CS-19 Programming with JAVA BCA Semester – 4



Journal

BCA Department

Page 1 of 31

BCA Department

Laboratory Certificate

This is to certify that Smt./Shri \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ has satisfactory completed BCA Semester-4 practical experiments of subject CS-19 Programming with Java during the academic year 2022-23. Her/His enrollment number is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ registered at Saurashtra University, Rajkot.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

~~Subject In-Charge Head of the Department~~

Page 2 of 31

Index

Sr. Name of Experiments Page No.

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 Priorates 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

Date of

Experiment

Date of

Supervision Remarks

| 34 | HelloWorld Applet | 23 |  |  |  |
| --- | --- | --- | --- | --- | --- |

Page 3 of 31

Index

Sr. Name of Experiments Page No.

35 Applet Life Cycle and Mouse Event

Listener 23 36 Applet Graphics 24 37 Passing Parameter in Applet 24 38 Image in Applet 25 39 Border layout 25 40 Grid layout 25 Unit-5

41 JFrame and JPanel 28 42 JButton with Event 28 43 JTextField Example 29 44 CheckBox Example 29

Date of

Experiment

Date of

Supervision Remarks

| 45 | JList Example | 30 |  |  |  |
| --- | --- | --- | --- | --- | --- |

Page 4 of 31

Unit – 1

History, Introduction and Language Basics, Classes and Objects

Page 5 of 31

1. Hello World Program

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

2. Java Variables

| 1  2  3  4  5  6  7  8  9  10  11 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | //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': |
| --- | --- |

Page 6 of 31

| 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | //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);  }  } |
| --- | --- |

Page 7 of 31

7. Class with Method

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | //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

| 1  2  3  4  5  6  7  8  9 | //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; |
| --- | --- |

Page 8 of 31

| 10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | }  //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | //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

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

Page 9 of 31

| 6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | }  }  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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | //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));  }  } |
| --- | --- |

Page 10 of 31

Unit – 2

Inheritance, Java Packages

Page 11 of 31

14. Constructor in Inheritance

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | //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

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

Page 12 of 31

| 11  12  13  14  15  16  17  18  19  20  21 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | //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;  }  } |
| --- | --- |

Page 13 of 31

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

18. Inner Class

| 1  2  3  4  5  6  7  8  9  10  11  12  13 | //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

| 1  2  3  4  5  6  7  8  9  10  11 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | //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); |
| --- | --- |

Page 14 of 31

| 28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | 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

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

22. Java StringTokenizer

| 1  2  3  4  5  6  7  8  9 | 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());  }  }  } |
| --- | --- |

Page 15 of 31

Unit – 3

Exception Handling, Threading and Streams (Input and Output)

Page 16 of 31

23. Exception Handling

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | //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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | //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);} |
| --- | --- |

Page 17 of 31

| 21  22  23 | System.out.println("rest of the code...");  }  } |
| --- | --- |

25. Multithreading using Thread Class

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13 | 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");  }    } |
| --- | --- |

Page 18 of 31

| 14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | 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");  }    } |
| --- | --- |

Page 19 of 31

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

30. File Class

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | 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

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | 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){}  }  } |
| --- | --- |

Page 20 of 31

33. Character stream Class to read and write file

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | 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"); }  } |
| --- | --- |

Page 21 of 31

Unit – 4

Applets, Layout Managers

Page 22 of 31

34. HelloWorld Applet

| 1  2  3  4  5  6  7  8  9  10 | //HelloWorld Applet.  import java.applet.Applet;  import java.awt.Graphics;  public class HelloWorldApplet extends Applet {  public void paint (Graphics g) {  g.drawString ("Hello World", 25, 50);  }  } |
| --- | --- |

35. Applet Life Cycle and Mouse Event Listener

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | import java.awt.event.MouseListener;  import java.awt.event.MouseEvent;  import java.applet.Applet;  import java.awt.Graphics;  public class ExampleEventHandling extends Applet implements MouseListener {  StringBuffer strBuffer;  public void init() {  addMouseListener(this);  strBuffer = new StringBuffer();  addItem("initializing the apple ");  }  public void start() {  addItem("starting the applet ");  }  public void stop() {  addItem("stopping the applet ");  }  public void destroy() {  addItem("unloading the applet");  }  void addItem(String word) {  System.out.println(word);  strBuffer.append(word);  repaint();  }  public void paint(Graphics g) {  // Draw a Rectangle around the applet's display area. g.drawRect(0, 0,  getWidth() - 1,  getHeight() - 1);  // display the string inside the rectangle.  g.drawString(strBuffer.toString(), 10, 20);  }  public void mouseEntered(MouseEvent event) {  addItem("mouse entered! ");  } |
| --- | --- |

Page 23 of 31

| 48  49  50  51  51  52  53  54  55  56  57  58  59  60 | public void mouseExited(MouseEvent event) {  addItem("mouse exit! ");  }  public void mousePressed(MouseEvent event) {  }  public void mouseReleased(MouseEvent event) {  }  public void mouseClicked(MouseEvent event) {  addItem("mouse clicked! ");  }  } |
| --- | --- |

36. Applet Graphics

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | //Applet Graphics Demo  import java.applet.Applet;  import java.awt.Color;  import java.awt.Graphics;  public class GraphicsDemo extends Applet {  public void paint(Graphics g){  g.setColor(Color.red);  g.drawString("Welcome",50, 50);  g.setColor(Color.black);  g.drawLine(20,30,50,300);  g.drawRect(70,100,30,30);  g.setColor(Color.blue);  g.fillRect(170,100,30,30);  g.drawOval(70,200,30,30);  g.setColor(Color.pink);  g.fillOval(170,200,30,30);  g.drawArc(90,150,30,30,30,270);  g.fillArc(270,150,30,30,0,180);  }  } |
| --- | --- |

37. Passing Parameter in Applet

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | //Passing Parameter in Applet  import java.applet.Applet;  import java.awt.Graphics;  public class UseParam extends Applet{  public void paint(Graphics g){  String str=getParameter("msg");  g.drawString(str,50, 50);  }  }  /\*  <html>  <body>  <applet code="UseParam.class" width="300" height="300"> <param name="msg" value="Welcome to applet"> </applet>  </body>  </html>  \*/ |
| --- | --- |

Page 24 of 31

38. Image in Applet

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | import java.awt.\*;  import java.applet.\*;  public class DisplayImage extends Applet {  Image picture;    public void init() {  picture = getImage(getDocumentBase(),"car.jpg");  }    public void paint(Graphics g) {  g.drawImage(picture, 30,30, this);  }  } |
| --- | --- |

39. Border layout

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | package borderlayout;  import javax.swing.\*;  import java.awt.\*;  public class BorderLayoutDemo {  JFrame f;  BorderLayoutDemo() {  f=new JFrame();  JButton b1=new JButton("NORTH");  JButton b2=new JButton("SOUTH");  JButton b3=new JButton("EAST");  JButton b4=new JButton("WEST");  JButton b5=new JButton("CENTER");  f.add(b1,BorderLayout.NORTH);  f.add(b2,BorderLayout.SOUTH);  f.add(b3,BorderLayout.EAST);  f.add(b4,BorderLayout.WEST);  f.add(b5,BorderLayout.CENTER);  f.setSize(300,300);  f.setVisible(true);  }    public static void main(String[] args) {  new BorderLayoutDemo();  }  } |
| --- | --- |

40. Grid layout

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | package gridlayoutdemo;  import javax.swing.\*;  import java.awt.\*;  public class GridLayoutDemo {  GridLayoutDemo(){  JFrame f=new JFrame();  JButton b1=new JButton("1");  JButton b2=new JButton("2");  JButton b3=new JButton("3");  JButton b4=new JButton("4");  JButton b5=new JButton("5"); |
| --- | --- |

Page 25 of 31

| 15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | JButton b6=new JButton("6");  JButton b7=new JButton("7");  JButton b8=new JButton("8");  JButton b9=new JButton("9");  f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5); f.add(b6);f.add(b7);f.add(b8);f.add(b9);  f.setLayout(new GridLayout(3,3));  f.setSize(300,300);  f.setVisible(true);  }  public static void main(String[] args) {  new GridLayoutDemo();  }  } |
| --- | --- |

Page 26 of 31

Unit – 5

GUI using SWING Event Handling

Page 27 of 31

41. JFrame and JPanel

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | //JFrame and JPanel Demo.  package swing1;  import java.awt.FlowLayout;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JLabel;  import javax.swing.JPanel;  public class Swing1 {  public static void main(String[] args) {  JFrame frame = new JFrame("JFrame Example");  JPanel panel = new JPanel();  panel.setLayout(new FlowLayout());  JLabel label = new JLabel("JFrame By Example"); JButton button = new JButton();  button.setText("Button");  panel.add(label);  panel.add(button);  frame.add(panel);  frame.setSize(200, 300);  frame.setLocationRelativeTo(null);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setVisible(true);  }    } |
| --- | --- |

42. JButton with Event

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | //JButton with Event Example  package swing2;  import javax.swing.\*;  import java.awt.event.\*;  public class Swing2 {  public static void main(String[] args) {  JFrame f=new JFrame("Button Example");  final JTextField tf=new JTextField();  tf.setBounds(50,50, 150,20);  JButton b=new JButton("Click Here");  b.setBounds(50,100,95,30);  b.addActionListener(new ActionListener(){  public void actionPerformed(ActionEvent e){  tf.setText("Welcome to Swing in Java"); }  });  f.add(b);f.add(tf);  f.setSize(400,400);  f.setLayout(null);  f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); f.setVisible(true);  }  } |
| --- | --- |

Page 28 of 31

43. JTextField Example

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | //JTextField Example  package swing3;  import java.awt.event.\*;  import javax.swing.\*;  public class Swing3 implements ActionListener {    JTextField tf1,tf2,tf3;  JButton b1,b2;  Swing3(){  JFrame f= new JFrame("Calculator");  tf1=new JTextField();  tf1.setBounds(50,50,150,20);  tf2=new JTextField();  tf2.setBounds(50,100,150,20);  tf3=new JTextField();  tf3.setBounds(50,150,150,20);  tf3.setEditable(false);  b1=new JButton("+");  b1.setBounds(50,200,50,50);  b2=new JButton("-");  b2.setBounds(120,200,50,50);  b1.addActionListener(this);  b2.addActionListener(this);  f.add(tf1);f.add(tf2);f.add(tf3);f.add(b1);f.add(b2); f.setSize(300,300);  f.setLayout(null);  f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); f.setVisible(true);  }  public void actionPerformed(ActionEvent e) {  String s1=tf1.getText();  String s2=tf2.getText();  int a=Integer.parseInt(s1);  int b=Integer.parseInt(s2);  int c=0;  if(e.getSource()==b1){  c=a+b;  }else if(e.getSource()==b2){  c=a-b;  }  String result=String.valueOf(c);  tf3.setText(result);  }  public static void main(String[] args) {  new Swing3();  }  } |
| --- | --- |

44. CheckBox Example

| 1  2  3  4  5  6  7  8  9 | //CheckBox Example  package swing4;  import javax.swing.\*;  import java.awt.event.\*;  public class Swing4 extends JFrame implements ActionListener { |
| --- | --- |

Page 29 of 31

| 10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | JLabel l;  JCheckBox cb1,cb2,cb3;  JButton b;  Swing4(){  this.setTitle("Cafeteria");  l=new JLabel("Food Ordering System");  l.setBounds(50,50,300,20);  cb1=new JCheckBox("Pizza @ 100");  cb1.setBounds(100,100,150,20);  cb2=new JCheckBox("Burger @ 30");  cb2.setBounds(100,150,150,20);  cb3=new JCheckBox("Tea @ 10");  cb3.setBounds(100,200,150,20);  b=new JButton("Order");  b.setBounds(100,250,80,30);  b.addActionListener(this);  add(l);add(cb1);add(cb2);add(cb3);add(b);  setSize(400,400);  setLayout(null);  setVisible(true);  setDefaultCloseOperation(EXIT\_ON\_CLOSE);  }  public void actionPerformed(ActionEvent e){  float amount=0;  String msg="";  if(cb1.isSelected()){  amount+=100;  msg="Pizza: 100\n";  }  if(cb2.isSelected()){  amount+=30;  msg+="Burger: 30\n";  }  if(cb3.isSelected()){  amount+=10;  msg+="Tea: 10\n";  }  msg+="-----------------\n";  JOptionPane.showMessageDialog(this,msg+"Total: "+amount); }  public static void main(String[] args) {  new Swing4();  }    } |
| --- | --- |

45. JList Example

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | //JList Demo.  package swing5;  import javax.swing.\*;  import java.awt.event.\*;  import static javax.swing.JFrame.EXIT\_ON\_CLOSE;  public class Swing5 {    Swing5(){  JFrame f= new JFrame();  final JLabel label = new JLabel();  label.setSize(500,100);  JButton b=new JButton("Show"); |
| --- | --- |

Page 30 of 31

| 15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | b.setBounds(200,150,80,30);  final DefaultListModel<String> l1 = new DefaultListModel<>(); l1.addElement("C");  l1.addElement("Python");  l1.addElement("Java");  l1.addElement("PHP");  final JList<String> list1 = new JList<>(l1);  list1.setBounds(100,100, 75,75);  DefaultListModel<String> l2 = new DefaultListModel<>(); l2.addElement("DJango");  l2.addElement("Struts");  l2.addElement("Spring");  l2.addElement("Larawel");  final JList<String> list2 = new JList<>(l2);  list2.setBounds(100,200, 75,75);  f.add(list1); f.add(list2); f.add(b); f.add(label); f.setSize(450,450);  f.setLayout(null);  f.setVisible(true);  f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);  b.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) { String data = "";  if (list1.getSelectedIndex() != -1) {  data = "Programming language Selected: " + list1.getSelectedValue();  label.setText(data);  }  if(list2.getSelectedIndex() != -1){  data += ", FrameWork Selected: ";  for(Object frame :list2.getSelectedValues()){ data += frame + " ";  }  }  label.setText(data);  }  });  }  public static void main(String[] args) {  new Swing5();  }    } |
| --- | --- |

Page 31 of 31