Student4(222,"Ravi");
19        //calling method to display the values of object
20        s1.display();
21        s2.display();
22    }
23 }
```

9. Constructor Overloading

```
1 //Java program to overload constructors
2 class Student5{
3     int id;
4     String name;
5     int age;
6     //creating two arg constructor
7     Student5(int i,String n){
8         id = i;
9         name = n;
```

```

10 }
11 //creating three arg constructor
12 Student5(int i,String n,int a){
13     id = i;
14     name = n;
15     age=a;
16 }
17 void display(){System.out.println(id+" "+name+" "+age);}
18
19 public static void main(String args[]){
20     Student5 s1 = new Student5(111,"Mohit");
21     Student5 s2 = new Student5(222,"Priyanshu",25);
22     s1.display();
23     s2.display();
24 }
25 }
```

10. Jagged Array

```

1 //Program to Jagged Array.
2
3 class Test
4 {
5     public static void main(String[] args)
6     {
7         int[][] arr = new int[2][];// Declare the array
8
9         arr[0] = new int[] { 11, 21, 56, 78 };// Initialize the array
10        arr[1] = new int[] { 42, 61, 37, 41, 59, 63 };
11
12        // Traverse array elements
13        for (int i = 0; i < arr.length; i++)
14        {
15            for (int j = 0; j < arr[i].length; j++)
16            {
17                System.out.print(arr[i][j] + " ");
18            }
19            System.out.println();
20        }
21    }
22 }
```

11. Copy constructor

```

1 //Copy constructor...
2
3 class Student6{
4     int id;
5     String name;
6     //constructor to initialize integer and string
7     Student6(int i,String n){
8         id = i;
9         name = n;
10    }
11    //constructor to initialize another object
12    Student6(Student6 s){
13        id = s.id;
14        name = s.name;
15    }
16    void display(){System.out.println(id+" "+name);}
17
18    public static void main(String args[]){
19        Student6 s1 = new Student6(111,"Krupa");
20        Student6 s2 = new Student6(s1);
21        s1.display();
22        s2.display();
23    }
24 }
```

12. Java Inheritance

```

1 //Java Inheritance Demo
2
3 class Animal{
4     void eat(){
5         System.out.println("eating....");
```

```

6         }
7     }
8     class Dog extends Animal{
9         void bark(){
10            System.out.println("barking...");
11        }
12    }
13    class BabyDog extends Dog{
14        void weep(){
15            System.out.println("weeping...");
16        }
17    }
18    class TestInheritance{
19        public static void main(String args[]){
20            BabyDog d=new BabyDog();
21            d.weep();
22            d.bark();
23            d.eat();
24        }
25    }

```

13. Method Overloading

```

1 //Method Overloading Demo...
2 class Adder{
3     static int add(int a, int b) {
4         return a+b;
5     }
6     static double add(double a, double b) {
7         return a+b;
8     }
9 }
10 class TestOverloading{
11     public static void main(String[] args){
12         System.out.println(Adder.add(11,11));
13         System.out.println(Adder.add(12.3,12.6));
14     }
15 }

```

Unit – 2

Inheritance, Java Packages

14. Constructor in Inheritance

```
1 //Constructor in Inheritance
2 class Animal{
3     Animal() {
4         System.out.println("From animal constructor");
5     }
6     void eat(){
7         System.out.println("eating....");
8     }
9     protected void finalize() {
10         System.out.println("End of animal");
11     }
12 }
13 class Dog extends Animal{
14     Dog() {
15         System.out.println("From dog constructor");
16     }
17     void bark(){
18         System.out.println("barking...");
19     }
20     protected void finalize() {
21         System.out.println("End of dog");
22     }
23 }
24 }
25 class BabyDog extends Dog{
26     BabyDog() {
27         System.out.println("From babydog constructor");
28     }
29     void weep(){
30         System.out.println("weeping...");
31     }
32     protected void finalize() {
33         System.out.println("End of babydog");
34     }
35 }
36 }
37 class TestInheritance2{
38     public static void main(String args[]){
39         BabyDog d=new BabyDog();
40         d.weep();
41         d.bark();
42         d.eat();
43         d=null;
44         System.gc();
45     }
46 }
```

15. Abstract Class

```
1 //abstract class demo.
2
3 abstract class Shape{
4     abstract void draw();
5 }
6
7 class Rectangle extends Shape{
8     void draw(){System.out.println("drawing rectangle");}
9 }
10 class Circle extends Shape{
```

```

11     void draw(){System.out.println("drawing circle");}
12 }
13
14 class TestAbstraction{
15     public static void main(String args[]){
16         Shape s1=new Circle();
17         Shape s2=new Rectangle();
18         s1.draw();
19         s2.draw();
20     }
21 }
```

16. Final Class

```

1 //Final Class
2
3 final class ParentClass
4 {
5     void showData()
6     {
7         System.out.println("This is a method of final Parent class");
8     }
9 }
10
11 //It will throw compilation error
12 class ChildClass extends ParentClass
13 {
14     void showData()
15     {
16         System.out.println("This is a method of Child class");
17     }
18 }
19 class MainClass
20 {
21     public static void main(String arg[])
22     {
23         ParentClass obj = new ChildClass();
24         obj.showData();
25     }
26 }
```

17. Java Interface

```

1 //Interface Demo...
2 interface Animal {
3     public void eat();
4     public void travel();
5 }
6
7 class MammalInt implements Animal {
8
9     public void eat() {
10         System.out.println("Mammal eats");
11     }
12
13     public void travel() {
14         System.out.println("Mammal travels");
15     }
16
17     public int noOfLegs() {
18         return 0;
19     }
20
21 }
22 }
```

```

23 public class Main {
24     public static void main(String args[]) {
25         MammalInt m = new MammalInt();
26         m.eat();
27         m.travel();
28     }
29 }
```

18. Inner Class

```

1 //Inner class demo.
2
3 class Main {
4     private int data=30;
5     class Inner{
6         void msg(){System.out.println("data is "+data);}
7     }
8     public static void main(String args[]){
9         Main obj=new Main();
10        Main.Inner in=obj.new Inner();
11        in.msg();
12    }
13 }
```

19. util.Date class

```

1 import java.util.Date;
2
3 public class Main {
4
5     public static void main(String args[]) {
6
7         Date date = new Date();
8
9         System.out.println(date.toString());
10    }
11 }
```

20. Java Wrapper Classes

```

1 //wrapper classes objects and vice-versa
2
3 public class Main {
4     public static void main(String args[]){
5         byte b=10;
6         short s=20;
7         int i=30;
8         long l=40;
9         float f=50.0F;
10        double d=60.0D;
11        char c='a';
12        boolean b2=true;
13
14        //Autoboxing: Converting primitives into objects
15        Byte byteobj=b;
16        Short shortobj=s;
17        Integer intobj=i;
18        Long longobj=l;
19        Float floatobj=f;
20        Double doubleobj=d;
21        Character charobj=c;
22        Boolean boolobj=b2;
23
24        //Printing objects
25        System.out.println("---Printing object values---");
26        System.out.println("Byte object: "+byteobj);
27        System.out.println("Short object: "+shortobj);
```

```

28     System.out.println("Integer object: "+intobj);
29     System.out.println("Long object: "+longobj);
30     System.out.println("Float object: "+floatobj);
31     System.out.println("Double object: "+doubleobj);
32     System.out.println("Character object: "+charobj);
33     System.out.println("Boolean object: "+boolobj);
34
35     //Unboxing: Converting Objects to Primitives
36     byte bytevalue=byteobj;
37     short shortvalue=shortobj;
38     int intvalue=intobj;
39     long longvalue=longobj;
40     float floatvalue=floatobj;
41     double doublevalue=doubleobj;
42     char charvalue=charobj;
43     boolean boolvalue=boolobj;
44
45     //Printing primitives
46     System.out.println("---Printing primitive values---");
47     System.out.println("byte value: "+bytevalue);
48     System.out.println("short value: "+shortvalue);
49     System.out.println("int value: "+intvalue);
50     System.out.println("long value: "+longvalue);
51     System.out.println("float value: "+floatvalue);
52     System.out.println("double value: "+doublevalue);
53     System.out.println("char value: "+charvalue);
54     System.out.println("boolean value: "+boolvalue);
55 }
56 }
```

21. Creating user defined package

```

1 //Creating user-defined package..
2
3 package mypack;
4
5 public class Simple{
6     public static void main(String args[]){
7         System.out.println("Welcome to package");
8     }
9 }
```

22. Java StringTokenizer

```

1 import java.util.StringTokenizer;
2
3 public class Simple {
4     public static void main(String args[]){
5         StringTokenizer st = new StringTokenizer("Java OOP Programming Language", " ");
6         while (st.hasMoreTokens()) {
7             System.out.println(st.nextToken());
8         }
9     }
}
```

Unit – 3

Exception Handling, Threading and Streams (Input and Output)

23. Exception Handling

```
1 //Exception Handling Demonstration
2 public class Main {
3 {
4     public static void main(String[] args) {
5         int a=10,b=0,c=0;
6         System.out.println("Start of main()");
7         try{
8             c=a/b;
9         }catch(ArithmaticException ae) {
10             System.out.println(ae);
11         }finally {
12             System.out.println("I am always there...");
13         }
14         System.out.println("Value of C:"+c);
15         System.out.println("End of main()");
16     }
17 }
```

24. Multiple catch statements

```
1 //multiple catch statements
2 public class Main {
3
4     public static void main(String[] args) {
5
6         try{
7             int a[]={};
8             a[5]=30/0;
9         }
10        catch(ArithmaticException e)
11        {
12            System.out.println("Arithmatic Exception occurs");
13        }
14        catch(ArrayIndexOutOfBoundsException e)
15        {
16            System.out.println("ArrayIndexOutOfBoundsException occurs");
17        }
18        catch(Exception e)
19        {
20            System.out.println("Parent Exception occurs");
21        }
22     }
23 }
24 }
```

24. Custom exception

```
1 //Custom exception example...
2 class InvalidAgeException extends Exception{
3     InvalidAgeException(String s){
4         super(s);
5     }
6 }
7 class Main {
8
9     static void validate(int age)throws InvalidAgeException{
10         if(age<18)
11             throw new InvalidAgeException("not valid");
12         else
13             System.out.println("welcome to vote");
14     }
15
16     public static void main(String args[]){
17         try{
18             validate(13);
19         }catch(Exception m){System.out.println("Exception occurred: "+m);}
20 }
```

```

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

22 }  

23 }
```

25. Multithreading using Thread Class

```

1  public class ThreadDemo1 {  

2  

3      public static void main(String[] args) {  

4          System.out.println("Start of main");  

5          MyThread1 mt1 = new MyThread1();  

6          MyThread2 mt2 = new MyThread2();  

7          mt1.start();  

8          mt2.start();  

9          System.out.println("End of main");  

10     }  

11 }  

12  

13 class MyThread1 extends Thread{  

14     public void run(){  

15         for(int i=1;i<=10;i++) {  

16             System.out.println("MyThread-1."+i);  

17         }  

18     }  

19 }  

20  

21 class MyThread2 extends Thread{  

22     public void run(){  

23         for(int i=1;i<=10;i++) {  

24             System.out.println("MyThread-2."+i);  

25         }  

26     }  

27 }
```

26. Multithreading using Runnable interface

```

1  public class ThreadDemo2 {  

2      public static void main(String[] args) {  

3          System.out.println("Start of main");  

4          MyThread mt = new MyThread();  

5          Thread t1 = new Thread(mt,"Thread-1");  

6          Thread t2 = new Thread(mt,"Thread-2");  

7          t1.start();  

8          t2.start();  

9          System.out.println("End of main");  

10     }  

11 }  

12  

13 class MyThread implements Runnable {  

14     public void run() {  

15         for(int i=1;i<=10;i++) {  

16             System.out.println(Thread.currentThread().getName()+"."+i);  

17         }  

18     }  

19 }
```

27. Thread Scheduling

```

1  public class ThreadDemo3 {  

2  

3      public static void main(String[] args) {  

4          System.out.println("Start of main");  

5          MyThread1 mt1 = new MyThread1();  

6          MyThread2 mt2 = new MyThread2();  

7          mt1.start();  

8          mt2.start();  

9          System.out.println("End of main");  

10     }  

11 }  

12  

13 }
```

```

14 class MyThread1 extends Thread{
15     public void run(){
16         for(int i=1;i<=10;i++) {
17             System.out.println("MyThread-1."+i);
18             Thread.yield();
19         }
20     }
21 }
22
23 class MyThread2 extends Thread{
24     public void run(){
25         for(int i=1;i<=10;i++) {
26             System.out.println("MyThread-2."+i);
27             Thread.yield();
28         }
29     }
30 }
```

28. Thread Joins

```

1 public class ThreadDemo3 {
2
3     public static void main(String[] args) {
4         try {
5             System.out.println("Start of main");
6             MyThread1 mt1 = new MyThread1();
7             MyThread2 mt2 = new MyThread2();
8             mt1.start();
9             mt1.join();
10            mt2.start();
11            mt2.join();
12            System.out.println("End of main");
13        }catch(Exception e){}
14    }
15
16 }
17
18 class MyThread1 extends Thread{
19     public void run(){
20         for(int i=1;i<=10;i++) {
21             System.out.println("MyThread-1."+i);
22             try {
23                 sleep(100);
24             }catch(Exception e){ }
25         }
26     }
27 }
28
29 class MyThread2 extends Thread{
30     public void run(){
31         for(int i=1;i<=10;i++) {
32             System.out.println("MyThread-2."+i);
33             try {
34                 sleep(200);
35             }catch(Exception e){ }
36         }
37     }
38 }
```

29. Thread Priorities

```

1 public class ThreadDemo4 {
2
3     public static void main(String[] args) {
4         System.out.println("Start main");
5         MyThread mt = new MyThread();
6         Thread t1 = new Thread(mt,"Thread-1");
7         Thread t2 = new Thread(mt,"Thread-2");
8         t1.start();
9         t2.start();
10        t2.setPriority(t1.getPriority()+5);
11        System.out.println("End main");
12    }
13
14 }
15 }
```

```

16 class MyThread implements Runnable {
17     public void run() {
18         for(int i = 1;i<=10;i++) {
19             System.out.println(Thread.currentThread().getName());
20         }
21     }
22 }
```

30. File Class

```

1 import java.io.*;
2
3 public class IODemo1 {
4
5     public static void main(String[] args) {
6
7         try {
8             File f = new File("abc.txt");
9             if(f.createNewFile()) {
10                 System.out.println("File Sucessfully created");
11             }
12             else {
13                 System.out.println("File already exist");
14             }
15             System.out.println("File name : "+f.getName());
16             System.out.println("Path: "+f.getPath());
17             System.out.println("Absolute path: " +f.getAbsolutePath());
18             System.out.println("Parent: "+f.getParent());
19             System.out.println("Exists : "+f.exists());
20             System.out.println("Is writable: "+f.canWrite());
21             System.out.println("Is readable: "+f.canRead());
22             System.out.println("Is a directory: "+f.isDirectory());
23             System.out.println("File Size in bytes: "+f.length());
24         }catch(Exception e){
25             System.out.println(e);
26         }
27     }
28 }
```

31. Bytestream Class to read file

```

1 import java.io.*;
2
3 public class IODemo3 {
4     public static void main(String[] args) {
5         System.out.println("Content of output.txt file:\n");
6         try{
7             FileInputStream fin = new FileInputStream("output.txt");
8             int c;
9
10            while((c=fin.read())!= -1 ){
11                System.out.print((char)c);
12            }
13        }catch(Exception e) { }
14    }
15 }
```

32. Bytestream Class to create file

```

1 import java.io.*;
2
3 public class IODemo2 {
4
5     public static void main(String[] args) {
6         try{
7             //DataInputStream out = new DataInputStream(System.in);
8             BufferedInputStream out = new BufferedInputStream(System.in);
9             FileOutputStream fout = new FileOutputStream("output.txt");
10            System.out.println("Enter text (enter & to end): &");
11            int ch;
12            while ((ch = (char) out.read()) != '&')
13                fout.write((char)ch);
14            fout.close();
15        }catch(Exception e){}
16    }
17 }
```

33. Character stream Class to read and write file

```
1 import java.io.File;
2 import java.io.FileReader;
3 import java.io.FileWriter;
4 import java.io.IOException;
5 public class IOStreamsExample {
6     public static void main(String args[]) throws IOException {
7         //Creating FileReader object
8         File file = new File("D:/myFile.txt");
9         FileReader reader = new FileReader(file);
10        char chars[] = new char[(int) file.length()];
11        //Reading data from the file
12        reader.read(chars);
13        //Writing data to another file
14        File out = new File("D:/CopyOfmyfile.txt");
15        FileWriter writer = new FileWriter(out);
16        //Writing data to the file
17        writer.write(chars);
18        writer.flush();
19        System.out.println("Data successfully written in the specified file");
20    }
21 }
```

Unit – 4

JavaFx Basics and Event-driven programming and animations

34. JavaFX HelloWorld

```
1 //HelloWorld
2 package Unit4;
3
4 import javafx.application.Application;
5 import javafx.event.ActionEvent;
6 import javafx.event.EventHandler;
7 import javafx.scene.Scene;
8 import javafx.scene.control.Button;
9 import javafx.scene.layout.StackPane;
10 import javafx.stage.Stage;
11
12
13 public class FirstApp extends Application {
14
15     @Override
16     public void start(Stage primaryStage) {
17         Button btn = new Button();
18         btn.setText("Say 'Hello World'");
19         btn.setOnAction(new EventHandler<ActionEvent>() {
20
21             @Override
22             public void handle(ActionEvent event) {
23                 System.out.println("Hello World!");
24             }
25         });
26
27         StackPane root = new StackPane();
28         root.getChildren().add(btn);
29         Scene scene = new Scene(root, 300, 250);
30         primaryStage.setTitle("Hello World!");
31         primaryStage.setScene(scene);
32         primaryStage.show();
33     }
34
35     public static void main(String[] args) {
36         launch(args);
37     }
38
39 }
```

35. Text and Font Example

```
1 package Unit4;
2 import javafx.application.Application;
3 import javafx.scene.Scene;
4 import javafx.scene.layout.StackPane;
5 import javafx.scene.paint.Color;
6 import javafx.scene.text.Font;
7 import javafx.scene.text.FontPosture;
8 import javafx.scene.text.FontWeight;
9 import javafx.scene.text.Text;
10 import javafx.stage.Stage;
11
12 public class TextExample extends Application {
13
14     @Override
15     public void start(Stage primaryStage) {
16
17         Text text = new Text();
18         text.setText("Hello !! Welcome to JavaFX");
```

```

19     text.setFill(Color.BLUE);
20     text.setStrokeWidth(2);
21     text.setStroke(Color.RED);
22     text.setFont(Font.font("Comic Sans
23         MS", FontWeight.BOLD, FontPosture.REGULAR, 30));
24     StackPane root = new StackPane();
25     root.getChildren().add(text);
26
27     Scene scene = new Scene(root, 600, 400);
28
29     primaryStage.setTitle("Text Example");
30     primaryStage.setScene(scene);
31     primaryStage.show();
32 }
33
34 public static void main(String[] args) {
35     launch(args);
36 }
```

36. 2D Shape Example

```

1 package Unit4;
2 import javafx.application.Application;
3 import javafx.scene.Group;
4 import javafx.scene.Scene;
5 import javafx.scene.paint.Color;
6 import javafx.scene.shape.Circle;
7 import javafx.scene.shape.Rectangle;
8 import javafx.stage.Stage;
9
10 public class ShapeExample extends Application {
11     @Override
12     public void start(Stage primaryStage) {
13         Group group = new Group();
14         Rectangle rect = new Rectangle();
15         rect.setX(10);
16         rect.setY(10);
17         rect.setWidth(100);
18         rect.setHeight(100);
19         rect.setArcHeight(35);
20         rect.setArcWidth(35);
21         rect.setFill(Color.RED);
22
23         Circle circle = new Circle();
24         circle.setCenterX(300);
25         circle.setCenterY(200);
26         circle.setRadius(100);
27         circle.setFill(Color.GREEN);
28
29         group.getChildren().addAll(rect,circle);
30         Scene scene = new Scene(group, 600, 400);
31         primaryStage.setTitle("2D Shape Example");
32         primaryStage.setScene(scene);
33         primaryStage.show();
34     }
35
36     public static void main(String[] args) {
37         launch(args);
38     }
39 }
40 }
```

37. 3D Shape Example

```
1 package Unit4;
2 import javafx.application.Application;
3 import javafx.scene.Group;
4 import javafx.scene.PerspectiveCamera;
5 import javafx.scene.Scene;
6 import javafx.scene.paint.Color;
7 import javafx.scene.shape.Box;
8 import javafx.scene.shape.Cylinder;
9 import javafx.stage.Stage;
10
11 public class Shapes3DExample extends Application {
12     @Override
13     public void start(Stage primaryStage) throws Exception {
14         // TODO Auto-generated method stub
15         //Creating Boxes
16         Box box1 = new Box();
17
18         //Setting properties for first box
19         box1.setHeight(100);
20         box1.setWidth(100);
21         box1.setDepth(400);
22         box1.setTranslateX(200);
23         box1.setTranslateY(200);
24         box1.setTranslateZ(200);
25         Cylinder cyn = new Cylinder();
26
27         //setting the radius and height for the cylinder
28         cyn.setRadius(80);
29         cyn.setHeight(200);
30         cyn.setTranslateX(400);
31         cyn.setTranslateY(250);
32
33         //Setting the perspective camera
34         PerspectiveCamera camera = new PerspectiveCamera();
35         camera.setTranslateX(100);
36         camera.setTranslateY(100);
37         camera.setTranslateZ(50);
38
39         //Configuring Group, Scene and Stage
40         Group root = new Group();
41         root.getChildren().addAll(box1, cyn);
42         Scene scene = new Scene(root, 450, 350, Color.LIMEGREEN);
43         scene.setCamera(camera);
44         primaryStage.setScene(scene);
45         primaryStage.setTitle("3DShape Example");
46         primaryStage.show();
47     }
48
49     public static void main(String[] args) {
50         launch(args);
51     }
52 }
```

38. Grid Layout Example

```
1 package Unit4;
2 import javafx.application.Application;
3 import javafx.scene.Scene;
4 import javafx.scene.control.Button;
5 import javafx.scene.control.Label;
6 import javafx.scene.control.TextField;
```

```

7 import javafx.scene.layout.GridPane;
8 import javafx.stage.Stage;
9
10 public class GridPaneExample extends Application {
11     @Override
12     public void start(Stage primaryStage) throws Exception {
13         Label first_name=new Label("First Name");
14         Label last_name=new Label("Last Name");
15         TextField tf1=new TextField();
16         TextField tf2=new TextField();
17         Button Submit=new Button ("Submit");
18         GridPane root=new GridPane();
19         Scene scene = new Scene(root,400,200);
20         root.addRow(0, first_name,tf1);
21         root.addRow(1, last_name,tf2);
22         root.addRow(2, Submit);
23         primaryStage.setTitle("GridPane Layout");
24         primaryStage.setScene(scene);
25         primaryStage.show();
26     }
27     public static void main(String[] args) {
28         launch(args);
29     }
30 }
```

39. Image Input Example

```

1 package Unit4;
2 import javafx.application.Application;
3 import javafx.scene.Group;
4 import javafx.scene.Scene;
5 import javafx.scene.effect.ImageInput;
6 import javafx.scene.image.Image;
7 import javafx.scene.paint.Color;
8 import javafx.scene.shape.Rectangle;
9 import javafx.stage.Stage;
10
11 public class ImageInputExample extends Application {
12     @Override
13     public void start(Stage primaryStage) throws Exception {
14         // TODO Auto-generated method stub
15         Image img = new
16             Image("https://upload.wikimedia.org/wikipedia/commons/e/e3/Animhorse.gif
17         ");
18         ImageInput imginput = new ImageInput();
19         Rectangle rect = new Rectangle();
20         imginput.setSource(img);
21         imginput.setX(20);
22         imginput.setY(100);
23         Group root = new Group();
24         rect.setEffect(imginput);
25         root.getChildren().add(rect);
26         Scene scene = new Scene(root, 530, 500, Color.BLACK);
27         primaryStage.setScene(scene);
28         primaryStage.setTitle("ImageInput Example");
29         primaryStage.show();
30     }
31     public static void main(String[] args) {
32         launch(args);
33     }
34 }
```

40. Animation Example

```
1 package Unit4;
2 import javafx.animation.FadeTransition;
3 import javafx.application.Application;
4 import javafx.scene.Group;
5 import javafx.scene.Scene;
6 import javafx.scene.paint.Color;
7 import javafx.scene.shape.Circle;
8 import javafx.stage.Stage;
9 import javafx.util.Duration;
10 public class AnimationExample extends Application{
11
12     @Override
13     public void start(Stage primaryStage) throws Exception {
14         Circle cir = new Circle(250,120,80);
15         cir.setFill(Color.RED);
16         cir.setStroke(Color.BLACK);
17         FadeTransition fade = new FadeTransition();
18         fade.setDuration(Duration.millis(5000));
19         fade.setFromValue(10);
20         fade.setToValue(0.1);
21         fade.setCycleCount(1000);
22         fade.setAutoReverse(true);
23         fade.setNode(cir);
24         fade.play();
25         Group root = new Group();
26         root.getChildren().addAll(cir);
27         Scene scene = new Scene(root,500,250,Color.WHEAT);
28         primaryStage.setScene(scene);
29         primaryStage.setTitle("Translate Transition example");
30         primaryStage.show();
31     }
32     public static void main(String[] args) {
33         launch(args);
34     }
35 }
36 }
```

Unit – 5

GUI using SWING Event Handling

H. & H. B. Kotak Institute of Science,
Rajkot

41. Label Example

```
1 package Unit5;
2 import javafx.application.Application;
3 import javafx.scene.Group;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Label;
6 import javafx.scene.paint.Color;
7 import javafx.scene.text.Font;
8 import javafx.scene.text.FontPosture;
9 import javafx.scene.text.FontWeight;
10 import javafx.stage.Stage;
11 public class JavaFxLabel extends Application {
12     public void start(Stage stage) {
13         Label label = new Label("Sample label");
14         Font font = Font.font("Brush Script MT", FontWeight.BOLD,
FontPosture.REGULAR, 25);
15         label.setFont(font);
16         label.setTextFill(Color.BROWN);
17         label.setTranslateX(10);
18         label.setTranslateY(10);
19         Group root = new Group();
20         root.getChildren().add(label);
21         Scene scene = new Scene(root, 400, 300, Color.BEIGE);
22         stage.setTitle("Label Example");
23         stage.setScene(scene);
24         stage.show();
25     }
26     public static void main(String args[]){
27         launch(args);
28     }
29 }
```

42. Button and Textfield Example

```
1 package Unit5;
2 import javafx.application.Application;
3 import javafx.event.ActionEvent;
4 import javafx.event.EventHandler;
5 import javafx.geometry.Insets;
6 import javafx.scene.Scene;
7 import javafx.scene.control.Button;
8 import javafx.scene.control.Label;
9 import javafx.scene.control.TextField;
10 import javafx.scene.layout.HBox;
11 import javafx.scene.paint.Color;
12 import javafx.stage.Stage;
13
14 public class JavaFxTextfield extends Application {
15     public void start(Stage stage) {
16         TextField textField1 = new TextField();
17         TextField textField2 = new TextField();
18         Label label1 = new Label("Enter Text: ");
19         Label label2 = new Label("Entered Text: ");
20         Button btn = new Button("Click");
21         btn.setOnAction(new EventHandler<ActionEvent>() {
22             @Override
23             public void handle(ActionEvent arg0) {
24                 // TODO Auto-generated method stub
25                 textField2.setText(textField1.getText());
26             }
27         });
28     }
29 }
```

```

28         HBox box = new HBox(5);
29         box.setPadding(new Insets(25, 5, 5, 50));
30         box.getChildren().addAll(label1, textField1, label2, textField2,
31         btn);
32         //Setting the stage
33         Scene scene = new Scene(box, 595, 150, Color.BEIGE);
34         stage.setTitle("Text Field Example");
35         stage.setScene(scene);
36         stage.show();
37     }
38
39     public static void main(String args[]) {
40         launch(args);
41     }

```

43. ListView Example

```

1 package Unit5;
2
3 import javafx.application.Application;
4 import javafx.collections.FXCollections;
5 import javafx.collections.ObservableList;
6 import javafx.event.ActionEvent;
7 import javafx.event.EventHandler;
8 import javafx.geometry.Insets;
9 import javafx.scene.Scene;
10 import javafx.scene.control.Button;
11 import javafx.scene.control.Label;
12 import javafx.scene.control.ListView;
13 import javafx.scene.layout.VBox;
14 import javafx.scene.text.Font;
15 import javafx.scene.text.FontPosture;
16 import javafx.scene.text.FontWeight;
17 import javafx.stage.Stage;
18
19 public class JavafxListview extends Application {
20     public void start(Stage stage) {
21         Label label = new Label("Educational qualification:");
22         Label label2 = new Label();
23         Font font = Font.font("verdana", FontWeight.BOLD,
24             FontPosture.REGULAR, 12);
25         label.setFont(font);
26         ObservableList<String> names =
27             FXCollections.observableArrayList("Engineering", "MCA", "MBA",
28             "Graduation", "MTECH", "Mphil", "Phd");
29         ListView<String> listView = new ListView<String>(names);
30         listView.setMaxSize(200, 160);
31
32         Button btn = new Button("Select Course");
33         btn.setOnAction(new EventHandler<ActionEvent>() {
34             @Override
35             public void handle(ActionEvent arg0) {
36                 // TODO Auto-generated method stub
37                 label2.setText(listView.getItems().toString());
38             }
39         });
40         VBox layout = new VBox(10);
41         layout.setPadding(new Insets(5, 5, 5, 50));
42         layout.getChildren().addAll(label, listView, btn, label2);
43         layout.setStyle("-fx-background-color: BEIGE");
44         //Setting the stage
45         Scene scene = new Scene(layout, 400, 300);

```

```

43         stage.setTitle("List View Example");
44         stage.setScene(scene);
45         stage.show();
46     }
47     public static void main(String args[]) {
48         launch(args);
49     }
50 }
```

44. Slider Example

```

1 package Unit5;
2 import javafx.application.Application;
3 import javafx.scene.Group;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Slider;
6 import javafx.stage.Stage;
7
8 public class SliderExample extends Application {
9     public void start(Stage stage)
10    {
11        Group root = new Group();
12        Scene scene = new Scene(root, 600, 400);
13        stage.setScene(scene);
14        stage.setTitle("Slider Sample");
15        Slider slider = new Slider();
16        root.getChildren().add(slider);
17        stage.show();
18    }
19    public static void main(String[] args)
20    {
21        launch(args);
22    }
23 }
```

45. Play Audio Example

```

1 package Unit5;
2
3 import java.io.File;
4 import javafx.application.Application;
5 import javafx.scene.media.Media;
6 import javafx.scene.media.MediaPlayer;
7 import javafx.stage.Stage;
8
9 public class PlayAudio extends Application {
10     @Override
11     public void start(Stage primaryStage) throws Exception {
12         String path = "C:\\\\Users\\\\malay\\\\Downloads\\\\test.mp3";
13         Media media = new Media(new File(path).toURI().toString());
14         MediaPlayer mediaPlayer = new MediaPlayer(media);
15         mediaPlayer.setAutoPlay(true);
16         primaryStage.setTitle("Playing Audio");
17         primaryStage.show();
18     }
19     public static void main(String[] args) {
20         launch(args);
21     }
22 }
```

45. Play Video Example

```
1 package Unit5;
2 import java.io.File;
3 import javafx.application.Application;
4 import javafx.scene.Group;
5 import javafx.scene.Scene;
6 import javafx.scene.media.Media;
7 import javafx.scene.media.MediaPlayer;
8 import javafx.scene.media.MediaView;
9 import javafx.stage.Stage;
10
11 public class PlayVideo extends Application {
12     @Override
13     public void start(Stage primaryStage) throws Exception {
14         String path = "C:\\\\Users\\\\malay\\\\Downloads\\\\test.mp4";
15         Media media = new Media(new File(path).toURI().toString());
16         MediaPlayer mediaPlayer = new MediaPlayer(media);
17         MediaView mediaView = new MediaView(mediaPlayer);
18         mediaPlayer.setAutoPlay(true);
19         Group root = new Group();
20         root.getChildren().add(mediaView);
21         Scene scene = new Scene(root, 500, 400);
22         primaryStage.setScene(scene);
23         primaryStage.setTitle("Playing video");
24         primaryStage.show();
25     }
26     public static void main(String[] args) {
27         launch(args);
28     }
29 }
```