SEARCHING AND SORTING ALGORITHMS

LINEAR SEARCH PROGRAM:

Sequential.java

```
package exlcode;
public class Sequential
{
   public static void main(String[] args)
       {
         int[] exampleVariableOne = {2, 9, 6, 7, 4, 5, 3, 0, 1};
         int target = 4;
         sequentialSearch(exampleVariableOne, target);
       }
    public static void sequentialSearch(int[] parameterOne, int parameterTwo)
      {
         int index = -1;
         for (int i = 0; i < parameterOne.length; i++)
         {
          if (parameterOne[i] == parameterTwo)
              {
                index = i;
                break;
              }
          }
   if (index == -1)
          {
             System.out.println("Your target integer does not exist in the array");
          }
```

```
  else
        {
           System.out.println("Your target integer is in index " + index + " of the array");
        }
  }
}
```

BINARY SEARCH PROGRAM:

```
package exlcode;
public class BinarySearchExample {
 public static void main(String[] args) {
  int[] exampleVariableOne = {1, 11, 24, 34, 67, 89, 102};
  int target = 102;
  binarySearch(exampleVariableOne, target);
 }
 public static void binarySearch(int[] parameterOne, int parameterTwo) {
  int index = -1;
  int lowEnd = 0;
  int highEnd = parameterOne.length - 1;
  while (highEnd >= lowEnd) {
   int middle = (lowEnd + highEnd) / 2;
   if (parameterOne[middle] == parameterTwo) {
    index = middle; // the target is found
    break;
   } else if (parameterOne[middle] < parameterTwo) {
    lowEnd = middle + 1;
   } else if (parameterOne[middle] > parameterTwo) {
    highEnd = middle - 1;
   }
  }
  if (index == -1) {
   System.out.println("Your target integer does not exist in the array:");
  }
 else
```

```
{
 System.out.println("Your  target integer  is in index  " + index  + " of the array");
    }
  }
```

SEQUENTIAL SEARCH PROGRAM:

```
package exlcode;
public class SequentialSearchExample {
  public static void main(String[] args) {
    int[] exampleVariableOne = {2, 9, 6, 7, 4, 5, 3, 0, 1};
    int target = 4;
    sequentialSearch(exampleVariableOne, target);
  }
  public static void sequentialSearch(int[] parameterOne, int parameterTwo) {
    int index = -1;
    for (int i = 0; i < parameterOne.length; i++) {
      if (parameterOne[i] == parameterTwo) {
      index = i;
        break;
      }
    }
    if (index == -1) {
      System.out.println("Your target integer does not exist in the array");
    } else {
      System.out.println("Your target integer is in index " + index + " of the array");
    }
  }
}
```

SELECTION SORT:

```
package exlcode;
public class SelectionSortExample {
public static void main(String[] args) {
  int[] exampleVariableOne = {17, 5, 21, 8, 19, 2, 23, 15, 4, 13};
  selectionSort(exampleVariableOne);
  System.out.println("Sorted Values: ");
  for (int val : exampleVariableOne) {
    System.out.print(val + " ");
  }
}
public static void selectionSort(int[] paraImeterOne) {
  for (int i = 0; i < parameterOne.length - 1; i++) {
    int min = i;
    for (int j = i + 1; j < parameterOne.length; j++) {
      if (parameterOne[j] < parameterOne[min]) {
        min = j;
      }
    } // finds the smallest value in the array and swaps it with     // the value at index 0
    // the process continues until the array is sorted
    int temp = parameterOne[i];
    parameterOne[i] = parameterOne[min];
    parameterOne[min] = temp;
  }
}}
```

INSERTING SORT:

```
package exlcode;
public class InsertionSortExample {
  public static void main(String[] args) {
    int[] exampleVariableOne = {17, 5, 21, 8, 19, 2, 23, 15, 4, 13};
    insertionSort(exampleVariableOne);
    System.out.println("Sorted Values: ");
    for (int val : exampleVariableOne) {
      System.out.print(val + " ");
    }
  }
  public static void insertionSort(int[] parameterOne) {
    for (int j = 1; j < parameterOne.length; j++) {
      int k = j;
      while (k > 0 && parameterOne[k - 1] > parameterOne[k]) {
        int temp = parameterOne[k - 1];
        parameterOne[k - 1] = parameterOne[k];
        parameterOne[k] = temp;
        k--;
      }
    }
```

STACK AND QUEUE DATA STRUCTURES USING CLASSES AND OBJECTS

STACK IMPLEMENTATION

PROGRAM:

```
class Stack
{
   private int arr[];
   private int top;
   private int capacity;
   Stack(int size)
   {
      arr = new int[size];
      capacity = size;
      top = -1;
   }
   public void push(int x)
   {
      if (isFull())
      {
         System.out.println("Overflow\nProgram Terminated\n");
         System.exit(-1);
      }

      System.out.println("Inserting " + x);
      arr[++top] = x;
   }
   public int pop()
   {
      if (isEmpty())
      {
         System.out.println("Underflow\nProgram Terminated");
         System.exit(-1);
      }

      System.out.println("Removing " + peek());
       return arr[top--];
   }
// Utility function to return the top element of the stack
   public int peek()
   {
      if (!isEmpty()) {
         return arr[top];
```

```
 }
       else {
          System.exit(-1);
       }
 return -1;
    }
  // Utility function to return the size of the stack
    public int size() {
       return top + 1;
    }
 // Utility function to check if the stack is empty or not
    public boolean isEmpty() {
       return top == -1;              // or return size() == 0;
    }
  // Utility function to check if the stack is full or not
    public boolean isFull() {
       return top == capacity - 1;     // or return size() == capacity;
    }
}
 class Main
{
    public static void main (String[] args)
    {
       Stack stack = new Stack(3);
stack.push(1);       // inserting 1 in the stack
       stack.push(2);       // inserting 2 in the stack
 stack.pop();        // removing the top element (2)
       stack.pop();        // removing the top element (1)
  stack.push(3);       // inserting 3 in the stack
 System.out.println("The top element is " + stack.peek());
       System.out.println("The stack size is " + stack.size());
 stack.pop();        // removing the top element (3)
 // check if the stack is empty
       if (stack.isEmpty()) {
          System.out.println("The stack is empty");
       }
       else {
          System.out.println("The stack is not empty");
       }
    }
}
```

QUEUE IMPLEMENTATION

PROGRAM:

```
class Queue
{
   private int[] arr;      // array to store queue elements
   private int front;      // front points to the front element in the queue
   private int rear;       // rear points to the last element in the queue
   private int capacity;   // maximum capacity of the queue
   private int count;      // current size of the queue

   // Constructor to initialize a queue
   Queue(int size)
   {
      arr = new int[size];
      capacity = size;
      front = 0;
      rear = -1;
      count = 0;
   }

   // Utility function to dequeue the front element
   public int dequeue()
   {
      // check for queue underflow
      if (isEmpty())
      {
         System.out.println("Underflow\nProgram Terminated");
         System.exit(-1);
      }
int x = arr[front];
  System.out.println("Removing " + x);
front = (front + 1) % capacity;
      count--;
return x;
   }
// Utility function to add an item to the queue
   public void enqueue(int item)
   {
      // check for queue overflow
      if (isFull())
      {
         System.out.println("Overflow\nProgram Terminated");
         System.exit(-1);
```

```
        }
 System.out.println("Inserting  " + item);

        rear = (rear + 1) % capacity;
        arr[rear] = item;
        count++;
    }
 // Utility  function  to return the front element  of the queue
    public  int  peek()
    {
        if (isEmpty())
        {
            System.out.println("Underflow\nProgram  Terminated");
            System.exit(-1);
        }
        return arr[front];
    }
 // Utility  function  to return the size  of the queue
    public  int  size() {
        return count;
    }
 // Utility  function  to check if the queue is empty  or not
    public  boolean isEmpty()  {
        return (size()  == 0);
    }
 // Utility  function  to check if the queue is full  or not
    public  boolean isFull()  {
        return (size()  == capacity);
    }
}
 class Main
{
    public  static  void main (String[]  args)
    {
        // create a queue of capacity 5
        Queue q = new Queue(5);
 q.enqueue(1);
    q.enqueue(2);
        q.enqueue(3);
System.out.println("The  front element  is " + q.peek());
        q.dequeue();
        System.out.println("The  front element  is " + q.peek());
         System.out.println("The  queue size  is " + q.size());
```

```
        q.dequeue();
        q.dequeue();
  if (q.isEmpty())  {
            System.out.println("The  queue is empty");
        }
        else {
            System.out.println("The  queue is not empty");
        }
    }
}
```

INHERITANCE IN JAVA

PROGRAM:

```
import java.util.Scanner;
class Employee{
String Emp_name;
int Emp_id;
String Address;
String Mail_id;
int Mobile_no;
void display(){
System.out.println(Emp_name);
//Syetem.out.println(Address);
System.out.println(Emp_id);
System.out.println(Mail_id);
System.out.println(Mobile_no);
}
}
class Programmer extends Employee{
 int BP;
/*int DA= (int) (0.97*BP);
 HRA=(int) (0.10*BP);
 PF=(int) (0.12*BP); */
void display(){
 System.out.println(BP);
 System.out.println("DA"+0.97*BP);
 System.out.println("HRA"+0.10*BP);
 System.out.println("PF"+0.12*BP);
 System.out.println("SATFF CLUD FUND"+0.001*BP);

}
}
class Assistant_Professor extends Employee{
 int BP;
void display(){
```

```
 System.out.println(BP);
 System.out.println("DA"+0.97*BP);
 System.out.println("HRA"+0.10*BP);
 System.out.println("PF"+0.12*BP);
 System.out.println("SATFF  CLUD FUND"+0.001*BP);
}
}
class Associate_Professor  extends Employee{
 int BP;
void display(){
 System.out.println(BP);
 System.out.println("DA"+0.97*BP);
 System.out.println("HRA"+0.10*BP);
 System.out.println("PF"+0.12*BP);
 System.out.println("SATFF  CLUD FUND"+0.001*BP);
}
}
class Professor extends Employee{
 int BP;
void display(){
 System.out.println(BP);
 System.out.println("DA"+0.97*BP);
 System.out.println("HRA"+0.10*BP);
 System.out.println("PF"+0.12*BP);
 System.out.println("SATFF  CLUD FUND"+0.001*BP);
}
}
class Main{
public static void main(String args[]){
System.out.println("\n1.Programmer\n2.Assistant_Professor\n3.Associate_Professor\n4.Profe
ssor");
Scanner input=new  Scanner(System.in);
System.out.print("Enter  an integer:  ");
int ch=input.nextInt();
```

```
switch (ch) {
        case 1:
          Employee  e1=new Employee();
          Programmer  p1=new Programmer();
          e1.Emp_name="ABC";
          e1.Address="y-city";
          e1.Mail_id="praw@gmail.com";
          e1.Emp_id=567;
          e1.Mobile_no=2345678;
          p1.BP=15000;
          p1.display();
             e1.display();
             break;
        case 2:
          Employee  e2=new Employee();
          Assistant_Professor  p2=new Assistant_Professor();
          e2.Emp_name="DEF";
          e2.Address="A-city";
          e2.Mail_id="RAJAN@gmail.com";
          e2.Emp_id=123;
          e2.Mobile_no=987321;
          p2.BP=30000;
          p2.display();
         e2.display();
         break;
         case 3:
          Employee  e3=new Employee();
          Associate_Professor  p3=new Associate_Professor();
          e3.Emp_name="GHF";
          e3.Address="B-city";
          e3.Mail_id="MAIN@gmail.com";
          e3.Emp_id=456;
          e3.Mobile_no=98710;
          p3.BP=30000;
```

```
p3.display();
e3.display();
break;
case 4:
Employee  e4=new Employee();
Professor p4=new Professor();
e4.Emp_name="KANNAN";
e4.Address="TRICHY";
e4.Mail_id="kanna@gmail.com";
e4.Emp_id=789;
e4.Mobile_no=9810;
p4.BP=30000;
p4.display();
e4.display();
 break;
 case 5:
  //exit(1);
  default:
  System.out.println("enter  correct choice");
  } }}
```

ABSTRACT CLASS IN JAVA

PROGRAM:

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void area(double x,double y);
}
class Rectangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void area(double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
{
void area(double x,double y)
{
System.out.println("Area of triangle is :"+(0.5*x*y));
}
}
public class AbstactDDemo
{
public static void main(String[] args)
{
Rectangle r=new Rectangle();
r.area(2,5);
Circle c=new Circle();
c.area(5,5);
Triangle t=new Triangle();
t.area(2,5);
}}
```

INTERFACE IN JAVA

PROGRAM:

```
import java.lang.*;
import java.util.*;

interface Rectangle
  {
    void print_area(int x,int y);
  }
interface Circle
  {
    void print_area(int x);
  }
interface Triangle
  {
    void print_area(double x1,double y1);
  }
public class InterfaceDemo implements Rectangle,Circle,Triangle
{
  public void print_area(int x,int y)
   {
    System.out.println("Area of rectangle is:"+(x*y));
   }
  public void print_area(int x)
   {
    System.out.println("Area of circle is:"+(3.14*x*x));
   }
  public void print_area(double x1,double y1)
   {
```

```
    System.out.println("Area of triangle is:"+(0.5*x1*y1));
   }
  public static void main(String args[])
   {
   InterfaceDemo id=new InterfaceDemo();
   id.print_area(2,3);
   id.print_area(10);
   id.print_area(2.5,3.5);
   }
}
```

USER DEFINED EXCEPTION HANDLING

PROGRAM:

```
// class representing custom exception
class InvalidAgeException extends Exception
{
   public InvalidAgeException (String str)
   {
      // calling the constructor of parent Exception
      super(str);
   }
}
public class TestCustomException1
{
   static void validate (int age) throws InvalidAgeException{
     if(age < 18){
        throw new InvalidAgeException("age is not valid to vote");
   }
     else {
      System.out.println("welcome to vote");
      }
    }
       public static void main(String args[])
   {
      try
      {
                validate(13);
      }
```

```
        catch (InvalidAgeException  ex)
        {
           System.out.println("Caught  the exception");
            System.out.println("Exception  occured: " + ex);
        }
         System.out.println("rest  of the code...");
    }
}
```

MULTITHREADING IN JAVA

PROGRAM:

```
import java.util.Random;
class Square extends Thread
{
int x;
Square(int n)
{
x = n;
}
public void run()
{
int sqr = x * x;
System.out.println("Square of " + x + " = " + sqr );
}
}
class Cube extends Thread
{
int x;
Cube(int n)
{
x = n;
}
public void run()
{
int cub = x * x * x;
System.out.println("Cube of " + x + " = " + cub );
}
}
class Number extends Thread
{
public void run()
{
Random random = new Random();
for(int i =0; i<10; i++)
{
int randomInteger = random.nextInt(100);
System.out.println("Random Integer generated : " + randomInteger);
Square s = new Square(randomInteger);
s.start();
```

```
Cube c = new Cube(randomInteger);
c.start();
try {
Thread.sleep(1000);
} catch (InterruptedException ex) {
System.out.println(ex);
}
}
}
}
public class LAB3B {
public static void main(String args[])
{
Number n = new Number();
n.start();
}
}
```

FILE OPERATION

PROGRAM:

```
import java.util.Scanner;
import java.io.File;
class FileDemo
{
public static void main(String args[])
{
System.out.println("Enter the name of the file");
Scanner input=new Scanner(System.in);
String s=input.nextLine();
File f1=new File(s);
System.out.println(" ");
System.out.println("File name:"+f1.getName()); System.out.println("Path:"+f1.getPath());
System.out.println("Abs Path:"+f1.getAbsolutePath());
System.out.println("The file is:"+(f1.exists()?"Exists":"Does not Exists"));
System.out.println("Is
file:"+f1.isFile());
System.out.println("Is Directory:"+f1.isDirectory());
System.out.println("Is Readable:"+f1.canRead());
System.out.println("Is Writable:"+f1.canWrite());
System.out.println("Is Absolute:"+f1.isAbsolute());
System.out.println("File Size:"+f1.length()+"bytes");
System.out.println("Is Hidden:"+f1.isHidden());
}
}
```

GENERIC METHOD IMPLEMENTATION

PROGRAM:

```
import java.util.*;
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 GenericTest{
public static void drawShapes(List<? extends Shape> lists){
for(Shape s:lists){
s.draw();}}
public static void main(String args[]){
List<Rectangle> list1=new ArrayList<Rectangle>();
list1.add(new Rectangle());
List<Circle> list2=new ArrayList<Circle>();
list2.add(new Circle());
list2.add(new Circle());
drawShapes(list1);
drawShapes(list2);
}
}
```

JAVAFX CONTROLS, LAYOUTS AND MENUS

PROGRAM:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.*;
import javafx.stage.Stage;
import javafx.scene.control.Alert.AlertType;
import java.time.LocalDate;
public class MenuBar_2 extends Application {
   public void start(Stage s)
   {
      s.setTitle("creating MenuBar");
      Menu m = new Menu("Menu");
      MenuItem m1 = new MenuItem("menu item 1");
      MenuItem m2 = new MenuItem("menu item 2");
      MenuItem m3 = new MenuItem("menu item 3");
      m.getItems().add(m1);
      m.getItems().add(m2);
      m.getItems().add(m3);
      Label l = new Label("\t\t\t\t"  + "no menu item selected");
      EventHandler<ActionEvent> event = new EventHandler<ActionEvent>() {
         public void handle(ActionEvent e)
         {
```

```
            l.setText("\t\t\t\t"  + ((MenuItem)e.getSource()).getText()  +   " selected");
           }
      };
      m1.setOnAction(event);
      m2.setOnAction(event);
      m3.setOnAction(event);
      MenuBar  mb = new MenuBar();
      mb.getMenus().add(m);
      VBox vb = new VBox(mb,  l);
      Scene sc = new Scene(vb,  500, 300);
      s.setScene(sc);
      s.show();
   }
   public  static  void main(String  args[])
   {
      launch(args);
   }
}
```

PROGRAM :

```
package org.mano.example;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
public class LayoutDemo extends Application {
  public static void main(String[] args) {
    Application.launch(args);
  }
  @Override
  public void start(Stage stage) throws Exception {
    Scene scene = new Scene(createHBoxLayout(), 650, 100);
    stage.setTitle("Layout Demo");
    stage.setScene(scene);
    stage.show();
  }
  public HBox createHBoxLayout() {
    HBox hbox = new HBox();
    hbox.setSpacing(10);
    hbox.setPadding(new Insets(5));
```

```
        hbox.setAlignment(Pos.CENTER_LEFT);
        Label userLabel=new Label("User Name ");
        Label passLabel=new Label("Password ");
        TextField userTextField=new TextField();
        PasswordField passwordField=new PasswordField();
        Button loginButton=new Button("Login");
        hbox.getChildren().addAll(userLabel,userTextField,
          passLabel,passwordField,loginButton);
        return hbox;
    }
}
```

MINI PROJECT

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class MyCalc extends WindowAdapter implements ActionListener{
  Frame f;
Label l1;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button badd,bsub,bmult,bdiv,bmod,bcalc,bclr,bpts,bneg,bback;
double xd;
double num1,num2,check;
MyCalc(){
  f= new Frame("MY  CALCULATOR");
l1=new Label();
l1.setBackground(Color.LIGHT_GRAY);
l1.setBounds(50,50,260,60);
b1=new Button("1");
  b1.setBounds(50,340,50,50);
b2=new Button("2");
  b2.setBounds(120,340,50,50);
b3=new Button("3");
  b3.setBounds(190,340,50,50);
b4=new Button("4");
  b4.setBounds(50,270,50,50);
b5=new Button("5");
  b5.setBounds(120,270,50,50);
b6=new Button("6");
```

```
  b6.setBounds(190,270,50,50);
b7=new Button("7");
  b7.setBounds(50,200,50,50);
b8=new Button("8");
  b8.setBounds(120,200,50,50);
b9=new Button("9");
  b9.setBounds(190,200,50,50);
b0=new Button("0");
  b0.setBounds(120,410,50,50);
bneg=new  Button("+/-");
  bneg.setBounds(50,410,50,50);
bpts=new Button(".");
  bpts.setBounds(190,410,50,50);
bback=new Button("back");
 bback.setBounds(120,130,50,50);
badd=new Button("+");
  badd.setBounds(260,340,50,50);
bsub=new  Button("-");
  bsub.setBounds(260,270,50,50);
bmult=new  Button("*");
  bmult.setBounds(260,200,50,50);
bdiv=new  Button("/");
  bdiv.setBounds(260,130,50,50);
bmod=new Button("%");
  bmod.setBounds(190,130,50,50);
bcalc=new Button("=");
  bcalc.setBounds(245,410,65,50);
```

```
bclr=new  Button("CE");
  bclr.setBounds(50,130,65,50);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);
bpts.addActionListener(this);
bneg.addActionListener(this);
bback.addActionListener(this);
badd.addActionListener(this);
bsub.addActionListener(this);
bmult.addActionListener(this);
bdiv.addActionListener(this);
bmod.addActionListener(this);
bcalc.addActionListener(this);
bclr.addActionListener(this);
f.addWindowListener(this);
f.add(l1);
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.add(b0);
f.add(badd);  f.add(bsub);  f.add(bmod);  f.add(bmult);  f.add(bdiv);  f.add(bmod);f.add(bcalc);
```

```
f.add(bclr); f.add(bpts);f.add(bneg); f.add(bback);
f.setSize(360,500);
f.setLayout(null);
f.setVisible(true);
}
public void windowClosing(WindowEvent e) {
 f.dispose();
}
public void actionPerformed(ActionEvent e){
 String z,zt;
if(e.getSource()==b1){
 zt=l1.getText();
 z=zt+"1";
 l1.setText(z);
}
if(e.getSource()==b2){
zt=l1.getText();
z=zt+"2";
l1.setText(z);
}
if(e.getSource()==b3){
 zt=l1.getText();
 z=zt+"3";
 l1.setText(z);
}
if(e.getSource()==b4){
 zt=l1.getText();
```

```
  z=zt+"4";
  l1.setText(z);
}
if(e.getSource()==b5){
  zt=l1.getText();
  z=zt+"5";
  l1.setText(z);
}
if(e.getSource()==b6){
  zt=l1.getText();
  z=zt+"6";
  l1.setText(z);
}
if(e.getSource()==b7){
  zt=l1.getText();
  z=zt+"7";
  l1.setText(z);
}
if(e.getSource()==b8){
  zt=l1.getText();
  z=zt+"8";
  l1.setText(z);
}
if(e.getSource()==b9){
  zt=l1.getText();
  z=zt+"9";
  l1.setText(z);
```

```
}
if(e.getSource()==b0){
  zt=l1.getText();
  z=zt+"0";
  l1.setText(z);
}
if(e.getSource()==bpts){
  zt=l1.getText();
  z=zt+".";
  l1.setText(z);
}
if(e.getSource()==bneg){
  zt=l1.getText();
  z="-"+zt;
  l1.setText(z);
}
if(e.getSource()==bback){
  zt=l1.getText();
  try{
    z=zt.substring(0,  zt.length()-1);
    }catch(StringIndexOutOfBoundsException  f){return;}
  l1.setText(z);
}
if(e.getSource()==badd){
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException  f){
```

```
      l1.setText("Invalid  Format");
      return;
    }
  z="";
  l1.setText(z);
  check=1;
}
if(e.getSource()==bsub){
  try{
     num1=Double.parseDouble(l1.getText());
      }catch(NumberFormatException  f){
    l1.setText("Invalid  Format");
    return;
    }
  z="";
  l1.setText(z);
  check=2;
}
if(e.getSource()==bmult){
  try{
      num1=Double.parseDouble(l1.getText());
      }catch(NumberFormatException  f){
      l1.setText("Invalid  Format");
      return;
     }
  z="";
  l1.setText(z);
```

```
 check=3;
}
if(e.getSource()==bdiv){
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=4;
}
if(e.getSource()==bmod){
  try{
    num1=Double.parseDouble(l1.getText());
    }catch(NumberFormatException f){
      l1.setText("Invalid Format");
      return;
    }
  z="";
  l1.setText(z);
  check=5;
}
if(e.getSource()==bcalc){
  try{
    num2=Double.parseDouble(l1.getText());
```

```
    }catch(Exception f){
      l1.setText("ENTER NUMBER FIRST ");
      return;
    }
  if(check==1)
    xd =num1+num2;
  if(check==2)
    xd =num1-num2;
  if(check==3)
    xd =num1*num2;
  if(check==4)
    xd =num1/num2;
  if(check==5)
    xd =num1%num2;
  l1.setText(String.valueOf(xd));
}
if(e.getSource()==bclr){
  num1=0;
  num2=0;
  check=0;
  xd=0;
   z="";
   l1.setText(z);
   }
}
public static void main(String args[]){
      new MyCalc();  }}
```