COURSE OBJECTIVES AND OUTCOMES

OBJECTIVES:

The student should be made:

1. To build software development skills using java programming for real-world applications.
2. To understand and apply the concepts of classes, packages, interfaces, arraylist, exception handling and file processing.
3. To develop applications using generic programming and event handling.

LIST OF EXPERIMENTS:

IMPLEMENTATION IN THE FOLLOWING TOPICS:

1. Class and object creation.
2. Packages.
3. Inheritance.
4. Interface and exception handling.
5. String manipulation using Arraylist.
6. Abstract class.
7. User defined exception handling.
8. File processing.
9. Multithreading.
10. Generic function.
11. Event-driven programming.
12. Mini project using java concepts.

OUTCOMES:

At the end of the course, the student should be able to:

1. Develop and implement Java programs for simple applications that make use of classes, packages and interfaces.
2. Develop and implement Java programs with arraylist, exception handling and multithreading.
3. Design applications using file processing, generic programming and event handling

INDEX

CONTENT BEYOND THE SYLLABUS

EXPERIMENT NO	DATE OF EXPERIMENT	TITLE	DATE OF COMPLETION	SIGNATURE
13		Traffic Lights		
14		Mouse Events		

EX.NO : 1	**JAVA CLASS**
DATE:	

Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, and type of EB connection (i.e. domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- >501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- >501 units - Rs. 7 per unit

AIM:

To develop a java program to implement class creation to generate EB bill.

PROBLEM STATEMENT:

1. To create a class for generating electricity bill.
2. To implement control statements for different types of EB connection.
3. To calculate bill amount for various tariffs and display it

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Declare a class to generate the electricity bill.
Step 3: Create 2 methods called domestic() and commercial() for calculating domestic and commercial tariff.
Step 4: Write the necessary calculation for finding the bill amount for various consumed units.
Step 5: Enter the necessary details.
Step 6: Invoke the above two methods.
Step 7: Compute the bill amount and display the EB bill.

PROGRAM:

```
import java.lang.*;
import java.util.*;
class ElectricityBill
{
        public static void main(String...args)
        {
                String consumer_no, consumer_name;
                double prev_month_reading, current_month_reading,no_units;

                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the consumer no :");
                consumer_no = scan.nextLine();
                System.out.println("Enter the consumer name :");
                consumer_name=scan.nextLine();
                System.out.println("Enter the previous month reading :");
                prev_month_reading = scan.nextFloat();
                System.out.println("Enter the current month reading :");
                current_month_reading = scan.nextFloat();
                no_units = current_month_reading - prev_month_reading;
                System.out.println("Select the type of electricity bill :");
                System.out.println("1.Domestic");
                System.out.println("2.Commercial");

                int opt = scan.nextInt();

                switch(opt)
                {
                        case 1 :
                                System.out.println("---------------------------------------------");
                                System.out.println("      ElectricityBill                 ");
                                System.out.println("---------------------------------------------");
                                System.out.println("Consumer name    : "+consumer_name);
                                System.out.println("Consumer no      : "+consumer_no);
                                System.out.println("No.of units     : "+no_units);
                                System.out.println("Electricity Bill : Rs."+domestic(no_units));
                                System.out.println("---------------------------------------------");
                                break;
                        case 2 :

                                System.out.println("---------------------------------------------");
                                System.out.println("      ElectricityBill                ");
                                System.out.println("---------------------------------------------");
                                System.out.println("Consumer name    : "+consumer_name);
                                System.out.println("Consumer no      : "+consumer_no);
```

```
            System.out.println("No.of units     : "+no_units);
            System.out.println("Electricity Bill : Rs."+commercial(no_units));
            System.out.println("---------------------------------------------");
            break;
        default :
            System.out.println("Select the appropriate option!");
            break;
    }

}

public static double domestic(double no_units)
{
    double cost = 1;
    if(no_units <= 100)
    {
        cost = no_units * 1 ;
    }else if(no_units >= 101 && no_units <= 200)
    {
        cost = no_units * 2.50 ;
    }else if(no_units >= 201 && no_units <=500)
    {
        cost = no_units * 4 ;
    }
    else
    {
        cost = no_units * 6;
    }
    return cost;
}

public static double commercial(double no_units)
{
    double cost = 1;
    if(no_units <= 100)
    {
        cost = no_units * 2 ;
    }else if(no_units >= 101 && no_units <= 200)
    {
        cost = no_units * 4.50 ;
    }else if(no_units >= 201 && no_units <= 500)
    {
        cost = no_units * 6 ;
    }
    else
    {
```

```
                    cost = no_units * 7;
            }
            return cost;
        }
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 2	**PACKAGES**
DATE:	

Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages.

AIM:

To design a java application to implement currency, distance and time converter using packages.

PROBLEM STATEMENT:

1. To create three different packages for currency, distance and time conversion.
2. To compute the conversions using respective formulae in respective packages.
3. To include the package in a class and access its functions.
4. To display the conversions.

ALGORITHM:

Step 1: Create a package called pack1 with a class for currency conversion
Step 2: Create a package called pack2 with a class for distance conversion
Step 3: Create another class for main() by importing the above packages
Step 4: Using a special command create the above 2 packages physically.
Step 5: Compile the main class thereby accessing the currency() and distance() of the packages.
Step 6: Display the currency and distance conversions.

PROGRAM:

```
package pack1;
import java.util.Scanner;
public class Currency_convert{
public static void currency(Scanner scan)
        {


                System.out.println("Choose the following :");
                System.out.println("1.Dollar to INR");
                System.out.println("2.INR to Dollar");

                System.out.println("3.EURO to INR");
                System.out.println("4.INR to EURO");
                System.out.println("5.Yen to INR");
                System.out.println("6.INR to Yen");
                int opt = scan.nextInt();
```

```
                    double inr , usd ,euro,yen;
                    String str;
                    switch(opt)
                    {

                         case 1 :
                              System.out.println("Enter the dollars to be converted to INR:");
                               usd = scan.nextFloat();
                               inr = usd * (68.45) ;
                              System.out.printf("Rs.%.2f",inr);
                              System.out.println("\nExit? Y : N");
                               str = scan.next();
                              if( str.equals("N") || str.equals("n") )
                              {
                                   currency(scan);
                              }else
                              {

          System.out.println("Thanks for using the converter ...!");

                              }
                              break;
                         case 2:
                              System.out.println("Enter the rupees to be converted to dollars :");
                               inr = scan.nextFloat();
                               usd = inr / (68.45) ;
                              System.out.printf("$%.2f",usd);
                              System.out.println("\nExit? Y : N");
                               str = scan.next();
                              if( str.equals("N") ||str.equals("n") )
                              {
                                   currency(scan);
                              }else
                              {
          System.out.println("Thanks for using the converter ...!");
                              }
                              break;


                         case 3 :
                              System.out.println("Enter the euros to be converted to INR:");
                               euro = scan.nextFloat();
                               inr = euro * (80.10) ;
                              System.out.printf("Rs.%.2f",inr);
                              System.out.println("\nExit? Y : N");
```

```
                str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    currency(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;
            case 4:
                System.out.println("Enter the rupees to be converted to EURO:");
                 inr = scan.nextFloat();
                 euro = inr / (80.10) ;
                System.out.printf("E%.2f",euro);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    currency(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;

            case 5 :
                System.out.println("Enter the YEN to be converted to INR:");
                 yen = scan.nextFloat();
                 inr = yen * (0.62) ;
                System.out.printf("Rs.%.2f",inr);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    currency(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;
            case 6:
                System.out.println("Enter the rupees to be converted to YEN:");
                 inr = scan.nextFloat();
                 yen = inr / (0.62) ;
                System.out.printf("Y%.2f",yen);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    currency(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;
            default :
                System.out.println("Select one from the above options.");
                break;
        }
    }
}
package pack2;
import java.util.Scanner;
public class Distance_convert{
public static void distance(Scanner scan)
    {
        System.out.println("Choose the following :");
        System.out.println("1.Meters to KM");
        System.out.println("2.KM to Meters");

        System.out.println("3.Miles to KM");
        System.out.println("4.KM to Miles");

        int opt = scan.nextInt();

        double met , km ,miles;
        String str;
        switch(opt)
        {


            case 1 :
                System.out.println("Enter the metres to be converted to KM:");
                 met = scan.nextFloat();
                 km = met / (1000);
                System.out.printf("%.2fkm",km);
                System.out.println("\nExit? Y : N");
                 str = scan.next();


                if( str.equals("N") || str.equals("n") )
                {
                    distance(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;
            case 2:
                System.out.println("Enter the km to be converted to metres :");
                 km = scan.nextFloat();
                 met = km * (1000) ;
                System.out.printf("%.2fm",met);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    distance(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;

            case 3 :
                System.out.println("Enter the miles to be converted to KM:");
                 miles = scan.nextFloat();
                 km = miles / (0.621) ;
                System.out.printf("%.2fkm",km);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
                {
                    distance(scan);
                }else
                {
System.out.println("Thanks for using the converter ...!");
                }
                break;
            case 4:
                System.out.println("Enter the km to be converted to miles:");
                 km = scan.nextFloat();
                 miles = km * (0.621) ;
                System.out.printf("E%.2f",miles);
                System.out.println("\nExit? Y : N");
                 str = scan.next();
                if( str.equals("N") || str.equals("n") )
```

```
                                        {
                                                distance(scan);
                                        }else
                                        {
System.out.println("Thanks for using the converter ...!");
                                        }
                                        break;
                                default :
                                        System.out.println("Select one from the above options.");
                                        break;
                        }
                }
}
import pack1.Currency_convert;
import pack2.Distance_convert;
import java.lang.*;
import java.util.*;

public class Converter{
        public static void main(String...args)
        {
        Currency_convert c= new Currency_convert();
        Distance_convert d=new Distance_convert();
                Scanner scan = new Scanner(System.in);

                System.out.println("Select \n1 for currency converter :\n
                                        2 for  distance converter :");
                int choice = scan.nextInt();
                switch(choice)
                {
                                case 1 :

                                c.currency(scan);
                                        break;
                                case 2 :
                                d.distance(scan);
                                        break;
                                default :

System.out.println("Please choose an appropriate option!");
                                        break;
                }
        }

}
```

OUTPUT:

INFERENCE:

.

RESULT:

EX.NO : 3	**INHERITANCE**
DATE:	

Develop a java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

AIM:

To implement the concept of inheritance in java using pay slip generation application.

PROBLEM STATEMENT:

1. To create a super class for Employee.
2. To implement inheritance concept by creating 4 different derived classes.
3. To inherit the super class to the derived class for pay slips calculation.
4. To generate pay slips for different employees.

ALGORITHM:

```
Step 1:  Import the necessary packages.
Step 2:  Create a super class called Employee with employee details.
Step 3:  Create another 4 derived classes for different categories of employees.
Step 4: In the derived classes, compute the salary details for payslip generation using read(),
        calculate() and display() methods.
Step 5:   Call the methods from their respective classes of employees
Step 6:   Calculate the pay slip values.
Step 7:   Generate the payslip with employee details and salary details.
.
```

PROGRAM:

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

class Employee
{

        String emp_name, emp_id, address, mail_id, mobile ;
        public static void main(String...args)
        {
                Scanner scan = new Scanner(System.in);

                System.out.println("Choose the employee category : ");
                System.out.println("1.Programmer");
                System.out.println("2.Assistant Professor");
                System.out.println("3.Associate Professor");
                System.out.println("4.Professor");
                int cat = scan.nextInt();
                switch(cat)
                {
                        case 1:
                                Programmer programmer = new Programmer();
                                programmer.read();
                                programmer.calculate();
                                programmer.display();
                                break;
                        case 2:
                                AssistantProfessor assistant = new AssistantProfessor();
                                assistant.read();
                                assistant.calculate();
                                assistant.display();
                                break;
                        case 3:
                                AssociateProfessor associate = new AssociateProfessor();
                                associate.read();
                                associate.calculate();
                                associate.display();
                                break;
                        case 4:
                                Professor professor = new Professor();
                                professor.read();
                                professor.calculate();
                                professor.display();
                                break;
                        default :
                                System.out.println("Choose the above category!");
                                break;
                }
        }
}
```

```
class Programmer extends Employee
{
        double basic_pay ,da,hra,pf,fund;
        double gross ,net ;
        Scanner scan = new Scanner(System.in);

        public void read()
        {
                System.out.println("Enter the employee name : ");
                emp_name = scan.nextLine();
                System.out.println("Enter the employee id : ");
                emp_id = scan.nextLine();
                System.out.println("Enter the employee address : ");
                address = scan.nextLine();
                System.out.println("Enter the employee mail_id : ");
                mail_id = scan.nextLine();
                System.out.println("Enter the employee mobile : ");
                mobile = scan.next();
                if(mobile.length() != 10 )
                {
                        System.out.println("Enter the  correct mobile number :\n");
                        mobile = scan.next();
                }

                System.out.println("Enter the basic pay (BP) :");
                basic_pay = scan.nextFloat();
        }

        public void calculate()
        {
                da = (basic_pay * 97 ) / 100 ;
                hra = (basic_pay * 10 ) / 100 ;
                pf = (basic_pay * 12 ) / 100 ;
                fund = (basic_pay * 0.1 ) / 100 ;
                gross = basic_pay + da + hra ;
                net = gross - pf - fund ;
        }

        public void display()
        {
                System.out.println("Employee name : "+emp_name);
                System.out.println("Employee id : "+emp_id);
                System.out.println("Employee address : "+address);
                System.out.println("Employee mail : "+mail_id);
                System.out.println("Employee mobile : "+mobile);
        System.out.println("----------------------------------------------------------------------");
        System.out.println("                    PAY SLIP(Programmer)                     ");
        System.out.println("----------------------------------------------------------------------");
        System.out.println("DA          : Rs."+da+"\t\t\t" +"PF          : Rs."+pf);
        System.out.println("HRA         : Rs."+hra+"\t\t\t"+"FUND        : Rs."+fund);
        System.out.println("----------------------------------------------------------------------");
        System.out.println("Gross salary : Rs."+gross+"\t\t\t"+"Net salary : Rs."+net);
        System.out.println("----------------------------------------------------------------------");

    }
}

class AssistantProfessor extends Employee
{
    double basic_pay ,da,hra,pf,fund;
    double gross ,net ;
    Scanner scan = new Scanner(System.in);

    public void read()
    {
        System.out.println("Enter the employee name : ");
        emp_name = scan.nextLine();
        System.out.println("Enter the employee id : ");
        emp_id = scan.nextLine();
        System.out.println("Enter the employee address : ");
        address = scan.nextLine();
        System.out.println("Enter the employee mail_id : ");
        mail_id = scan.nextLine();
        System.out.println("Enter the employee mobile : ");
        mobile = scan.next();
        if(mobile.length() != 10 )
        {
            System.out.println("Enter the  correct mobile number :\n");
            mobile = scan.next();
        }

        System.out.println("Enter the basic pay (BP) :");
        basic_pay = scan.nextFloat();
    }

    public void calculate()
    {
        da = (basic_pay * 97 ) / 100 ;
        hra = (basic_pay * 10 ) / 100 ;
        pf = (basic_pay * 12 ) / 100 ;
        fund = (basic_pay * 0.1 ) / 100 ;
```

```
                gross = basic_pay + da + hra ;
                net = gross - pf - fund ;
        }

        public void display()
        {
                System.out.println("Employee name : "+emp_name);
                System.out.println("Employee id : "+emp_id);
                System.out.println("Employee address : "+address);
                System.out.println("Employee mail : "+mail_id);
                System.out.println("Employee mobile : "+mobile);
                System.out.println("--------------------------------------------------------------------");
                System.out.println("                    PAY SLIP(AssistantProfessor)                    ");
                System.out.println("--------------------------------------------------------------------");
                System.out.println("DA          : Rs."+da+"\t\t\t" +"PF          : Rs."+pf);
                System.out.println("HRA         : Rs."+hra+"\t\t\t"+"FUND        : Rs."+fund);
                System.out.println("--------------------------------------------------------------------");
                System.out.println("Gross salary : Rs."+gross+"\t\t\t"+"Net salary : Rs."+net);
                System.out.println("--------------------------------------------------------------------");

        }
}

class AssociateProfessor  extends Employee
{
        double basic_pay ,da,hra,pf,fund;
        double gross ,net ;
        Scanner scan = new Scanner(System.in);

        public void read()
        {
                System.out.println("Enter the employee name : ");
                emp_name = scan.nextLine();
                System.out.println("Enter the employee id : ");
                emp_id = scan.nextLine();
                System.out.println("Enter the employee address : ");
                address = scan.nextLine();
                System.out.println("Enter the employee mail_id : ");
                mail_id = scan.nextLine();
                System.out.println("Enter the employee mobile : ");
                mobile = scan.next();
                if(mobile.length() != 10 )
                {
                        System.out.println("Enter the  correct mobile number :\n");
                        mobile = scan.next();
                }
                System.out.println("Enter the basic pay (BP) :");
                basic_pay = scan.nextFloat();
        }

        public void calculate()
        {
                da = (basic_pay * 97 ) / 100 ;
                hra = (basic_pay * 10 ) / 100 ;
                pf = (basic_pay * 12 ) / 100 ;
                fund = (basic_pay * 0.1 ) / 100 ;
                gross = basic_pay + da + hra ;
                net = gross - pf - fund ;
        }

        public void display()
        {
                System.out.println("Employee name : "+emp_name);
                System.out.println("Employee id : "+emp_id);
                System.out.println("Employee address : "+address);
                System.out.println("Employee mail : "+mail_id);
                System.out.println("Employee mobile : "+mobile);
                System.out.println("----------------------------------------------------------------------");
                System.out.println("                    PAY SLIP(AssociateProfessor)                    ");
                System.out.println("----------------------------------------------------------------------");
                System.out.println("DA          : Rs."+da+"\t\t\t" +"PF          : Rs."+pf);
                System.out.println("HRA         : Rs."+hra+"\t\t\t"+"FUND        : Rs."+fund);
                System.out.println("----------------------------------------------------------------------");
                System.out.println("Gross salary : Rs."+gross+"\t\t\t"+"Net salary : Rs."+net);
                System.out.println("----------------------------------------------------------------------");

        }
}

class Professor extends Employee
{
        double basic_pay ,da,hra,pf,fund;
        double gross ,net ;
        Scanner scan = new Scanner(System.in);

        public void read()
        {
                System.out.println("Enter the employee name : ");
                emp_name = scan.nextLine();
                System.out.println("Enter the employee id : ");
                emp_id = scan.nextLine();
                System.out.println("Enter the employee address : ");
                address = scan.nextLine();
                System.out.println("Enter the employee mail_id : ");
                mail_id = scan.nextLine();
                System.out.println("Enter the employee mobile : ");
                mobile = scan.next();
                if(mobile.length() != 10 )
                {
                    System.out.println("Enter the  correct mobile number :\n");
                    mobile = scan.next();
                }

                System.out.println("Enter the basic pay (BP) :");
                basic_pay = scan.nextFloat();
        }

        public void calculate()
        {
                da = (basic_pay * 97 ) / 100 ;
                hra = (basic_pay * 10 ) / 100 ;
                pf = (basic_pay * 12 ) / 100 ;
                fund = (basic_pay * 0.1 ) / 100 ;
                gross = basic_pay + da + hra ;
                net = gross - pf - fund ;
        }

        public void display()
        {
                System.out.println("Employee name : "+emp_name);
                System.out.println("Employee id : "+emp_id);
                System.out.println("Employee address : "+address);
                System.out.println("Employee mail : "+mail_id);
                System.out.println("Employee mobile : "+mobile);
                System.out.println("-----------------------------------------------------------------------");
                System.out.println("                        PAY SLIP(Professor)                        ");
                System.out.println("-----------------------------------------------------------------------");
                System.out.println("DA          : Rs."+da+"\t\t\t" +"PF          : Rs."+pf);
                System.out.println("HRA         : Rs."+hra+"\t\t\t"+"FUND        : Rs."+fund);
                System.out.println("-----------------------------------------------------------------------");
                System.out.println("Gross salary : Rs."+gross+"\t\t\t"+"Net salary : Rs."+net);
                System.out.println("-----------------------------------------------------------------------");

        }
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 4	**INTERFACE AND EXCEPTION HANDLING**
DATE:	

Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.

AIM:

To implement the concept of interface by designing an ADT stack application and also providing necessary exception handling mechanism.

PROBLEM STATEMENT:

1. To create an interface for ADT stack with push, pop, peek and display methods.
2. To create an implementation class by defining the interface methods
3. To access the interface by calling each method.
4. To provide exception handling code for the whole application.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Declare an interface with push(), pop(), and display() methods.
Step 3: Create an implementation class for the interface.
Step 4: Write definition for the given methods in the implementation class.
Step 5: Call the methods from main method.
Step 6: Display the results.

PROGRAM:

```
import java.io.*;
import java.util.*;

interface MyInterface {
int n = 20;

public void pop();

public void push();


public void display();
}
```

```
class StackImplementation implements MyInterface {
int arr[] = new int[n];
int top = -1;

public void push() {
        try {
                DataInputStream dis = new DataInputStream(System.in);
                System.out.println("Enter Element");
                int ele = Integer.parseInt(dis.readLine());
                arr[++top] = ele;
             } catch (Exception e) {
                System.out.println("e");
                }
                }

public void pop() {
int popper = arr[top];
top--;
System.out.println("popped element " + popper);
}


public void display() {
if (top < 0) {
System.out.println("Stack is empty");
return;
} else {
String str = " ";
for (int i = 0; i <= top; i++)
str = str + " " + arr[i];
System.out.println("Elements are " + str);
}
}
}

class StackADT {
public static void main(String arg[]) throws IOException {
DataInputStream dis = new DataInputStream(System.in);
StackImplementation stk = new StackImplementation();
int menu = 0;

do {
System.out.println("1.push \n2.pop \n3.display \n4.Exit");
System.out.println();
System.out.print("Enter your choice: ");
menu = Integer.parseInt(dis.readLine());
switch (menu) {

case 1:
stk.push();
break;

case 2:
stk.pop();
break;

case 3:
stk.display();
break;

case 4:
System.exit(0);
}
} while (menu <= 4);
System.out.println();
}
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 5	**STRING MANIPULATION USING ARRAYLIST**
DATE:	

Write a program to perform string operations using ArrayList. Write functions for the following
a. Append - add at end
b. Insert – add at particular index
c. Search
d. List all string starts with given letter

AIM:

To develop a java program to perform string operations using ArrayList.

PROBLEM STATEMENTS:

1. To create a class using Arraylist.
2. To use the String manipulation methods of Arraylist to append, search and display.
3. To call the methods and display the results.

ALGORITHM:

```
Step 1: Import the necessary packages.
Step 2: Create an ArrayList class.
Step 3: Use the method to append the string at the end.
Step 4: Use another method to insert a string at a particular index.
Step 5: Use another method to search for a particular element.
Step 6: Write a method to find whether the given string starts with a particular letter.
Step 7: Display the results.
```

PROGRAM:

```
import java.util.*;
public class ArrayListDemo {
  public static void main(String args[]) {
List<String> listStrings = new ArrayList<String>();
//Appending elements one by one at the end
listStrings.add("One");
listStrings.add("Two");
listStrings.add("Three");
listStrings.add("Four");
System.out.println(listStrings);

// Inserting an element at a particular index
 listStrings.add(1, "A2");
System.out.println(listStrings);

//Search for a particular element
if (listStrings.contains("Hello")) {
    System.out.println("Found the element");
} else {
    System.out.println("There is no such element");
}

//Listing all elements starting with a given letter
for(int i=0;i<listStrings.size();i++)
        {
            String newString = listStrings.get(i);
            if (newString.startsWith("T")){
System.out.println("The found the element is "+newString);
                                                            }
        }
}
}
```

OUTPUT:

INFERENCE:

.

RESULT:

EX.NO : 6	**ABSTRACT CLASS**
DATE:	

Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

AIM:

To develop a java program using abstract class and find the area of the given shape.

PROBLEM STATEMENTS:

1. To create an abstract class with an abstract method to find the area.
2. To create 3 classes for 3 different shapes that implements the abstract class.
3. To call the methods and print the area of different shapes.

ALGORITHM:

```
Step 1: Import the necessary packages.
Step 2: Create an abstract class called shape with 2 integers and an empty method printArea().
Step 3: Create 3 different classes for Rectangle, Triangle and Circle shapes.
Step 4: Provide implementation details for the printArea() in each of the classes.
Step 5: Create object for the different classes created for different shapes.
Step 6: Call the printArea() using different objects created.
Step 7: Display the area of the given shapes.
```

PROGRAM:

```
import java.io.*;
abstract class shape
{
int a=3,b=4;
 abstract public void print_area();
}

//Rectangle
class rectangle extends shape
{
 public int area_rect;
        @Override
 public void print_area()
 {
  area_rect=a*b;
  System.out.println("The area of rectangle is:"+area_rect);
 }
}

//Triangle
class triangle extends shape
{
 int area_tri;
        @Override
 public void print_area()
 {
  area_tri=(int) (0.5*a*b);
   System.out.println("The area of triangle is:"+area_tri);
 }
}

//Circle
class circle extends shape
{
 int area_circle;
        @Override
 public void print_area()
 {
  area_circle=(int) (3.14*a*a);
   System.out.println("The area of circle is:"+area_circle);
 }
}

public class AbstractDemo {
       public static void main(String[] args) {
        rectangle r=new rectangle();
        r.print_area();
        triangle t=new triangle();
        t.print_area();
        circle r1=new circle();
        r1.print_area();
              }


}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 7	USER DEFINED EXCEPTION HANDLING
DATE:	

Write a Java program to implement user defined exception handling

AIM:

To develop a java program to implement user defined exception handling.

PROBLEM STATEMENTS:

1. To create a user-defined exception class with inherits Java Exception class.
2. To create constructors for password and age.
3. To enter the password and age and invoke the exception class if the values are wrong.

ALGORITHM:

Step 1: Import the necessary package.
Step 2: Create a class for user-defined exception which inherits Java.Exception clas.
Step 3: Inside the class create two constructors for checking password length and age.
Step 4: Create another class to access the user-defined exception.
Step 5: Enter the user name, password and age.
Step 6: If the password length and age doesn’t match, then user-defined exception is called.

PROGRAM:

```
import java.io.*;
class myException extends Exception

{ myException()
  { System.out.println("Error:Password too short");
  }

  myException(int n)
  { System.out.println("Error:Only adults can join");
  }
}



class User_exception
{ public static void main(String args[]) throws  IOException,myException
```

```
  { BufferedReader br=new BufferedReader(new

InputStreamReader(System.in));
    try
    { System.out.print("Enter user name : ");
      String n=br.readLine();
      System.out.print("Enter your password : ");
      String m=br.readLine();
      if(m.length() <6)
        throw new myException();
      System.out.print("Enter your age : ");
      int age=Integer.parseInt(br.readLine());
      if(age<18)
        throw new myException(age);
    }
    catch(Exception e)
    {    }
  }
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 8	**FILE PROCESSING**
DATE:	

Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

AIM:

To implement the concept of file processing using a java program.

PROBLEM STATEMENTS:

1. To create a class for file manipulation
2. To create different methods to get the file name and its properties.
3. To access the methods and display the file details.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Create a class.
Step 3: Write the methods for finding the given file and its properties.
Step 4: Call the file handling methods from main method.
Step 5: Display the details of the file.

PROGRAM:

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

class File_info
{
        public static void main(String...args)
        {
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter the file name with extension :");
                String name = scan.nextLine();
                File file = new File(name);
                String path = file.getAbsolutePath();

                System.out.println("File exists : "+file.exists());
                System.out.println("Readable : "+file.canRead());
                System.out.println("Writable : "+file.canWrite());
                System.out.println("Length of file : "+file.length()+" bytes");
                System.out.println("File Type : "+ getExtension(path));
```

```
            }

            public static String getExtension(String path) {
              int dot = path.lastIndexOf(".");
            return path.substring(dot + 1);
  }
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 9	**MULTITHREADING**
DATE:	

Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

AIM:

To implement multithreading concept using a java program.

PROBLEM STATEMENTS:

1. To create two classes one for odd and other for even value by implementing Runnable interface.
2. To create another class for creating Thread.
3. To create main thread to generate random numbers.
4. To check for even or odd values and print the respective square and cube values.

ALGORITHM:

Step 1: Import the necessary package.
Step 2: Create a class which extends Runnable interface for even value.
Step 3: Create another class which extends Runnable interface for odd value.
Step 4: Create a class with main method which creates another thread to generate random numbers.
Step 5: Call the respective constructors odd() & even() to print the square and cube of a number.
Step 6: If even, then square the value and print the result.
Step 7: If odd, then cube the value and print the result.

PROGRAM:

```
import java.util.*;

class even implements Runnable
{
public int x;
public even(int x)
{
this.x = x;
}
public void run()
{
System.out.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x * x);
}
}

class odd implements Runnable
{
public int x;
public odd(int x)
{
this.x = x;
}
public void run()
{
System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x * x * x);
}
}

class A extends Thread
{
public void run()
{
int num = 0;
Random r = new Random();
        try
                {
                for (int i = 0; i < 5; i++)
                {
                num = r.nextInt(100);
                System.out.println("Main Thread and Generated Number is " + num);
                if (num % 2 == 0)
                {
                Thread t1 = new Thread(new even(num));
                t1.start();
                } else {
                Thread t2 = new Thread(new odd(num));
                t2.start();
                }
                Thread.sleep(1000);
                System.out.println("--------------------------------------");
                }
                }
        catch (Exception ex)
                {
                System.out.println(ex.getMessage());
                }
                }
}
```

```
public class Multithreading
{
public static void main(String[] args)
{
A a = new A();
a.start();
}
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 10	GENERIC FUNCTION
DATE:	

Write a java program to find the maximum value from the given type of elements using a generic function.

AIM:

To implement a java generic program to find the maximum value from the given type of elements.

PROBLEM STATEMENTS:

1. To create a generic class for different elements.
2. To create a method to find the maximum value of given elements.
3. To invoke the generic method and find the result.

ALGORITHM:

```
Step 1:  Create a generic class
Step 2:  Declare a static method for finding the maximum element.
Step 3:  Write the code for finding the maximum value of an integer, double and String values using
         compareTo() which is used for comparing two strings lexicographically.
Step 4:  Get the values and display the maximum values of each element type.
```

PROGRAM:

```
public class MaximumTest {
  // determines the largest of three Comparable objects

  public static <T extends Comparable<T>> T maximum(T x, T y, T z) {
    T max = x;   // assume x is initially the largest

    if(y.compareTo(max) > 0) {
      max = y;   // y is the largest so far
    }

    if(z.compareTo(max) > 0) {
      max = z;   // z is the largest now
    }
    return max;   // returns the largest object
  }

  public static void main(String args[]) {
    System.out.printf("Max of %d, %d and %d is %d\n\n",
      3, 4, 5, maximum( 3, 4, 5 ));
      System.out.printf("Max of %.1f,%.1f and %.1f is %.1f\n\n",
         6.6, 8.8, 7.7, maximum( 6.6, 8.8, 7.7 ));

      System.out.printf("Max of %s, %s and %s is %s\n","pear",
         "apple", "orange", maximum("pear", "apple", "orange"));
   }
}
```

OUTPUT:

INFERENCE:

RESULT:

.

EX.NO : 11	EVENT-DRIVEN PROGRAMMING
DATE:	

Design a calculator using event-driven programming paradigm of Java with the following options.
a) Decimal manipulations
b) Scientific manipulations

AIM:

To create a scientific calculator using java event-driven programming concept.

PROBLEM STATEMENTS:

1. To create a class using AWT swing concept.
2. To implement the event driven programming concept in the same method.
3. To write the functionalities for different operations of a scientific calculator.
4. To invoke the class and display the calculator.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Create a class for scientific calculator by extending AWT swing class and implementing event-driven Programming interface.
Step 3: Declare and define the various functions of a calculator.
Step 4: Using main method invoke the calculator and display the functions of the calculator.

PROGRAM:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.event.*;
public class ScientificCalculator extends JFrame implements ActionListener
{
        JTextField tfield;

        double temp,temp1,result,a,ml;
        static double m1,m2;
        int k=1,x=0,y=0,z=0;
        char ch;
JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,zero,clr,pow2,pow3,exp,fac,plus,min,div,log,
rec,mul,eq,addsub,dot,mr,mc,mp,mm,sqrt,sin,cos,tan;
Container cont;
JPanel textpanel,buttonpanel;
ScientificCalculator()
{
        cont=getContentPane();
        cont.setLayout(new BorderLayout());
   JPanel textpanel=new JPanel();
   tfield=new JTextField(25);
   tfield.setHorizontalAlignment(SwingConstants.RIGHT);
   tfield.addKeyListener(new KeyAdapter()
   {
        public void keyTyped(KeyEvent keyevent)
        {
                char c=keyevent.getKeyChar();
                if(c>='0'&&c<='9')
                {
                }
                else
                {
                        keyevent.consume();
                }
        }
   });
textpanel.add(tfield);
buttonpanel=new JPanel();
buttonpanel.setLayout(new GridLayout(8,4,2,2));
boolean t=true;
mr=new JButton("MR");
buttonpanel.add(mr);
mr.addActionListener(this);
mc=new JButton("MC");
buttonpanel.add(mc);
mc.addActionListener(this);
mp=new JButton("M+");
buttonpanel.add(mp);
mp.addActionListener(this);
mm=new JButton("M-");
buttonpanel.add(mm);
mm.addActionListener(this);
b1=new JButton("1");
buttonpanel.add(b1);
b1.addActionListener(this);
b2=new JButton("2");
buttonpanel.add(b2);
b2.addActionListener(this);
```

```
b3=new JButton("3");
buttonpanel.add(b3);
b3.addActionListener(this);
b4=new JButton("4");
buttonpanel.add(b4);
b4.addActionListener(this);
b5=new JButton("5");
buttonpanel.add(b5);
b5.addActionListener(this);
b6=new JButton("6");
buttonpanel.add(b6);
b6.addActionListener(this);
b7=new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8=new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9=new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);
zero=new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);
plus=new JButton("+");
buttonpanel.add(plus);
plus.addActionListener(this);
min=new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);
mul=new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);
div=new JButton("/");
buttonpanel.add(div);
div.addActionListener(this);
addsub=new JButton("+/-");
buttonpanel.add(addsub);
addsub.addActionListener(this);
dot=new JButton(".");
buttonpanel.add(dot);
dot.addActionListener(this);
eq=new JButton("=");
buttonpanel.add(eq);
eq.addActionListener(this);
rec=new JButton("1/x");
```

```
buttonpanel.add(rec);
rec.addActionListener(this);
sqrt=new JButton("Sqrt");
buttonpanel.add(sqrt);
sqrt.addActionListener(this);
log=new JButton("log");
buttonpanel.add(log);
log.addActionListener(this);
sin=new JButton("SIN");
buttonpanel.add(sin);
sin.addActionListener(this);
cos=new JButton("COS");
buttonpanel.add(cos);
cos.addActionListener(this);
tan=new JButton("TAN");
buttonpanel.add(tan);
tan.addActionListener(this);
pow2=new JButton("x^2");
buttonpanel.add(pow2);
pow2.addActionListener(this);
pow3=new JButton("X^3");
buttonpanel.add(pow3);
pow3.addActionListener(this);
exp=new JButton("Exp");
buttonpanel.add(exp);
exp.addActionListener(this);
fac=new JButton("n!");
buttonpanel.add(fac);
fac.addActionListener(this);
clr=new JButton("AC");
buttonpanel.add(clr);
clr.addActionListener(this);
cont.add("Center",buttonpanel);
cont.add("North",textpanel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent e)
{
        String s=e.getActionCommand();
        if(s.equals("1"))
        {
                if(z==0)
                {
                        tfield.setText(tfield.getText()+"1");
                }
                else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"1");
                z=0;
        }
}
if(s.equals("2"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"2");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"2");
                z=0;
        }
}
if(s.equals("3"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"3");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"3");
                z=0;
        }
}
if(s.equals("4"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"4");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"4");
                z=0;
        }
}
if(s.equals("5"))
```

```
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"5");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"5");
                z=0;
        }
}
if(s.equals("6"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"6");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"6");
                z=0;
        }
}
if(s.equals("7"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"7");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"7");
                z=0;
        }
}
if(s.equals("8"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"8");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"8");
                z=0;
        }
}
if(s.equals("9"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"9");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"9");
                z=0;
        }
}
if(s.equals("0"))
{
        if(z==0)
        {
                tfield.setText(tfield.getText()+"0");
        }
        else
        {
                tfield.setText("");
                tfield.setText(tfield.getText()+"0");
                z=0;
        }
}
if(s.equals("AC"))
{
                tfield.setText("");
                x=0;y=0;
                z=0;
        }
if(s.equals("log"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.log(Double.parseDouble(tfield.getText()));
```

```
                tfield.setText("");
                tfield.setText(tfield.getText()+a);

        }
}
if(s.equals("1/x"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=1/(Double.parseDouble(tfield.getText()));
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("Exp"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.exp(Double.parseDouble(tfield.getText()));
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("x^2"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.pow(Double.parseDouble(tfield.getText()),2);
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("X^3"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.pow(Double.parseDouble(tfield.getText()),3);
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("+/-"))
{
        if(x==0)
        {
                tfield.setText("-"+tfield.getText());
                x=1;
        }
        else
        {
           tfield.setText(tfield.getText());
        }
}
if(s.equals("."))
{
        if(y==0)
        {
                tfield.setText(tfield.getText()+".");
                y=1;
        }
        else
        {
           tfield.setText(tfield.getText());
        }
}
if(s.equals("+"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
                temp=0;
                ch='+';
        }
        else
        {
           temp=Double.parseDouble(tfield.getText());
        tfield.setText("");
        ch='+';
        y=0;x=0;
    }
    tfield.requestFocus();
}
if(s.equals("-"))
{
    if(tfield.getText().equals(""))
    {
        tfield.setText("");
        temp=0;
        ch='-';
    }
    else
    {
            y=0;x=0;
     temp=Double.parseDouble(tfield.getText());
        tfield.setText("");
        ch='-';
    }
    tfield.requestFocus();
}
if(s.equals("/"))
{
    if(tfield.getText().equals(""))
    {
        tfield.setText("");
        temp=1;
        ch='/';
    }
    else
    {
            y=0;x=0;
     temp=Double.parseDouble(tfield.getText());
        ch='/';
        tfield.setText("");
    }
    tfield.requestFocus();
}
if(s.equals("*"))
{
    if(tfield.getText().equals(""))
    {
        tfield.setText("");
        temp=1;
```

```
                ch='*';
        }
        else
        {
                        y=0;x=0;
           temp=Double.parseDouble(tfield.getText());
                ch='*';
                tfield.setText("");
        }
        tfield.requestFocus();
}
if(s.equals("MC"))
{
        ml=0;
        tfield.setText("");
}
if(s.equals("MR"))
{
  tfield.setText("");
  tfield.setText(tfield.getText()+ml);
}
if(s.equals("M+"))
{
        if(k==1)
        {
                ml=Double.parseDouble(tfield.getText());
                k++;
        }
        else
        {
                ml+=Double.parseDouble(tfield.getText());
                tfield.setText(""+ml);
        }
}
if(s.equals("M-"))
{
        if(k==1)
        {
                ml=Double.parseDouble(tfield.getText());
                k++;
        }
        else
        {
                ml-=Double.parseDouble(tfield.getText());
                tfield.setText(""+ml);
        }
```

```
}
if(s.equals("Sqrt"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.sqrt(Double.parseDouble(tfield.getText()));
                tfield.setText("");
                tfield.setText(tfield.getText()+a);

        }
}
if(s.equals("SIN"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.sin(Double.parseDouble(tfield.getText()));
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("COS"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
        }
        else
        {
           a=Math.cos(Double.parseDouble(tfield.getText()));
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}
if(s.equals("TAN"))
{
        if(tfield.getText().equals(""))
        {
                tfield.setText("");
```

```
            }
            else
            {
                a=Math.tan(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText()+a);
            }
}
if(s.equals("="))
{
            if(tfield.getText().equals(""))
            {
                        tfield.setText("");
            }
            else
            {
                        temp1=Double.parseDouble(tfield.getText());
                        switch(ch)
                        {
                                    case '+':
                                                result=temp+temp1;
                                                break;
                            case '-':
                                                result=temp-temp1;
                                                break;
                                    case '/':
                                                result=temp/temp1;
                                                break;
                                    case '*':
                                                result=temp*temp1;
                                                break;
                        }
        tfield.setText("");
        tfield.setText(tfield.getText()+result);
        z=1;
            }
}
if(s.equals("n!"))
{
            if(tfield.getText().equals(""))
            {
                        tfield.setText("");
            }
            else
            {
                a=fact(Double.parseDouble(tfield.getText()));
```

```
                                        tfield.setText("");
                                        tfield.setText(tfield.getText()+a);
                                }
                        tfield.requestFocus();
                        }
                }
                double fact(double x)
                {
                        /*int er=0;
                        if(x>0)
                        {
                                er=20;
                                return 0;
                        }*/
                        double i,s=1;
                        for(i=2;i<=x;i+=1.0)
                          s*=i;
                          return s;
                }
            public static void main(String srgs[])
            {
                  ScientificCalculator f;
                        f=new ScientificCalculator();
                        f.setTitle("Scientific calculator");
                        f.pack();
                        f.setVisible(true);

            }
        }
```

OUTPUT:

INFERENCE:

RESULT:

.

EX.NO : 12	MINI PROJECT USING JAVA CONCEPTS
DATE:	

Develop a mini project for any application using Java concepts.

AIM:

To implement an online quiz application using java SWING concepts.

PROBLEM STATEMENTS:

1. To implement a class using AWT swing package.
2. To create coding for 10 java questions with answers.
3. To create a GUI window to take the exam and display the results.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Declare the class for creating Online exam using AWT swing concept.
Step 3: Declare the necessary API components.
Step 4: Write the code for finding the answer.
Step 5: Consolidate the scores and display the total score.

PROGRAM:

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

class OnlineTest extends JFrame implements ActionListener
{
   JLabel l;
   JRadioButton jb[]=new JRadioButton[5];
   JButton b1,b2;
   ButtonGroup bg;
   int count=0,current=0,x=1,y=1,now=0;
   int m[]=new int[10];
   OnlineTest(String s)
   {
      super(s);
      l=new JLabel();
      add(l);
      bg=new ButtonGroup();
```

```
        for(int i=0;i<5;i++)
        {
            jb[i]=new JRadioButton();
            add(jb[i]);
            bg.add(jb[i]);
        }
        b1=new JButton("Next");
        b2=new JButton("Bookmark");
        b1.addActionListener(this);
        b2.addActionListener(this);
        add(b1);add(b2);
        set();
        l.setBounds(30,40,450,20);
        jb[0].setBounds(50,80,100,20);
        jb[1].setBounds(50,110,100,20);
        jb[2].setBounds(50,140,100,20);
        jb[3].setBounds(50,170,100,20);
        b1.setBounds(100,240,100,30);
        b2.setBounds(270,240,100,30);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);
        setLocation(250,100);
        setVisible(true);
        setSize(600,350);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            if(check())
                count=count+1;
            current++;
            set();
            if(current==9)
            {
                b1.setEnabled(false);
                b2.setText("Result");
            }
        }
        if(e.getActionCommand().equals("Bookmark"))
        {
            JButton bk=new JButton("Bookmark"+x);
            bk.setBounds(480,20+30*x,100,30);
            add(bk);
            bk.addActionListener(this);
            m[x]=current;
            x++;
            current++;
            set();
            if(current==9)
                b2.setText("Result");
            setVisible(false);
            setVisible(true);
        }
        for(int i=0,y=1;i<x;i++,y++)
        {
        if(e.getActionCommand().equals("Bookmark"+y))
        {
            if(check())
                count=count+1;
            now=current;
            current=m[y];
            set();
            ((JButton)e.getSource()).setEnabled(false);
            current=now;
        }
        }

        if(e.getActionCommand().equals("Result"))
        {
            if(check())
                count=count+1;
            current++;
            //System.out.println("correct ans="+count);
            JOptionPane.showMessageDialog(this,"correctans="+count);
            System.exit(0);
        }
    }
    void set()
    {
        jb[4].setSelected(true);
        if(current==0)
        {
            l.setText("Que1: Which one among these is not a primitive datatype?");
            jb[0].setText("int"); jb[1].setText("Float"); jb[2].setText("boolean"); jb[3].setText("char");
        }
        if(current==1)
        {
            l.setText("Que2: Which class is available to all the class automatically?");
             jb[0].setText("Swing"); jb[1].setText("Applet"); jb[2].setText("Object");
              jb[3].setText("ActionEvent");
        }
```

```
if(current==2)
{
   l.setText("Que3: Which package is directly available to our class without importing it?");
   jb[0].setText("swing"); jb[1].setText("applet"); jb[2].setText("net"); jb[3].setText("lang");
}
if(current==3)
{
   l.setText("Que4: String class is defined in which package?");
   jb[0].setText("lang"); jb[1].setText("Swing"); jb[2].setText("Applet"); jb[3].setText("awt");
}
if(current==4)
{
   l.setText("Que5: Which is the best programming language?");

    jb[0].setText("C"); jb[1].setText("C++"); jb[2].setText("Java"); jb[3].setText("Dot Net");
}
if(current==5)
{
   l.setText("Que6: Which one among these is not a keyword?");
   jb[0].setText("class"); jb[1].setText("int"); jb[2].setText("get"); jb[3].setText("if");
}
if(current==6)
{
   l.setText("Que7: Which one among these is not a class? ");
   jb[0].setText("Swing"); jb[1].setText("Actionperformed"); jb[2].setText("ActionEvent");
   jb[3].setText("Button");
}
if(current==7)
{
   l.setText("Que8: which one among these is not a function of Object class?");
   jb[0].setText("toString"); jb[1].setText("finalize"); jb[2].setText("equals");
   jb[3].setText("getDocumentBase");
}
if(current==8)
{
   l.setText("Que9: which function is not present in Applet class?");
   jb[0].setText("init"); jb[1].setText("main"); jb[2].setText("start");
   jb[3].setText("destroy");
}

if(current==9)
{
   l.setText("Que10: Which one among these is not a valid component?");
   jb[0].setText("JButton"); jb[1].setText("JList"); jb[2].setText("JButtonGroup");
   jb[3].setText("JTextArea");
}
```

```
        l.setBounds(30,40,450,20);
        for(int i=0,j=0;i<=90;i+=30,j++)
            jb[j].setBounds(50,80+i,200,20);
    }
    boolean check()
    {
        if(current==0)
            return(jb[1].isSelected());
        if(current==1)
            return(jb[2].isSelected());
        if(current==2)
            return(jb[3].isSelected());
        if(current==3)
            return(jb[0].isSelected());
        if(current==4)
            return(jb[2].isSelected());
        if(current==5)
            return(jb[2].isSelected());
        if(current==6)
            return(jb[1].isSelected());
        if(current==7)
            return(jb[3].isSelected());
        if(current==8)
            return(jb[1].isSelected());
        if(current==9)
            return(jb[2].isSelected());
        return false;
    }
    public static void main(String s[])
    {
        new OnlineTest("Online Test Of Java");
    }
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO : 13	TRAFFIC LIGHTS
DATE:	

Write a Java Program that simulates a Traffic Lights. The program lets the use select one of three lights red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with “stop” or “Ready” or “GO” should appear above the button in selected color. Initially, there is no message shown.

AIM:

To implement a traffic light system using java Swing concept.

PROBLEM STATEMENTS:

1. To create a class for GUI using Swing and Event handling classes.
2. To write code for traffic lighting.
3. To invoke the GUI window and check for the results.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Declare the class extending swing classes and implementing event driven interfaces.
Step 3: Declare the AWT components.
Step 4: Write the functionality to implement traffic light system.
Step 5: Create the object for the class.
Step 6: Invoke the constructor to display the GUI.
Step 7: Check with the different available options in the AWT window.
Step 8: Display the results.

PROGRAM:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class Trafficlight extends JFrame implements ItemListener
{
public JLabel l1, l2;
public JRadioButton r1, r2, r3;
public ButtonGroup bg;
public JPanel p, p1;
CheckboxGroup lngGrp;
public Trafficlight()
```

```
{
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(2, 1));
setSize(800, 400);
p = new JPanel(new FlowLayout());
p1 = new JPanel(new FlowLayout());
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 60);
l1.setFont(f); add(l1); p.add(l1);
add(p);
l2 = new JLabel("Select Lights");
p1.add(l2);
lngGrp = new CheckboxGroup();
Checkbox r1 = new Checkbox("Red Light", lngGrp, true);
r1.setBackground(Color.red);
p1.add(r1);
r1.addItemListener(this);
Checkbox r2 = new Checkbox("Yellow Light", lngGrp, true);
r2.setBackground(Color.YELLOW);
p1.add(r2);
r2.addItemListener(this);
Checkbox r3 = new Checkbox("Green Light", lngGrp, true);
r3.setBackground(Color.GREEN);
p1.add(r3);
r3.addItemListener(this);
add(p1);
setVisible(true);
}
public void itemStateChanged(ItemEvent i)
{
Checkbox chk = lngGrp.getSelectedCheckbox();
switch (chk.getLabel())
{
case "Red Light":
l1.setText("STOP");
l1.setForeground(Color.red);
break;
case "Yellow Light":
l1.setText("Ready");
l1.setForeground(Color.YELLOW);
break;
case "Green Light":
l1.setText("GO");
l1.setForeground(Color.GREEN);
break;
}
}
}
public class TLights {
public static void main(String[] args)
{
Trafficlight a = new Trafficlight();
}
}
```

OUTPUT:

INFERENCE:

RESULT:

EX.NO :14	MOUSE EVENTS
DATE:	

Write a Java Program that handles all mouse events and show event name at the center of the window when the mouse event is fired.

AIM:

To implement a java program to demonstrate various mouse events.

PROBLEM STATEMENTS:

1. To create a class implementing MouseListener interface.
2. To write code for various mouse events.
3. To invoke the functions and display the mouse events.

ALGORITHM:

Step 1: Import the necessary packages.
Step 2: Declare the class which extends Swing classes and implementing event handling interfaces.
Step 3: Declare and define the functions to call the mouse events
Step 4: Create the object for the class and invoke the GUI.
Step 5: Display the GUI and demonstrate the different events of mouse.

PROGRAM:

```
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
class A extends JFrame implements MouseListener
{
JLabel l1;
public A()
{
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 400);
setLayout(new FlowLayout());
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 20);
l1.setFont(f);
l1.setForeground(Color.BLUE);
l1.setAlignmentX(Component.CENTER_ALIGNMENT);
l1.setAlignmentY(Component.CENTER_ALIGNMENT);
add(l1);
addMouseListener(this);
setVisible(true);
}
public void mouseExited(MouseEvent m)
{
l1.setText("Mouse Exited");
}
public void mouseEntered(MouseEvent m)
{
l1.setText("Mouse Entered");
}
public void mouseReleased(MouseEvent m)
{
l1.setText("Mouse Released");
}
public void mousePressed(MouseEvent m)
{
l1.setText("Mouse Pressed");
}
public void mouseClicked(MouseEvent m)
{
l1.setText("Mouse Clicked");
}
}
public class Mevents
{
public static void main(String[] args) {
A a = new A();
}
}
```

OUTPUT:

INFERENCE:

RESULT:

Some of the core java projects, students could undertake:

1. **LIBRARY MANAGEMENT SYSTEM**
2. **Exam system**
3. **Fee report**
4. **Billing system**
5. **Library Management system**
6. **Banking System**