ST. ANNE'S

COLLEGE OF ENGINEERING AND TECHNOLOGY

ANGUCHETTYPALAYAM, PANRUTI – 607 110

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

CS8383 - OBJECT ORIENTED PROGRAMMING LABORATORY

Regulation 2017

PREPARED BY

K.POORNAMBIGAI,

Asst. Professor / CSE

CS8383 OBJECT ORIENTED PROGRAMMING LABORATORY

L T P C
0 0 4 2

OBJECTIVES

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

LIST OF EXPERIMENTS

1. 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, 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

2. 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.

3. 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.

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

5. 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

6. 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.

7. Write a Java program to implement user defined exception handling.

8. 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.

9. 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.

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

11. Design a calculator using event-driven programming paradigm of Java with the following options.

a) Decimal manipulations
b) Scientific manipulations

12. Develop a mini project for any application using Java concepts.

TOTAL : 60 PERIODS

OUTCOMES

Upon completion of the course, the students will be able to

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

GENERATING ELECTRICITY BILL

AIM:

To Develop a Java application to generate Electricity bill.

ALGORITHM:

```
1. Import the java packages.
2. Create a class with members Consumer no., consumer name, previous month reading,
   current month reading, type of EB connection (i.e domestic or commercial).
3. Class also contains methods domesticbillcalc and commercialbillcalc with its parameters to
compute bill amount.
4. Check whether the type of connection is domestic or commercial.
    If domestic, calculate the bill amount 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 commercial, calculate the bill amount 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
5. Calculate the units consumed by finding the differences between previous month reading
and current month reading.
6. By using Scanner class get the input during runtime.
7. Create object for a class in memory and assign it to the reference variable, then the method
is invoked.
8. Finally, the bill amount is displayed based on type of connection.
```

PROGRAM:

//File Name should be Saved as ***Ebbill.java***

```
import java.io.*;
import java.util.*;
class ElectricityBill
{
     double bill;
     double domesticbillcalc (int units)
     {
         if(units<100)
           bill = units * 1 ;
          else if(units <= 200)
           bill = 100 * 1 + (units - 100) * 2.50 ;
         else if(units <= 500)
           bill = 100 * 1 + 200 * 2.50 + (units - 200) * 4 ;
         else
            bill = 100 * 1 + 200 * 2.50 + 500 * 4 + (units - 500) * 6 ;
         return bill;
     }
     double commercialbillcalc (int units)
     {
         if(units<100)
           bill = units * 2 ;
         else if(units <= 200)
           bill = 100 * 1 + (units - 100) * 4.50 ;
         else if(units <= 500)
           bill = 100 * 1 + 200 * 4.50 + (units - 200) * 6 ;
         else
            bill = 100 * 1 + 200 * 4.50 + 500 * 6 + (units - 500) * 7 ;
         return bill;
     }
     void show(String ptype,String consno,String consname,int pmr,int cmr,int units)
     {
         System.out.println("Type of Connection : " + ptype);
         System.out.println("Consumer Number : " + consno);
         System.out.println("Customer Name : " + consname);
         System.out.println("Previous Month Reading  : " + pmr);
         System.out.println("Current Month Reading  : " + cmr);
         System.out.println("Units Consumed  : " + units);
     }
}

class Ebbill
{
   public static void main(String[] args)
   {
       Scanner c = new Scanner(System.in);
       System.out.println("Enter the Type of Connection :");
       String ptype=c.next();
       System.out.println("Enter the Consumer Number :");
       String consno=c.next();
       System.out.println("Enter the Consumer Name :");
       String consname=c.next();
       System.out.println("Enter the Previous Month Reading :");
       int pmr=c.nextInt();
```

```
        System.out.println("Enter the Current Month Reading :");
        int cmr=c.nextInt();
        int units =  cmr-pmr;
        ElectricityBill b = new ElectricityBill();
        if(ptype.equalsIgnoreCase("DOMESTIC"))
        {
         b.show(ptype,consno,consname,pmr,cmr,units);
         b.domesticbillcalc(units);
         System.out.println("Bill to pay : " + b.bill);
         }
       else if(ptype.equalsIgnoreCase("COMMERCIAL"))
        {
         b.show(ptype,consno,consname,pmr,cmr,units);
         b.commercialbillcalc(units);
         System.out.println("Bill to pay : " + b.bill);
        }
  }
}
```

NOTE:

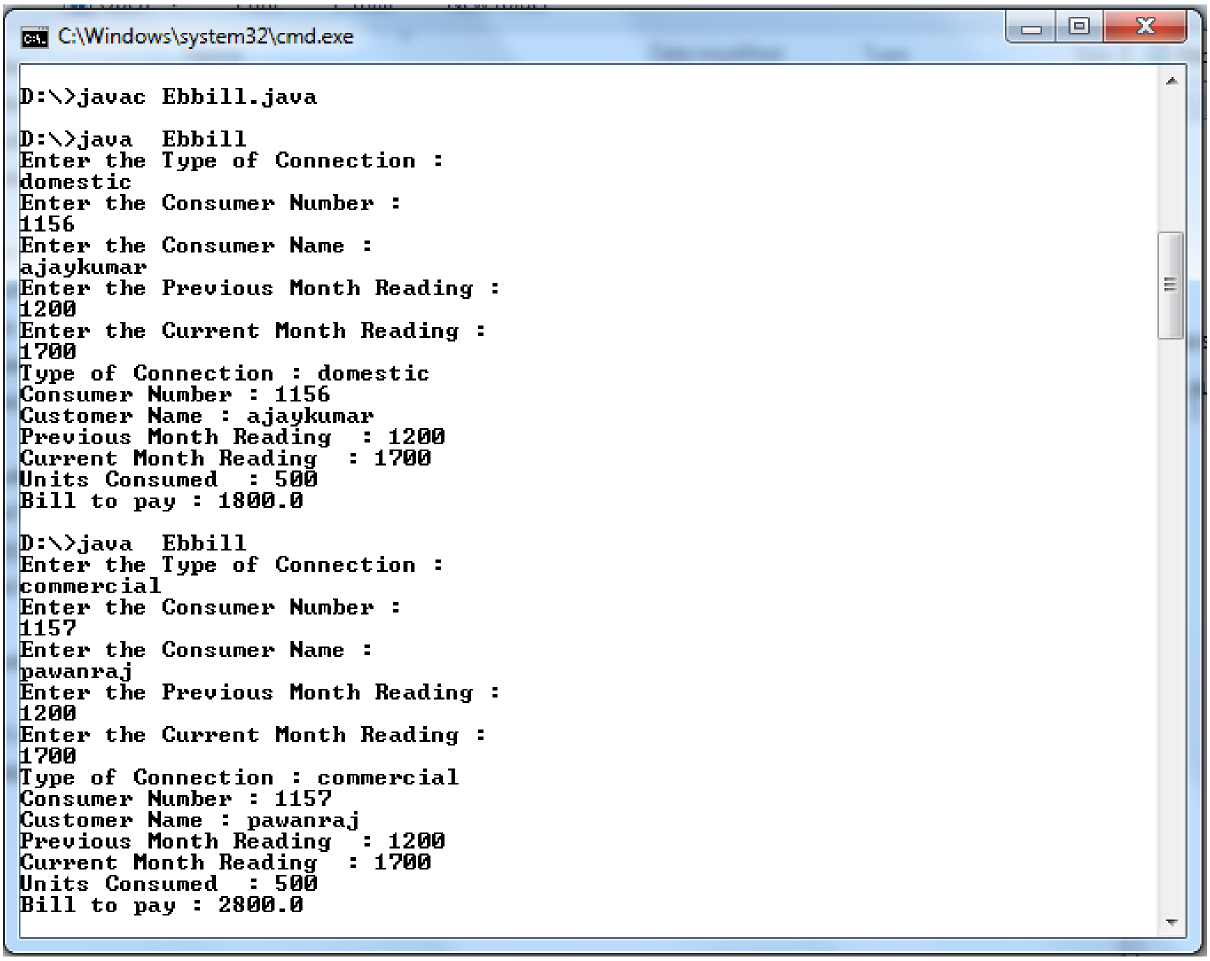
To Compile,

javac Ebbill.java

To Run

java Ebbill

OUTPUT:



```
D:\>javac Ebbill.java

D:\>java  Ebbill
Enter the Type of Connection :
domestic
Enter the Consumer Number :
1156
Enter the Consumer Name :
ajaykumar
Enter the Previous Month Reading :
1200
Enter the Current Month Reading :
1700
Type of Connection : domestic
Consumer Number : 1156
Customer Name : ajaykumar
Previous Month Reading  : 1200
Current Month Reading  : 1700
Units Consumed  : 500
Bill to pay : 1800.0

D:\>java  Ebbill
Enter the Type of Connection :
commercial
Enter the Consumer Number :
1157
Enter the Consumer Name :
pawanraj
Enter the Previous Month Reading :
1200
Enter the Current Month Reading :
1700
Type of Connection : commercial
Consumer Number : 1157
Customer Name : pawanraj
Previous Month Reading  : 1200
Current Month Reading  : 1700
Units Consumed  : 500
Bill to pay : 2800.0
```

RESULT:

Thus the application for generating Electricity bill has been successfully executed.

Viva questions:

1. How to calculate electricity bill?
2. How to calculate the Previous Month Reading?
3. How to calculate the Current Month Reading?
4. How to calculate the domesticbill?
5. How to calculate the commercialbill?

CURRENCY CONVERTER, DISTANCE CONVERTER AND TIME CONVERTER USING PACKAGES

AIM:

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

ALGORITHM:

1. The package keyword is used to create a package in java.
2. Create a class CurrencyConverter inside a package name CurrencyConverter.
3. Class also contains methods dollortoinr, inrtodollor, eurotoinr, inrtoeuro, yentoinr, and inrtoyen with its parameters to convert given currency.
4. Create a class DistanceConverter inside a package name DistanceConverter.
5. Class also contains methods metertokm, kmtometer, milestokm and kmtomiles with its parameters to convert given distance.
6. Create a class TimeConverter inside a package name TimeConverter.
7. Class also contains methods hourstominutes, minutestohours, hourstoseconds and secondstohours with its parameters to convert given time.
8. Import the CurrencyConverter, DistanceConverter, TimeConverter and other java packages.
9. Create a class Converter and object for a class in memory and assign it to the reference variable, then the method is invoked.
10. By using Scanner class get the choices for switch statement during runtime.
11. By using switch case statement we can convert currency, distance and time for each choice.
12. Create object for a class in memory and assign it to the reference variable, then the method is invoked.
13. Finally, the conversion is displayed based on type of converter.

PROGRAM:

//For Packages, Folder Name should be ***CurrencyConverter***
//File Name should be ***CurrencyConverter.java***

```
package CurrencyConverter;
public class CurrencyConverter
{
public double dollortoinr(double x)
{
double inr=x*67.86;
return inr;
}
public double inrtodollor(double x)
{
```

```
double dollor=x/67.86;
return dollor;
}
public double eurotoinr(double x)
{
double inr=x*79.18;
return inr;
}
public double inrtoeuro(double x)
{
double euro=x/79.18;
return euro;
}
public double yentoinr(double x)
{
double inr=x*0.62;
return inr;
}
public double inrtoyen(double x)
{
double yen=x/0.62;
return yen;
}
}
```

//For Packages, Folder Name should be ***DistanceConverter***
//File Name should be ***DistanceConverter.java***

```
package DistanceConverter;
public class DistanceConverter
{
public double metertokm(double x)
{
double km=x*0.001;
return km;
}
public double kmtometer(double x)
{
double meter=x/0.001;
return meter;
}
public double milestokm(double x)
{
double km=x*1.60934;
return km;
}
public double kmtomiles(double x)
{
double miles=x/1.60394;
return miles;
}
}
```

*//For Packages, Folder Name should be **TimeConverter***
*//File Name should be **TimeConverter.java***

```
package TimeConverter;
public class TimeConverter
{
public double hourstominutes(double x)
{
double minutes=x*60;
return minutes;
}
public double minutestohours(double x)
{
double hours=x/60;
return hours;
}
public double hourstoseconds(double x)
{
double seconds=x*3600;
return seconds;
}
public double secondstohours(double x)
{
double hours=x/3600;
return hours;
}
}
```

*//File Name should be **Converter.java** separate this file from above 3 folders*

```
import CurrencyConverter.*;
import DistanceConverter.*;
import TimeConverter.*;
import java.io.*;
import java.util.*;
```

```
class Converter
{
public static void main(String args[])
{
System.out.println("1.CurrencyConverter");
System.out.println("2.DistanceConverter");
System.out.println("3.TimeConverter");
Converter cr = new Converter();
Scanner c = new Scanner(System.in);
int choice = c.nextInt();
String op = null;
switch(choice)
{
case 1: cr.Currency(); break;
case 2: cr.Distance(); break;
case 3: cr.Time(); break;
default:
System.out.println("Invalid case");
return;
}
}

public void Currency()
{
  Scanner in = new Scanner(System.in);
  System.out.println("Welcome to Currency Converter");
  System.out.println("Enter the amount :");
  double amt = in.nextInt();
  CurrencyConverter cc = new CurrencyConverter();
  System.out.println("DOLLOR="+amt+" is INR="+cc.dollortoinr(amt));
  System.out.println("INR="+amt+" is DOLLOR="+cc.inrtodollor(amt));
  System.out.println("EURO="+amt+" is INR="+cc.eurotoinr(amt));
  System.out.println("INR="+amt+" is EURO="+cc.inrtoeuro(amt));
  System.out.println("YEN="+amt+" is INR="+cc.yentoinr(amt));
  System.out.println("INR="+amt+" is YEN="+cc.inrtoyen(amt));
}

public void Distance()
{
  Scanner in = new Scanner(System.in);
  System.out.println("Welcome to Distance Converter");
  System.out.println("Enter the distance :");
  double dis = in.nextInt();
  DistanceConverter dd = new DistanceConverter();
  System.out.println("METER="+dis+" is KM="+dd.metertokm(dis));
  System.out.println("KM="+dis+" is METER="+dd.kmtometer(dis));
  System.out.println("MILES="+dis+" is KM="+dd.milestokm(dis));
  System.out.println("KM="+dis+" is MILES="+dd.kmtomiles(dis));
}

public void Time()
{
  Scanner out = new Scanner(System.in);
  System.out.println("Welcome to Time Converter");
  System.out.println("Enter the time :");
  double tim = out.nextInt();
  TimeConverter tt = new TimeConverter();
  System.out.println("HOURS="+tim+" is MINUTES="+tt. hourstominutes(tim));
  System.out.println("MINUTES="+tim+" is HOURS="+tt.minutestohours(tim));
  System.out.println("HOURS="+tim+" is SECONDS="+tt.hourstoseconds(tim));
  System.out.println("SECONDS="+tim+" is HOURS="+tt.secondstohours(tim));
}
}
```

NOTE:

To Compile, go to ***CurrencyConverter*** *folder*

```
javac CurrencyConverter.java
```

To Compile, go to ***DistanceConverter*** *folder*

```
javac DistanceConverter.java
```

To Compile, go to ***TimeConverter*** *folder*

```
javac TimeConverter.java
```

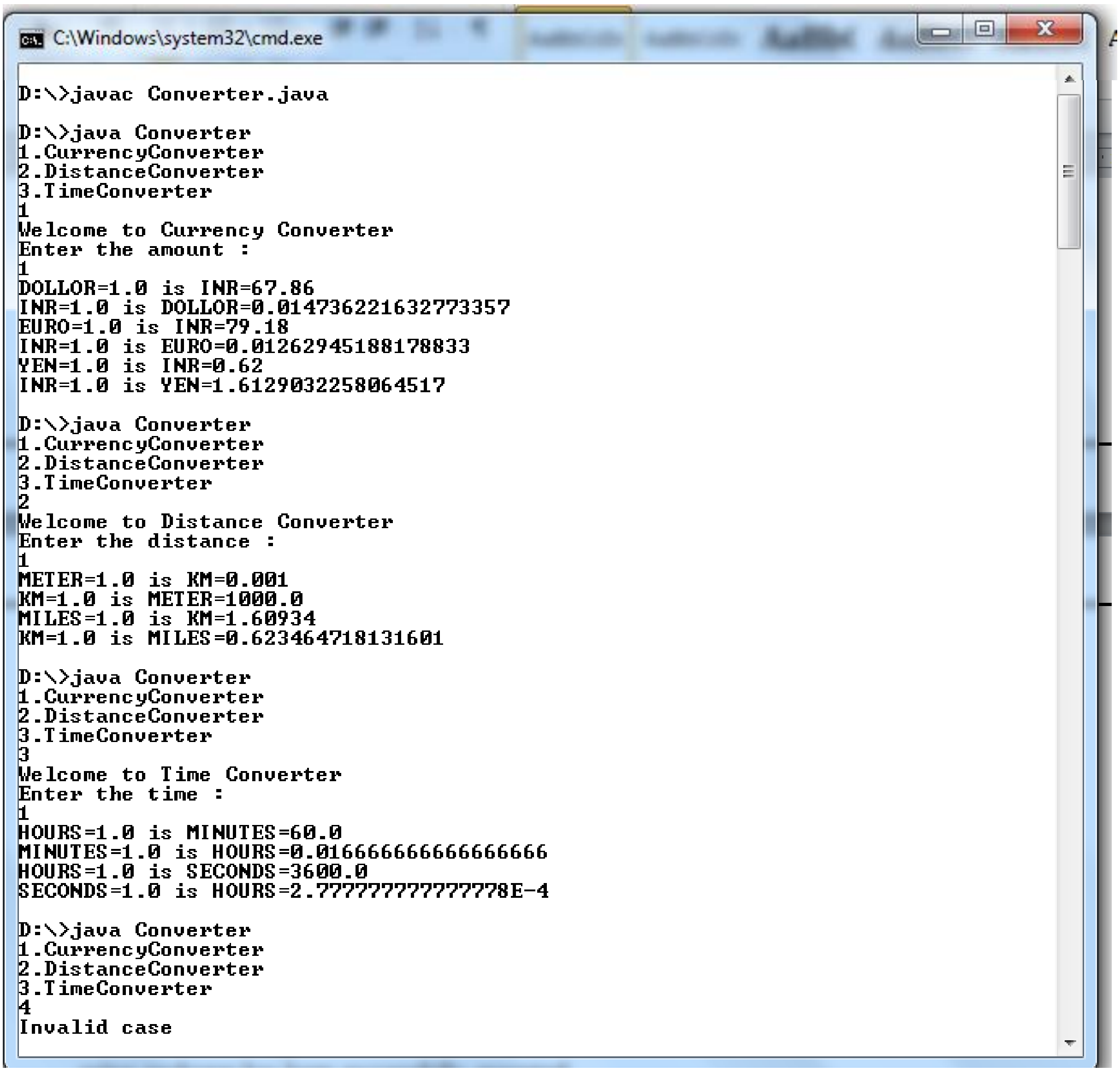
To Compile,

```
javac Converter.java
```

To Run

```
java Converter
```

OUTPUT:



```
D:\>javac Converter.java

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
1
Welcome to Currency Converter
Enter the amount :
1
DOLLOR=1.0 is INR=67.86
INR=1.0 is DOLLOR=0.014736221632773357
EURO=1.0 is INR=79.18
INR=1.0 is EURO=0.012629451881788833
YEN=1.0 is INR=0.62
INR=1.0 is YEN=1.6129032258064517

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
2
Welcome to Distance Converter
Enter the distance :
1
METER=1.0 is KM=0.001
KM=1.0 is METER=1000.0
MILES=1.0 is KM=1.60934
KM=1.0 is MILES=0.6234647181316 01

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
3
Welcome to Time Converter
Enter the time :
1
HOURS=1.0 is MINUTES=60.0
MINUTES=1.0 is HOURS=0.016666666666666666
HOURS=1.0 is SECONDS=3600.0
SECONDS=1.0 is HOURS=2.777777777777778E-4

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
4
Invalid case
```

RESULT:

Thus the application for currency converter, distance converter and time converter using packages has been successfully executed.

Viva questions:

1. How to create a package in java?

2. How to Create a class CurrencyConverter inside a package name CurrencyConverter?

3. How to Create a class DistanceConverter inside a package name DistanceConverter?

4. How to Create object for a class in memory?

5. How to Create a class TimeConverter inside a package name TimeConverter

GENERATING EMPLOYEE PAYROLL DETAILS

AIM:

To develop a java application for generating pay slips of employees with their gross and net salary.

ALGORITHM:

1. The package keyword is used to create a package in java.
2. Create a class Employee inside a package name employee.
3. Class Employee contains Emp_name, Emp_id, Address, Mail_id, Mobile_no as members.
4. By using Constructor initialize the instance variable of Employee class and display method is used to print employee details.
5. Create classes Programmer, AssistantProfessor, AssociateProfessor and Professor that extends Employee class and define necessary constructor for sub classes.
6. Each sub classes has its own instance variable like bPay and des.
7. Override the paySlip method in each sub classes to calculate the gross and net salary
8. By using super () method subclasses initialize the super class constructor.
9. Import employee package and create the object for Empolyee class.
10. Create different Employee object to add ArrayList<> classes.
11.DisplayEmployee method is used to display all employee playSlip details

PROGRAM:

//For Packages, Folder Name should be ***employee***
//File Name should be ***Employee.java***

```
package employee;
public class Employee
 {
   private String name;
   private String id;
   private String address;
   private String mailId;
   private String mobileNo;
   public Employee(String name, String id, String address, String mailId, String mobileNo)
  {
      this.name= name;
      this.id= id;
      this.address= address;
      this.mailId= mailId;
      this.mobileNo= mobileNo;
   }
```

```
     public void display()
    {
       System.out.println("Emp_Name : "+ name + "\t" + "Emp_id : "+ id);
       System.out.println("Address : " + address);
       System.out.println("Mail_id : "+ mailId + "\t" + "Mobile_no : " + mobileNo);
     }
    public void paySlip()
   {
    }
}
```

//For Packages, Folder Name should be ***employee***
//File Name should be ***Programmer.java***

```
package employee;
public class Programmer extends Employee
{
    private float bPay;
    private String des;
public Programmer(String name, String id, String address, String mailId, String mobileNo,
float bPay, String des)
 {
       super(name, id, address, mailId, mobileNo);
       this.bPay= bPay;
       this.des= des;
    }
    public void paySlip()
   {
       float da=bPay*97/100;
       float hra=bPay*10/100;
       double grossSalary=bPay + da + hra;
       float pf=bPay*12/100;
       double scf=bPay*0.1/100;
       double netSalary=grossSalary - pf - scf;
       System.out.println("------------ Employees Pay Slips ------------");
       super.display();
       System.out.println("Designation: "+des);
       System.out.println("Basic_Pay: "+bPay);
       System.out.println("Gross Salary : "+ grossSalary + "\t" + "Net Salary : " + netSalary);
       System.out.println("------------ End of the Statements -----------");
    }
}
```

```
//For Packages, Folder Name should be employee
// File Name should be AssistantProfessor.java

package employee;
public class AssistantProfessor extends Employee
{
   private float bPay;
   private String des;
public AssistantProfessor(String name, String id, String address, String mailId, String
mobileNo, float bPay, String des)
  {
      super(name, id, address, mailId, mobileNo);
      this.bPay= bPay;
      this.des= des;
      }
   public void paySlip()
     {
      float da=bPay*97/100;
      float hra=bPay*10/100;
      double grossSalary=bPay + da + hra;
      float pf=bPay*12/100;
      double scf=bPay*0.1/100;
      double netSalary=grossSalary - pf - scf;
      System.out.println("------------ Employees Pay Slips ------------");
      super.display();
      System.out.println("Designation: "+des);
      System.out.println("Basic_Pay: "+bPay);
      System.out.println("Gross Salary : "+ grossSalary + "\t" + "Net Salary : " + netSalary);
      System.out.println("------------ End of the Statements -----------");
   }
}

//For Packages, Folder Name should be employee
//File Name should be AssociateProfessor.java

package employee;
public class AssociateProfessor extends Employee
{
   private float bPay;
   private String des;
public AssociateProfessor(String name, String id, String address, String mailId, String
mobileNo, float bPay, String des)
{
      super(name, id, address, mailId, mobileNo);
```

```
        this.bPay= bPay;
        this.des= des;
    }
    public void paySlip()
    {
        float da=bPay*97/100;
        float hra=bPay*10/100;
        double grossSalary=bPay + da + hra;
        float pf=bPay*12/100;
        double scf=bPay*0.1/100;
        double netSalary=grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: "+des);
        System.out.println("Basic_Pay: "+bPay);
        System.out.println("Gross Salary : "+ grossSalary + "\t" + "Net Salary : " + netSalary);
        System.out.println("------------ End of the Statements -----------");
    }
}
```

*//For Packages, Folder Name should be **employee***
*//File Name should be **Professor.java***

```
package employee;
public class Professor extends Employee
{
    private float bPay;
    private String des;
public Professor(String name, String id, String address, String mailId, String mobileNo, float
bPay, String des)
{
        super(name, id, address, mailId, mobileNo);
        this.bPay= bPay;
        this.des= des;
    }
  public void paySlip()
   {
        float da=bPay*97/100;
        float hra=bPay*10/100;
        double grossSalary=bPay + da + hra;
        float pf=bPay*12/100;
        double scf=bPay*0.1/100;
        double netSalary=grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: "+des);
        System.out.println("Basic_Pay: "+bPay);
        System.out.println("Gross Salary : "+ grossSalary + "\t" + "Net Salary : " + netSalary);
        System.out.println("------------ End of the Statements -----------");
    }
}
```

*//File Name should be **Emp.java** separate this file from above folder*

```
import employee.*;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
public class Emp
{
    Employee e;
    ArrayList<Employee> obj= new ArrayList<>();
    Scanner get= new Scanner(System.in);
    public void addEmployee()
   {
        System.out.println("Enter the Emp_Name:");
        String name = get.next();
        System.out.println("Enter the Emp_id:");
        String id = get.next();
        System.out.println("Enter the Address:");
        String address = get.next();
        System.out.println("Enter the Mail_id:");
        String mailId = get.next();
        System.out.println("Enter the Mobile_no:");
        String mobileNo = get.next();
        System.out.println("Enter the Designation:");
        String des = get.next();
        System.out.println("Enter the Basic_Pay:");
        float bPay = get.nextFloat();
        if(des.equalsIgnoreCase("Programmer"))
       {
            e= new Programmer(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        }
        else if(des.equalsIgnoreCase("AssistantProfessor"))
       {
            e= new AssistantProfessor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        }
        else if(des.equalsIgnoreCase("AssociateProfessor"))
       {
            e= new AssociateProfessor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        }
        else if(des.equalsIgnoreCase("Professor"))
       {
            e= new Professor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        }
    }
    public void displayEmployee()
    {
        for(Employee e:obj)
       {
            e.paySlip();
        }
    }
    public static void main(String args[]) throws IOException
   {
        Emp x= new Emp();
        String check;
        do
        {
           x.addEmployee();
           System.out.println("Do you wnat continue press 'y'");
           check=x.get.next();
        }
        while(check.equalsIgnoreCase("y"));
        x.displayEmployee();
    }
}
```

NOTE:

To Compile, go to ***employee*** *folder*

```
javac Employee.java
javac Programmer.java
javac AssistantProfessor.java
javac AssociateProfessor.java
javac Professor.java
```

To Compile, it should be outside the package

```
javac Emp.java
```

To Run

```
java Emp
```

OUTPUT:

```
D:\>javac Emp.java

D:\>java  Emp
Enter the Emp_Name:
Suresh
Enter the Emp_id:
E708
Enter the Address:
cuddalore
Enter the Mail_id:
suresh708@tgarments.org
Enter the Mobile_no:
7894561230
Enter the Designation:
Programmer
Enter the Basic_Pay:
7500
Do you wnat continue press 'y'
y
Enter the Emp_Name:
Rakesh
Enter the Emp_id:
E705
Enter the Address:
pondy
Enter the Mail_id:
rakesh@gmail.com
Enter the Mobile_no:
4567891230
Enter the Designation:
Professor
Enter the Basic_Pay:
15000
Do you wnat continue press 'y'
y
Enter the Emp_Name:
kumar
Enter the Emp_id:
E405
Enter the Address:
madurai
Enter the Mail_id:
```

```
kumarat@ymail.com
Enter the Mobile_no:
1237894560
Enter the Designation:
AssistantProfessor
Enter the Basic_Pay:
18000
Do you wnat continue press 'y'
y
Enter the Emp_Name:
Naresh
Enter the Emp_id:
E102
Enter the Address:
villupuram
Enter the Mail_id:
nar12@rediffmail.com
Enter the Mobile_no:
9873214560
Enter the Designation:
AssociateProfessor
Enter the Basic_Pay:
20000
Do you wnat continue press 'y'
n
```

```
------------ Employees Pay Slips ------------
Emp_Name : Suresh       Emp_id : E708
Address : cuddalore
Mail_id : suresh708@tgarments.org       Mobile_no : 7894561230
Designation: Programmer
Basic_Pay: 7500.0
Gross Salary : 15525.0  Net Salary : 14617.5
------------ End of the Statements -----------
------------ Employees Pay Slips ------------
Emp_Name : Rakesh       Emp_id : E705
Address : pondy
Mail_id : rakesh@gmail.com      Mobile_no : 4567891230
Designation: Professor
Basic_Pay: 15000.0
Gross Salary : 31050.0  Net Salary : 29235.0
------------ End of the Statements -----------
------------ Employees Pay Slips ------------
Emp_Name : kumar        Emp_id : E405
Address : madurai
Mail_id : kumarat@ymail.com     Mobile_no : 1237894560
Designation: AssistantProfessor
Basic_Pay: 18000.0
Gross Salary : 37260.0  Net Salary : 35082.0
------------ End of the Statements -----------
------------ Employees Pay Slips ------------
Emp_Name : Naresh       Emp_id : E102
Address : villupuram
Mail_id : nar12@rediffmail.com  Mobile_no : 9873214560
Designation: AssociateProfessor
Basic_Pay: 20000.0
Gross Salary : 41400.0  Net Salary : 38980.0
------------ End of the Statements -----------

D:\>
```

RESULT:

Thus the application for generating pay slips of employees with their gross and net salary has been successfully executed.

Viva questions:

1. How to create a package in java?
2. How to Create a class Employee inside a package name employee?
3. How to Create classes Programmer?
4. How to initialize the instance variable of Employee class?
5. How to Compile the outside package?

DESIGN A JAVA INTERFACE FOR ADT STACK

AIM:

To Design a Java interface for ADT Stack and implement this interface using array, provide necessary exception handling in the implementation.

ALGORITHM:

1. Import the java packages.
2. Design an interface for MyStack with functions push, pop and display.
3. Define a class StackArray to implement the MyStack using array.
4. Define the functions of the interface accordingly and handle the stack overflow and underflow exceptions.
5. Create a class StackAdt and object for a class StackArray in memory and assign it to the reference variable, then the method is invoked.
6. By using Scanner class get the choices for switch statement during runtime.
7. By using switch case statement we can push, pop and display the elements for each choice.

PROGRAM:

*//File Name should be **StackAdt.java***

```
import java.util.Scanner;
interface MyStack
{
   public void pop();
   public void push();
   public void display();
}

class StackArray implements MyStack
{
   final static int n=5;
   int stack[]=new int[n];
   int top=-1;

  public void push()
   {
      Scanner in;
      try
      {
         in=new Scanner(System.in);
         if(top==(n-1))
```

```
           {
               System.out.println(" Stack Overflow");
               return;
           }
           else
           {
               System.out.println("Enter the element");
               int ele=in.nextInt();
               stack[++top]=ele;
           }
       }
       catch(Exception e)
       {
           System.out.println("e");
       }
   }

public void pop()
   {
       if(top<0)
       {
           System.out.println("Stack underflow");
           return;
       }
       else
       {
           int popper=stack[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+"  "+stack[i]+" -->";
```

```
            System.out.println("Elements are:"+str);
        }
    }
}

class StackAdt
{
    public static void main(String arg[])
    {
        Scanner in= new Scanner(System.in);
        System.out.println("Implementation of Stack using Array");
        StackArray stk=new StackArray();
        int ch=0;
        do
        {
            System.out.println("1.Push 2.Pop 3.Display 4.Exit");
            System.out.println("Enter your choice:");
            ch=in.nextInt();
            switch(ch)
            {
            case 1:
                stk.push();
                break;
            case 2:
                stk.pop();
                break;
            case 3:
                stk.display();
                break;
            case 4:
                System.exit(0);
            }
        }
        while(ch<4);
    }
}
```

NOTE:

To Compile,

javac StackAdt.java

To Run

java StackAdt

OUTPUT:

```
D:\>javac StackAdt.java

D:\>java  StackAdt
Implementation of Stack using Array
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
30
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
45
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
Enter the element
55
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
1
 Stack Overflow
```

```
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:   10 -->  20 -->  30 -->  45 -->  55 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped element:55
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:   10 -->  20 -->  30 -->  45 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped element:45
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:   10 -->  20 -->  30 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped element:30
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:   10 -->  20 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped element:20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:   10 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped element:10
```

```
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Stack is empty

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Stack is empty

1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
4
D:\>
```

RESULT:

Thus the Implementation for ADTStack interface using array has been successfully executed.

Viva questions:

1. How will you Import the java packages?
2. How will you a Create a class StackAdt and object for a class StackArray in memory?
3. How will you Create a class StackAdt and object for a class StackArray in memory?
4. How will you define a class StackArray to implement the MyStack using array.?
5. How will you define the functions of the interface accordingly and handle the stack overflow and underflow exceptions?

STRING OPERATIONS USING ARRAYLIST

AIM:

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

ALGORITHM:

1. Import the java packages.
2. Define a class ArrayList and perform following functions:
   a. Append - add at end
   b. Insert - add at particular index
   c. Search
   d. List all string starts with given letter
3. Create an object for ArrayList to add string elements.
4. By using ArrayList Method – add the elements are added into the Array List, the new element gets added after the last element unless the index is specified.
5. The elements in ArrayList are displayed.
6. Insert the specified element at the specified position index in this list.
7. After inserting the elements in ArrayList, the elements are displayed.
8. Search an object in ArrayList whether it is listed under this instance or not.
9. Finally List all string starts with given letter in ArrayList and displays the elements.

PROGRAM:

//File Name should be ***ArrayList.java***

```
import java.util.*;
public class ArrayList
{
  public static void main(String args[])
  {
          ArrayList<String> obj = new ArrayList<String>();

         /* Append - add at end */
          obj.add("Arun");
          obj.add("Balaji");
          obj.add("Chandru");
          obj.add("David");
          obj.add("Elango");

       System.out.println("\n"+"Elements in ArrayList:");
       System.out.print("\t"+obj+" "+"\n" + "\n");
```

```
           /* Insert - add at particular index */
            obj.add(0, "Arujun");
            obj.add(1, "Barath");

        System.out.println("After Inserting Elements:"+"");
        System.out.print("\t"+obj+" "+"\n" + "\n");

        /* Searching Elements in ArrayList */
            System.out.println("Search For Element:");
            Scanner in = new Scanner(System.in);
            String searchStr=in.nextLine();

        boolean ans = obj.contains(searchStr);

        if (ans)
         System.out.println("\t"+"ArrayList contains" +searchStr+ "\n");
        else
         System.out.println("ArrayList does not contains "+searchStr);

            /* List all string starts with given letter in ArrayList */
        System.out.println("Arraylist get the strings starting with given letter:");

        ArrayList<String> obj1 = new ArrayList<String>();

        String start= in.next();
        for(int i=0;i<obj.size();i++)
         {
          if(obj.get(i).startsWith(start.toUpperCase()))
           {
            obj1.add(obj.get(i));
           }
         }
         System.out.print("\t"+obj1+" "+"\n" + "\n");
   }
}
```

NOTE:

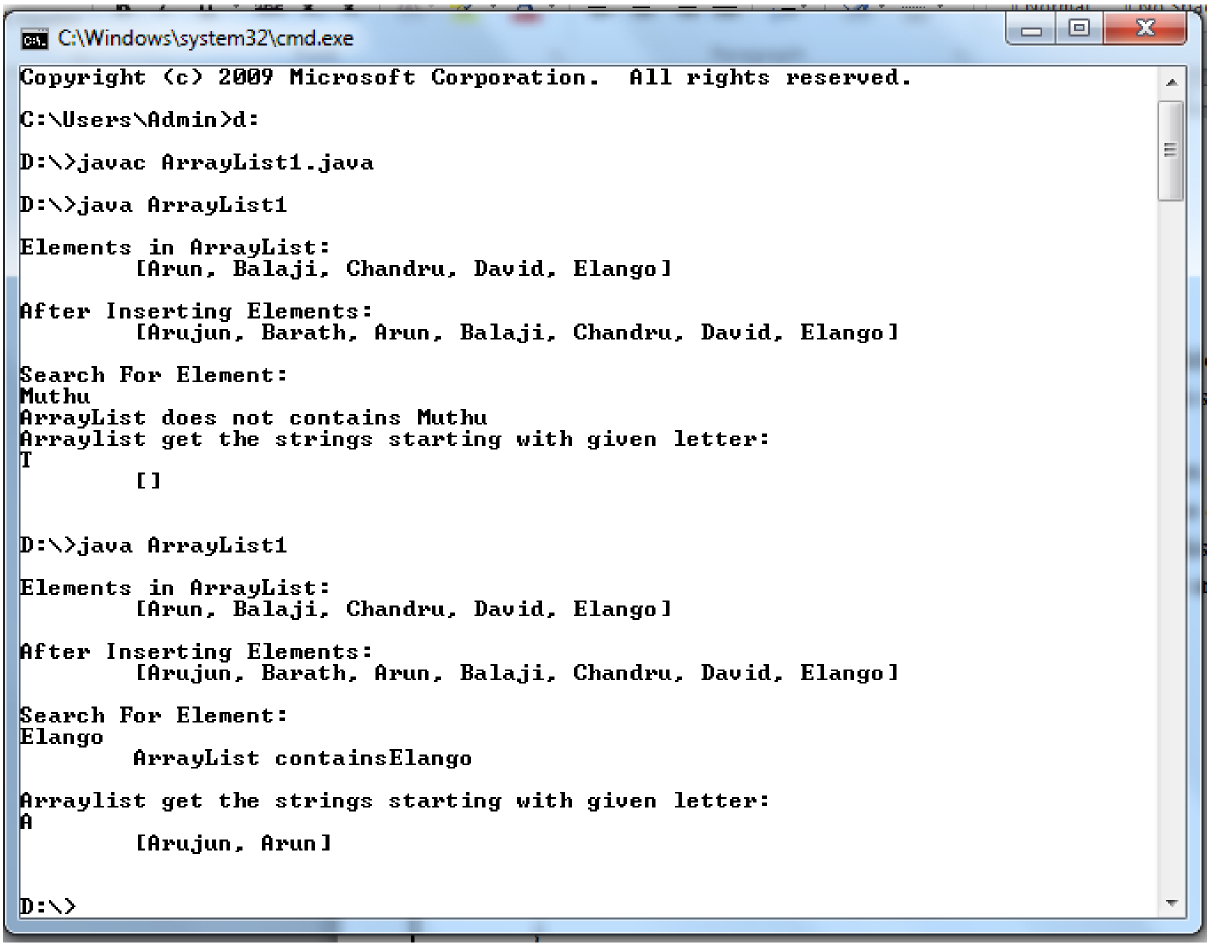
To Compile:

javac ArrayList.java

To Run:

java ArrayList

OUTPUT:



```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Admin>d:

D:\>javac ArrayList1.java

D:\>java ArrayList1

Elements in ArrayList:
        [Arun, Balaji, Chandru, David, Elango]

After Inserting Elements:
        [Arujun, Barath, Arun, Balaji, Chandru, David, Elango]

Search For Element:
Muthu
ArrayList does not contains Muthu
Arraylist get the strings starting with given letter:
T
        []


D:\>java ArrayList1

Elements in ArrayList:
        [Arun, Balaji, Chandru, David, Elango]

After Inserting Elements:
        [Arujun, Barath, Arun, Balaji, Chandru, David, Elango]

Search For Element:
Elango
        ArrayList containsElango

Arraylist get the strings starting with given letter:
A
        [Arujun, Arun]


D:\>
```

RESULT:

Thus the Implementation for string operations using ArrayList has been successfully executed.

Viva questions:

1. How to Import the java packages.?
2. How will you define a class ArrayList?
3. How will Create an object for ArrayList to add string elements?
4. How will perform the Append - add at end functions?
5. How will perform the Insert - add at particular index?

FINDING THE AREA OF DIFFERENT SHAPES

AIM:

To write a java program to find the area of different shapes by using abstract class.

ALGORITHM:

1. Import the java packages.
2. Create an abstract class named Shape that contains two integers and an empty method named printArea().
3. Create a class Rectangle that extends the class Shape. Override the method printArea () by getting Width and Length then compute the area and prints the area of the Rectangle.
4. Create a class Triangle that extends the class Shape. Override the method printArea () by getting Base and Height then compute the area and prints the area of the Triangle.
5. Create a class Circle that extends the class Shape. Override the method printArea () by getting the Radius, then compute the area and prints the area of the Circle.
6. By using Scanner class get the input during runtime.
7. Create object for a class in memory and assign it to the reference variable, then the method is invoked.

PROGRAM:

//File Name should be ***Area.java***

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

abstract class Shape
{
   double a = 0.0, b = 0.0;
   abstract public void printArea();
}

class Rectangle extends Shape
{
 double area = 0.0;
 public void printArea()
 {
   System.out.println("Area of Rectangle");
   System.out.println("----------");
   Scanner in = new Scanner(System.in);
   System.out.println("Enter the Width:");
   this.a = in.nextDouble();
   System.out.println("Enter the Length:");
```

```
    this.b = in.nextDouble();
    this.area = a*b;   /*  (width*length)   */
    System.out.println("The area of rectangle is:"+this.area);
 }
}

class Triangle extends Shape
{
 double area = 0.0;
 public void printArea()
 {
    System.out.println("-----Area of Triangle-----");
    System.out.println("----------");
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the Base:");
    this.a = in.nextDouble();
    System.out.println("Enter the Height:");
    this.b = in.nextDouble();
    this.area = 0.5*a*b;    /*  1/2 (base*height)  */
    System.out.println("The area of triangle is:"+this.area);
 }
}

class Circle extends Shape
{
 double area = 0.0;
 public void printArea()
 {
    System.out.println("-----Area of Circle-----");
    System.out.println("----------");
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the Radius:");
    this.a = in.nextDouble();
    this.area = 3.14*a*a;   /*  3.14*r*r   */
    System.out.println("The area of circle is:"+this.area);
 }
}

public class Area
{
  public static void main(String[] args)
  {
      System.out.println("-----Finding the Area of Shapes:-----");
       Shape s;
       s=new Rectangle();
```

```
        s.printArea();
        s=new Triangle();
        s.printArea();
        s=new Circle();
        s.printArea();
    }
}
```

NOTE:

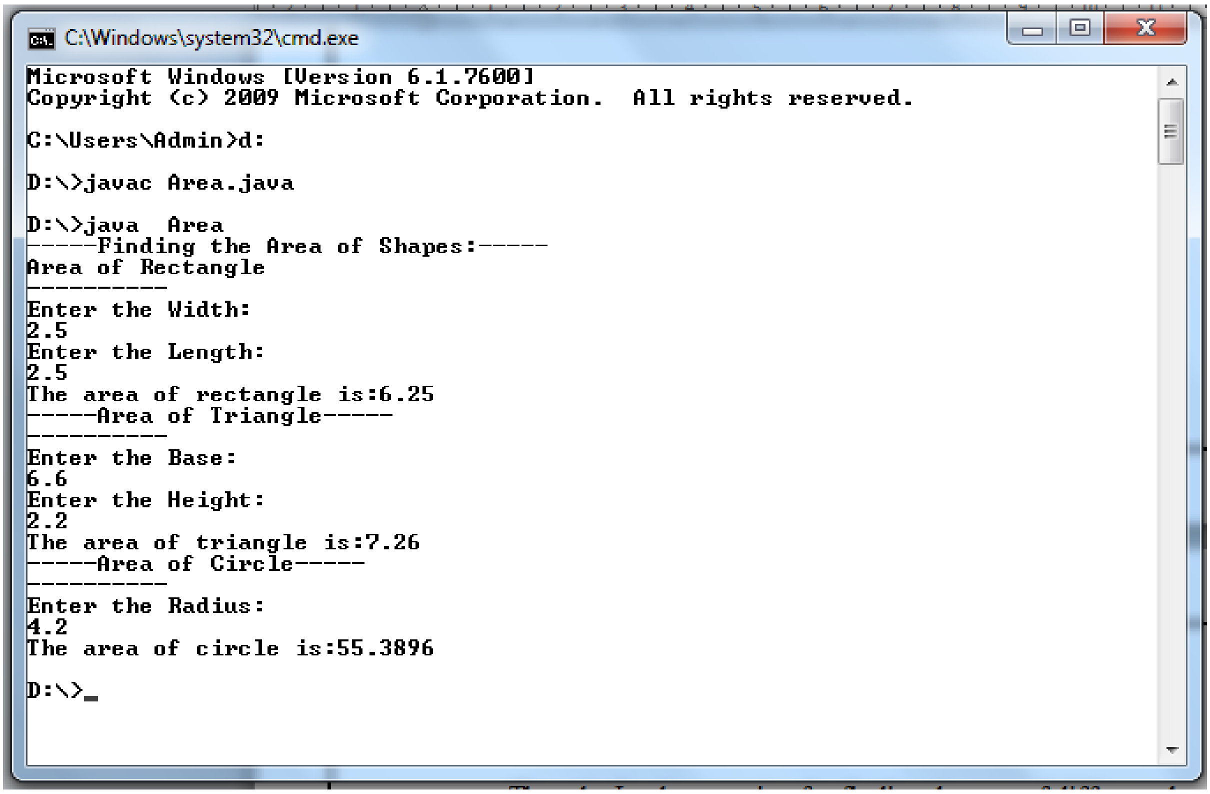
To Compile:

```
javac Area.java
```

To Run:

```
java Area
```

OUTPUT:



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Admin>d:

D:\>javac Area.java

D:\>java  Area
-----Finding the Area of Shapes:-----
Area of Rectangle
----------
Enter the Width:
2.5
Enter the Length:
2.5
The area of rectangle is:6.25
-----Area of Triangle-----
----------
Enter the Base:
6.6
Enter the Height:
2.2
The area of triangle is:7.26
-----Area of Circle-----
----------
Enter the Radius:
4.2
The area of circle is:55.3896

D:\>_
```

RESULT:

Thus the Implementation for finding the area of different shapes using abstract class has been successfully executed.

Viva questions:

1.How will you Import the java packages?

2. How will you Create an abstract class name?

3. How will you Create a class Rectangle, Triangle, Circle?

4. How will you get the input during runtime?

5. . How will you Create object for a class in memory?

CREATING OWN EXCEPTIONS

AIM:

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

ALGORITHM:

1. Import the java packages.
2. Create a subclass of Exception named as MyException it has only a constructor plus an overloaded toString ( ) method that displays the value of the exception.
3. The exception is thrown when compute ( ) integer parameter is greater than 10.
4. The main ( ) method sets up an exception handler for MyException, then calls compute ( ) with a legal value (less than 10) and an illegal one to show both paths through the code.

PROGRAM:

//File Name should be ***UserException.java***

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

class MyException extends Exception
{
   private int d;
   MyException(int a)
  {
   d = a;
  }

public String toString()
  {
   return "MyException [" + d + "]";
  }
}

class UserException
{
 static void compute(int a) throws MyException
{
             System.out.println ("Called Compute(" + a + ")");
             if(a>10)
                         throw new MyException(a);
             System.out.println ("Normal Exit");
 }
 public static void main(String args[])
  {
    try
    {
      compute(1);
      compute(20);
    }
    catch(MyException e)
   {
    System.out.println("Caught " + e);
    }
  }
}
```

NOTE:

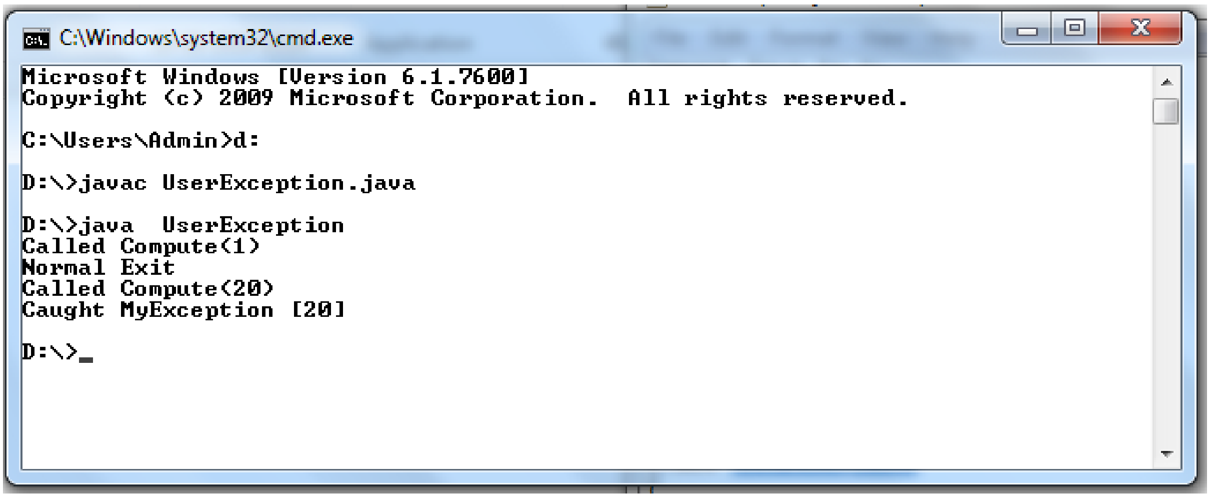
To Compile:

javac UserException.java

To Run:

java UserException

OUTPUT:



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Admin>d:

D:\>javac UserException.java

D:\>java  UserException
Called Compute(1)
Normal Exit
Called Compute(20)
Caught MyException [20]

D:\>_
```

RESULT:

Thus the Implementation for user defined exception handling has been successfully executed.

Viva questions:

1. How will you Import the java packages.?
2. How to Create a subclass of Exception named as MyException?
3. When the exception is thrown?
4. How the main ( ) method sets up an exception handler?
5. Define exception handling in java?

GETTING FILE INFORMATION

AIM:

To write a java program to implement file information such as 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.

ALGORITHM:

1. Import the java packages.
2. By using Scanner class get the input during runtime.
3. By using File class method create a File object associated with the file or directory specified by pathname. The pathname can contain path information as well as a file or directory name.
4. The exists() checks whether the file denoted by the pathname exists. Returns true if and only if the file denoted by the pathname exists; false otherwise
5. The getAbsolutePath() returns the absolute pathname string of the pathname.
6. The canRead() checks whether the application can read the file denoted by the pathname. Returns true if and only if the file specified by the pathname exists and can be read by the application; false otherwise.
7. The canWrite() checks whether the application can modify to the file denoted by the pathname. Returns true if and only if the file system actually contains a file denoted by the pathname and the application is allowed to write to the file; false otherwise.
8. The length() returns the length of the file denoted by the pathname. The return value is unspecified if the pathname denotes a directory.
9. The endsWith() returns true if the given string ends with the string given as argument for the method else it returns false.
10. The program uses conditional operator to check different functionalities of the given file.

PROGRAM:

*//File Name should be **FileInfo.java***

```
import java.io.*;
import java. util.*;
public class FileInfo
{
    public static void main(String[] args) throws IOException
   {
      Scanner in=new Scanner(System.in);

      System.out.print("\nEnter the FileName: ");
      String fName = in.next();
```

```
        File f = new File(fName);
        String result = f.exists() ? " exists." : " does not exist.";
        System.out.println("\nThe given file " +fName  + result);

        System.out.println("\nFile Location: "+f.getAbsolutePath());

        if(f.exists())
        {
           result = f.canRead() ? "readable." : "not readable.";
           System.out.println("\nThe file is " + result);

           result = f.canWrite() ? "writable." : "not writable.";
           System.out.println("\nThe file is " + result);

           System.out.println("\nFile length is " + f.length() + " in bytes.");

           if (fName.endsWith(".jpg") || fName.endsWith(".gif") || fName.endsWith(".png"))
           {
              System.out.println("\nThe given file is an image file.");
           }
           else if (fName.endsWith(".pdf"))
           {
              System.out.println("\nThe given file is an portable document format.");
           }
           else if (fName.endsWith(".txt"))
           {
              System.out.println("\nThe given file is a text file.");
           }
           else
           {
              System.out.println("The file type is unknown.");
           }
        }
    }
}
```

NOTE:

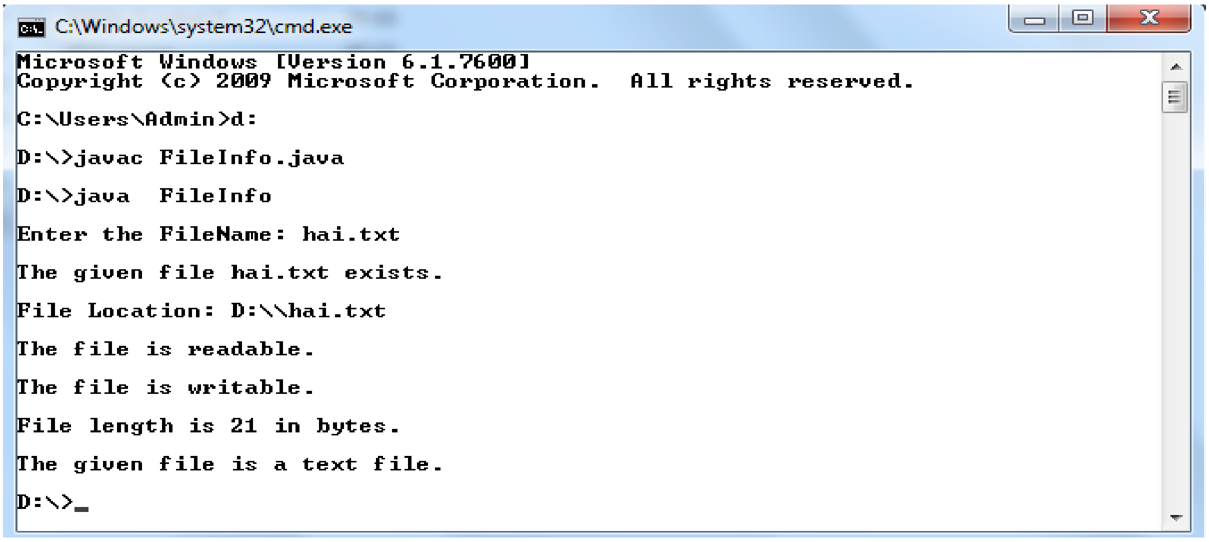
To Compile:

```
javac FileInfo.java
```

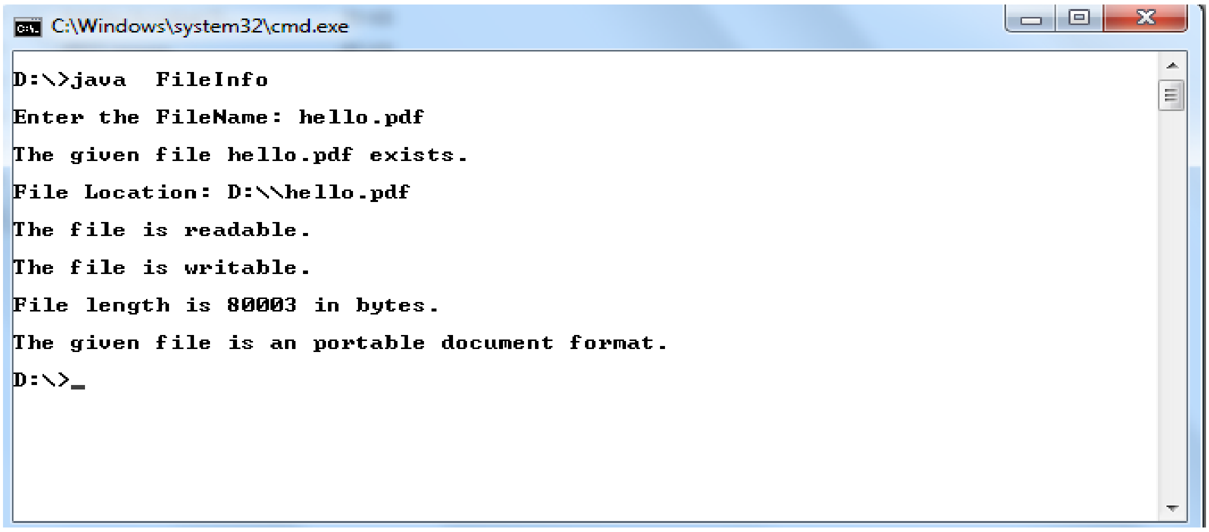
To Run:

```
java FileInfo
```

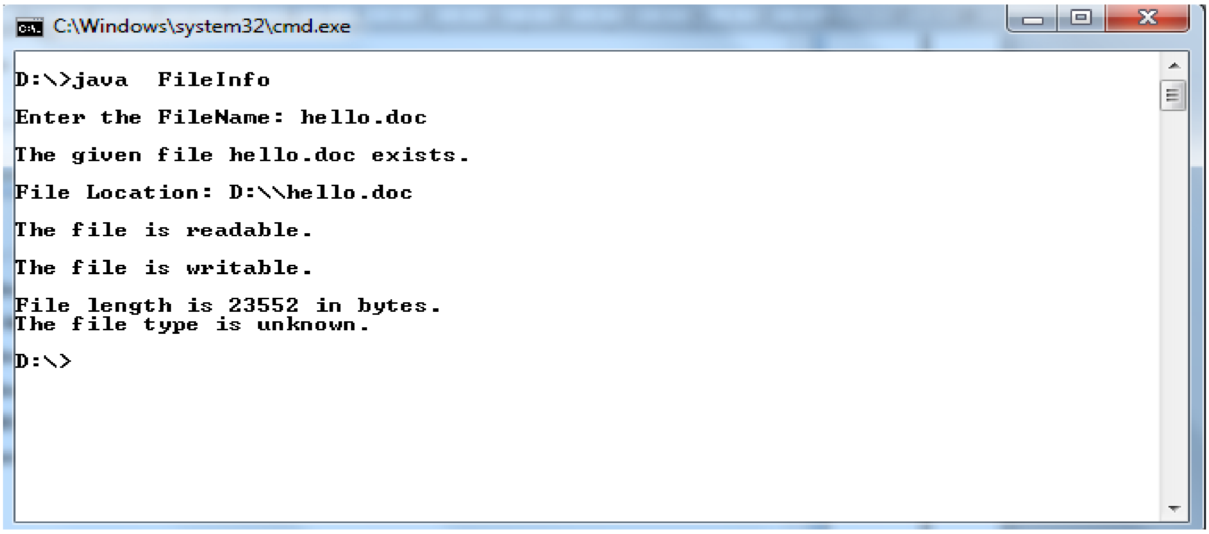
OUTPUT:



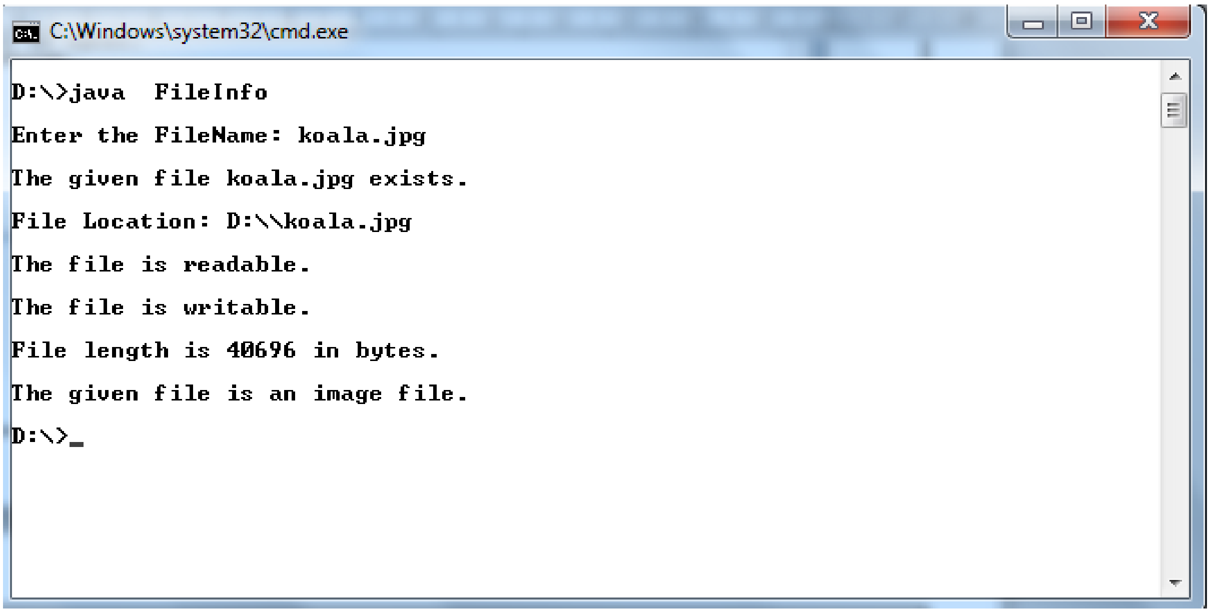
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Admin>d:

D:\>javac FileInfo.java

D:\>java  FileInfo

Enter the FileName: hai.txt

The given file hai.txt exists.

File Location: D:\\hai.txt

The file is readable.

The file is writable.

File length is 21 in bytes.

The given file is a text file.

D:\>_
```



```
D:\>java  FileInfo

Enter the FileName: hello.pdf

The given file hello.pdf exists.

File Location: D:\\hello.pdf

The file is readable.

The file is writable.

File length is 80003 in bytes.

The given file is an portable document format.

D:\>_
```



```
D:\>java  FileInfo

Enter the FileName: hello.doc

The given file hello.doc exists.

File Location: D:\\hello.doc

The file is readable.

The file is writable.

File length is 23552 in bytes.
The file type is unknown.

D:\>
```



```
D:\>java  FileInfo
Enter the FileName: koala.jpg
The given file koala.jpg exists.
File Location: D:\\koala.jpg
The file is readable.
The file is writable.
File length is 40696 in bytes.
The given file is an image file.
D:\>_
```

RESULT:

Thus the Implementation for getting file information has been successfully executed.

Viva questions:

1. How will you Import the java packages.?
2. How to write a java program to implement file information?
3. How to create a File object associated with the file?
4. How the the program uses conditional operator to check different functionalities of the given file.?
5. How will you get the input during runtime?

MULTI THREADED APPLICATION

AIM:

To write a 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.

ALGORITHM:

1. Import the java packages.
2. Create a thread that generates random number, Obtain one random number and check is odd or even.
3. If number is even then create and start thread that computes square of a number, Compute number * number and display the answer.
4. Notify to Random number thread and goto step 7.
5. If number is odd then create and start thread that computes cube of a number, Compute number * number * number and display the answer.
6. Notify to Random number thread and goto step 7.
7. Wait for 1 Second and Continue to Step 3 until user wants to exits.

PROGRAM:

//File Name should be ***Multithread.java***

```
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 Generate 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 Generates Random Integer: " + 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 Multithread
{
public static void main(String[] args)
{
Generate g = new Generate();
g.start();
}
}
```

NOTE:

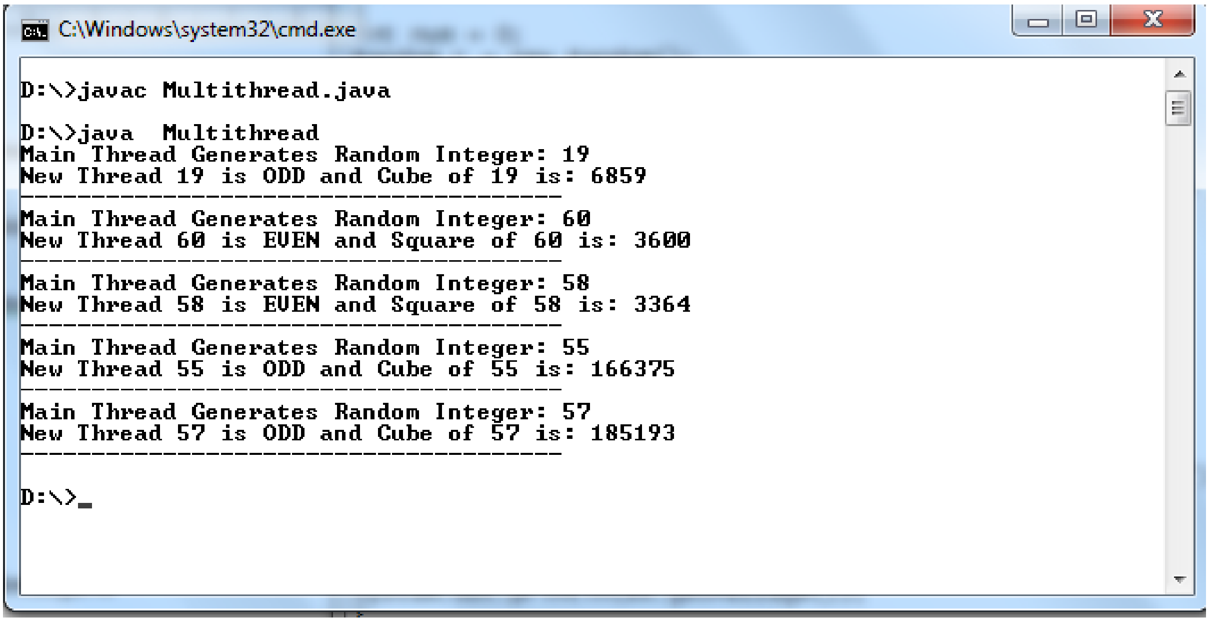
To Compile:

javac Multithread.java

To Run:

java Multithread

OUTPUT:



```
D:\>javac Multithread.java

D:\>java  Multithread
Main Thread Generates Random Integer: 19
New Thread 19 is ODD and Cube of 19 is: 6859
--------------------------------------
Main Thread Generates Random Integer: 60
New Thread 60 is EVEN and Square of 60 is: 3600
--------------------------------------
Main Thread Generates Random Integer: 58
New Thread 58 is EVEN and Square of 58 is: 3364
--------------------------------------
Main Thread Generates Random Integer: 55
New Thread 55 is ODD and Cube of 55 is: 166375
--------------------------------------
Main Thread Generates Random Integer: 57
New Thread 57 is ODD and Cube of 57 is: 185193
--------------------------------------

D:\>_
```

RESULT:

Thus the Implementation for application for multithreading has been successfully executed.

Viva questions:

1. How will you Import the java packages?
2. How to Create a thread?
3. How the thread generates a random integer?
4. How the thread computes the square of the number?
5. How the thread will print the value of cube of the number?

GENERIC PROGRAMMING

AIM:

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

ALGORITHM:

1. Import the java packages.
2. Comparable interface is used to order the objects of user-defined class.
3. This interface is found in java.lang package and contains only one method named compareTo(Object).
4. The compareTo() method works by returning an int value that is either positive, negative, or zero.
5. Create a generic method max(), that can accept any type of argument.
6. Then sets the first element as the max element, and then compares all other elements with the max element using compareTo() method
7. Finally the function returns an element which has the maximum value.
8. We can call generic method by passing with different types of arguments, the compiler handles each method.

PROGRAM:

//File Name should be ***MyGeneric.java***

```
import java.util.*;
class MyGeneric {
public static <T extends Comparable<T>> T max(T... elements)
{
   T max = elements[0];
   for (T element : elements) {
      if (element.compareTo(max) > 0)
      {
         max = element;
      }
   }
   return max;
 }

public static void main(String[] args)
{
    System.out.println("Integer Max: " + max(Integer.valueOf(32), Integer.valueOf(89)));
    System.out.println("String Max: "  + max("GaneshBabu", "Ganesh"));
    System.out.println("Double Max: "  + max(Double.valueOf(5.6), Double.valueOf(2.9)));
```

```
        System.out.println("Boolean Max: " + max(Boolean.TRUE, Boolean.FALSE));
        System.out.println("Byte Max: " + max(Byte.MIN_VALUE, Byte.MAX_VALUE));
        }
}
```

NOTE:

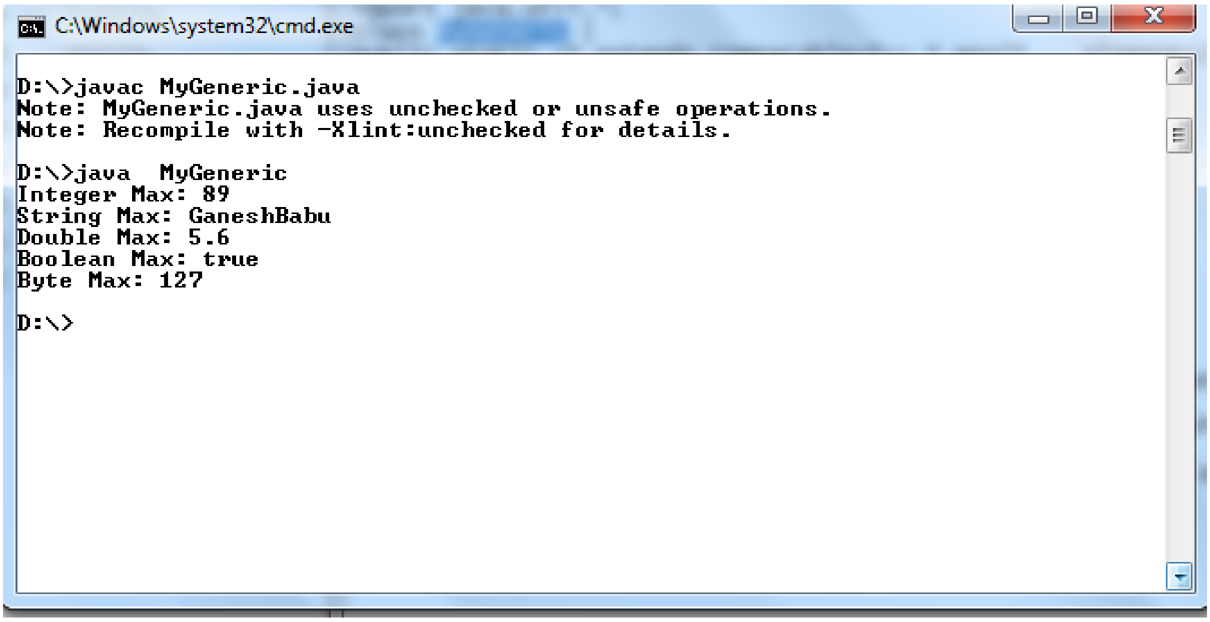
To Compile:

javac MyGeneric.java

To Run:

java MyGeneric

OUTPUT:



```
D:\>javac MyGeneric.java
Note: MyGeneric.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\>java  MyGeneric
Integer Max: 89
String Max: GaneshBabu
Double Max: 5.6
Boolean Max: true
Byte Max: 127

D:\>
```

RESULT:

Thus the Implementation for finding the maximum value from the given type of elements using a generic function has been successfully executed.

Viva questions:

1. How will you Import the java packages?
2. How will you the compare To() method works?
3. How will you Create a generic method max()?
4. How will you call the generic method?
5. How the compiler handles each method.

EVENT - DRIVEN PROGRAMMING

AIM:

To write a java program to design a calculator using event-driven programming paradigm of Java with the following options.

a) Decimal manipulations

b) Scientific manipulations

ALGORITHM:

1. Import the java packages.
2. Create the class calculator by implementing the class Jframe and interface actionListener.
3. Declare the buttons required using JButton.
4. Design the layout of the calculator using the setLayout,textpanel(), Panel(),Jtextfield(),setfont() methods.
5. Define the actions to be performed for each key using ActionListener.
6. Enable the scientific or standard calculator using the method method add().
7. Define the mathematical operations to be performed for the mathematical symbols.
8. Select the required mathematical operations using switch as the calculator.
9. Pass the parameters for the methods used.
10. Make the frame visible by using the method setVisible().

PROGRAM:

//File Name should be ***ScientificCalculator.java***

```
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;
        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, exp, plus, min, div, log, rec,
mul, eq, dot, 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;

    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("/");
div.addActionListener(this);
buttonpanel.add(div);

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);

        exp = new JButton("Exp");
        exp.addActionListener(this);
        buttonpanel.add(exp);

        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(".")) {
    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 {
        x = 0;
        y = 0;
        temp = Double.parseDouble(tfield.getText());
        tfield.setText("");
```

```
                ch = '-';
            }
            tfield.requestFocus();
        }

        if (s.equals("/")) {
            if (tfield.getText().equals("")) {
                tfield.setText("");
                temp = 1;
                ch = '/';
            } else {
                x = 0;
                y = 0;
                temp = Double.parseDouble(tfield.getText());
                ch = '/';
                tfield.setText("");
            }
            tfield.requestFocus();
        }

        if (s.equals("*")) {
            if (tfield.getText().equals("")) {
                tfield.setText("");
                temp = 1;
                ch = '*';
            } else {
                x = 0;
                y = 0;
                temp = Double.parseDouble(tfield.getText());
                ch = '*';
                tfield.setText("");
            }
            tfield.requestFocus();
        }

        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;
            }
        }
        tfield.requestFocus();
    }

public static void main(String args[]) {
try
{
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
}
catch (Exception e)
{
 }
    ScientificCalculator f = new ScientificCalculator();
    f.setTitle("ScientificCalculator");
    f.pack();
    f.setVisible(true);
  }
}
```

NOTE:

To Compile:

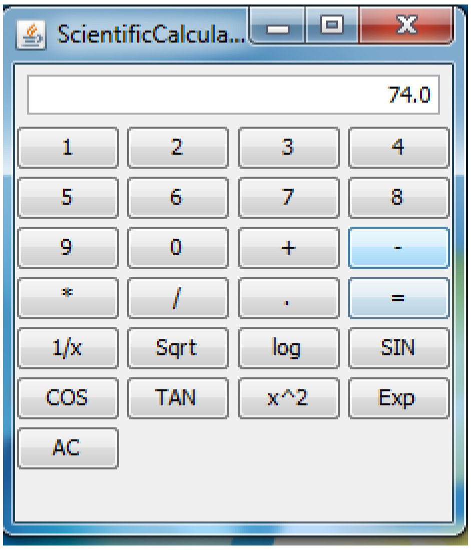
```
    javac ScientificCalculator.java
```

To Run:

```
    java ScientificCalculator
```

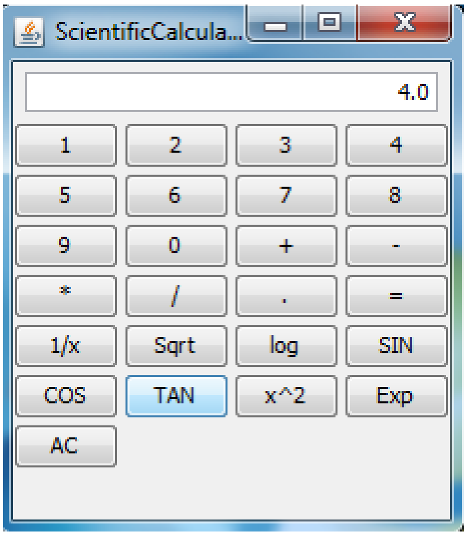
OUTPUT:

Addition [12+64]



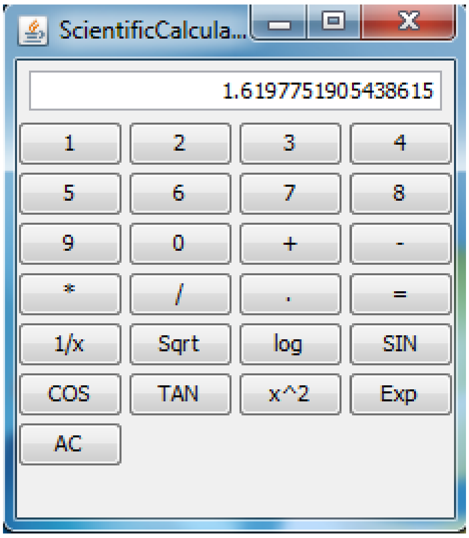
```
74.0
```

Square Root [16]



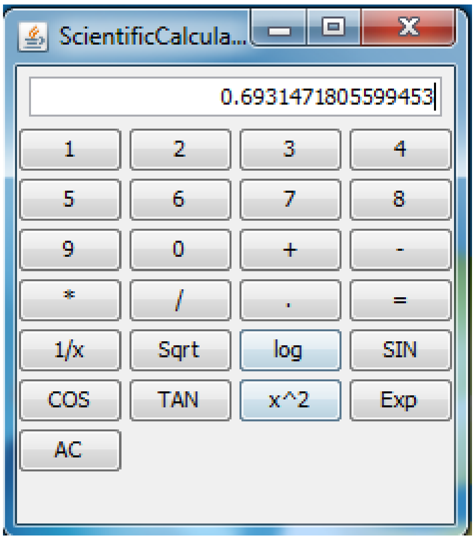
```
4.0
```

Tan 45



```
1.6197751905438615
```

Log 2



```
0.6931471805599453
```

RESULT:

Thus the Implementation for designing the scientific calculator has been successfully executed.

Viva questions:

1. How will you Import the java packages?
2. How will you Create the class calculator?
3. How will you Declare the buttons?
4. How will you Design the layout of the calculator?
5. How will you Pass the parameters for the methods?

MINI PROJECT - OPAC SYSTEM

AIM:

To develop a mini project OPAC system for library using Java concepts.

ALGORITHM:

1. Import the awt,swing packages.
2. Extend the JFrame which implements actionlistener to the class datas.
3. Create the textfield for id, name and button for next, address and the panel.
4. Create object for the getcontentpane().
5. Assign the length and breadth value for the layout using gridlayout.
6. Add the new labels for ISBN and book name.
7. Add the new button for the nextbook
8. Create the bookname under the driver jdbc odbc driver in the try block.
9. Create the object for exception as e and use it for catching the error.
10. Show all the records using showrecord.

PROGRAM:

//File Name should be ***Data.java***

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

  public class Data extends JFrame implements ActionListener
  {
    JTextField id;
    JTextField name;
    JButton next;
    JButton addnew;
    JPanel p;
    static ResultSet res;
    static Connection conn;
    static Statement stat;

   public Data()
   {
   super("My Application");
   Container c = getContentPane();
   c.setLayout(new GridLayout(5,1));

   id = new JTextField(20);
   name = new JTextField(20);
   next = new JButton("Next BOOK");
```

```
    p = new JPanel();

    c.add(new JLabel("ISBN Number",JLabel.CENTER));
    c.add(id);
    c.add(new JLabel("Book Name",JLabel.CENTER));
    c.add(name);
    c.add(p);

    p.add(next);
    next.addActionListener(this);
    pack();
    setVisible(true);
    addWindowListener(new WIN());
   }

public static void main(String args[])
  {
   Data d = new Data();
   try
    {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
     conn = DriverManager.getConnection("jdbc:odbc:stu");
     // cust is the DSN Name
     stat = conn.createStatement();
     res = stat.executeQuery("Select * from stu"); // stu is the table name
      res.next();
     }
    catch(Exception e)
     {
       System.out.println("Error" +e);
      }
        d.showRecord(res);
     }
      public void actionPerformed(ActionEvent e)
      {
          if(e.getSource() == next)
          {
            try
            {
               res.next();
            }
          catch(Exception e)
           {
            }
          showRecord(res);
        }
    }
     public void showRecord(ResultSet res)
     {
       try
```

```
           {
            id.setText(res.getString(2));
            name.setText(res.getString(3));
           }
       catch(Exception e)
        {
         }
}//end of the main

 //Inner class WIN implemented
class WIN extends WindowAdapter
 {
   public void windowClosing(WindowEvent w)
   {
     JOptionPane jop = new JOptionPane();
     jop.showMessageDialog(null,"Thank you","My
           Application",JOptionPane.QUESTION_MESSAGE);
   }
 }
}
```

NOTE:

Create a new Database

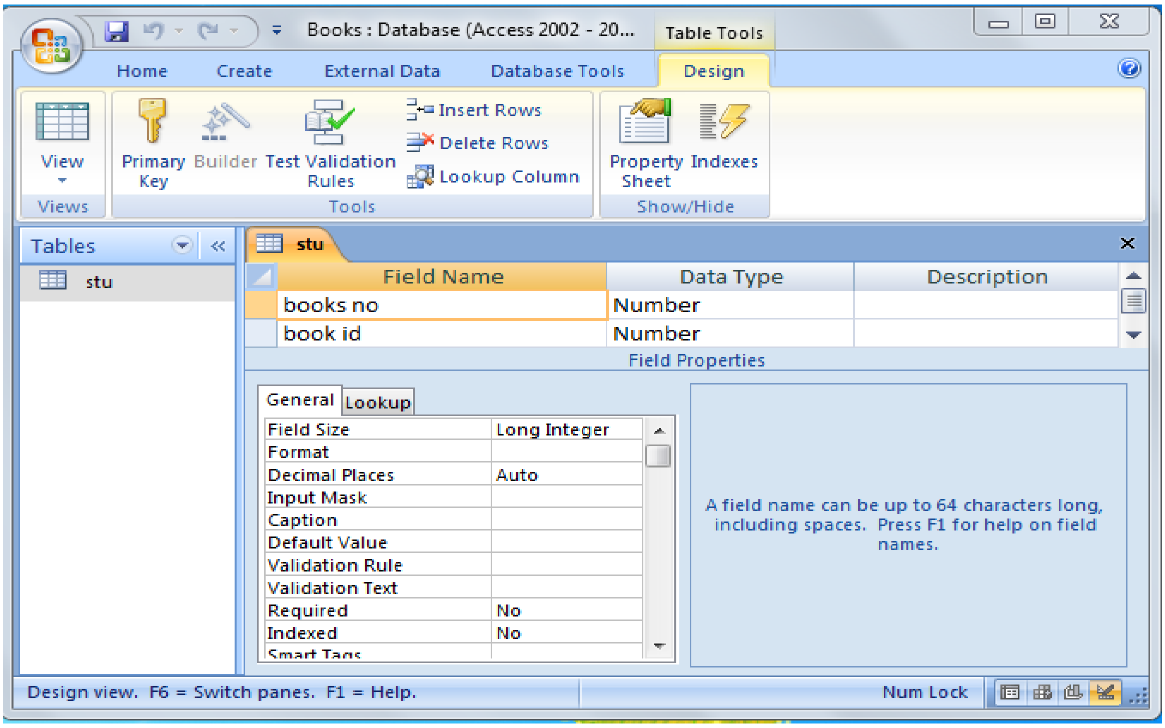
1. Create a new Database file in MS ACCESS (our backend) named **"books.mdb".**
2. Then create a table named **"stu"** in it.
3. The table **stu** contains the following fields and data types
   i. ISBN - Text
   ii. BookName - Text
4. Enter various records as you wish.
5. Save the database file.

Next step is to add our **"books.mdb"** to the **System DSN**. To do that follows the procedure given below,

i. Go to ***Start-> Control Panel -> Administrative tools***.
ii. In that double click ***"Data Sources (ODBC)"***.
iii. ***ODBC Data Source Administrator*** dialog appears.
iv. In that select ***"System DSN"*** tab and click the Add Button.
v. Select ***"Microsoft Access Driver(*.mdb)"*** and click Finish.
vi. ***ODBC Microsoft Access Setup*** appears. In the "Data Source name" type ***"stu"***.
vii. Click on the ***"Select"*** button and choose your database file. Then click ok.

Now your database file gets added to the System DSN.

Table: Design View



Field Name	Data Type	Description
books no	Number	
book id	Number	

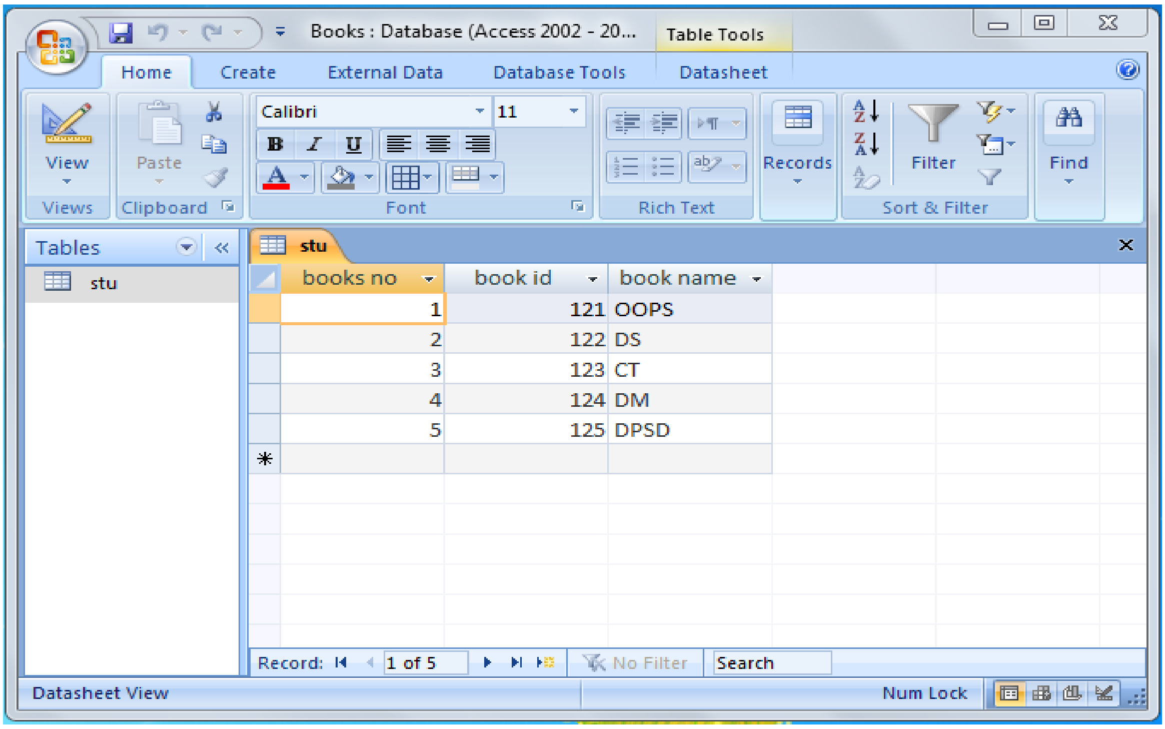
Field Properties

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

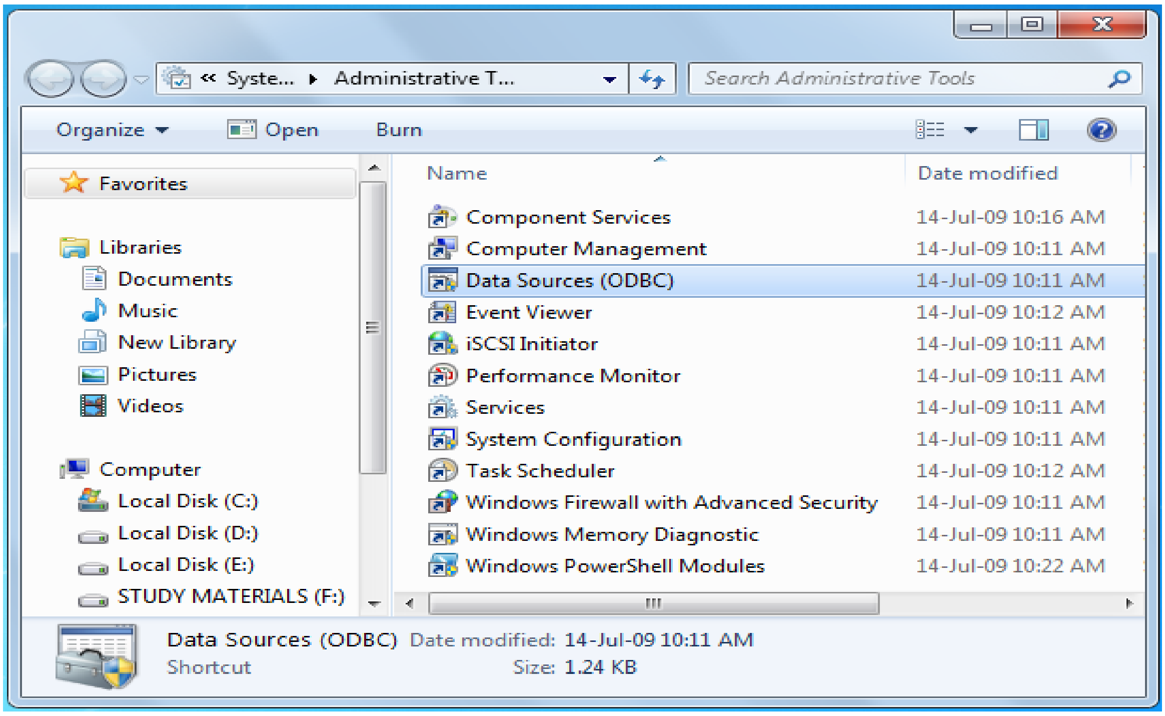
Table Name: stu



books no	book id	book name
1	121	OOPS
2	122	DS
3	123	CT
4	124	DM
5	125	DPSD
*		

Record: 1 of 5

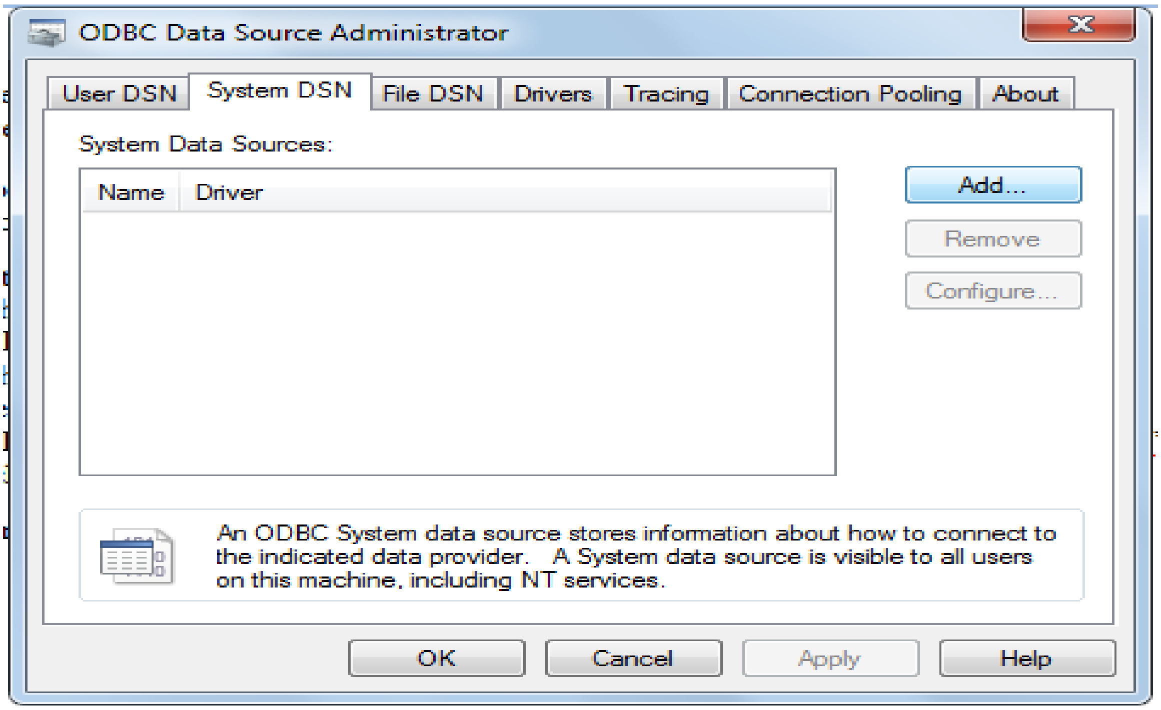
Administrative Tools.



Name	Date modified
Component Services	14-Jul-09 10:16 AM
Computer Management	14-Jul-09 10:11 AM
Data Sources (ODBC)	14-Jul-09 10:11 AM
Event Viewer	14-Jul-09 10:12 AM
iSCSI Initiator	14-Jul-09 10:11 AM
Performance Monitor	14-Jul-09 10:11 AM
Services	14-Jul-09 10:11 AM
System Configuration	14-Jul-09 10:11 AM
Task Scheduler	14-Jul-09 10:12 AM
Windows Firewall with Advanced Security	14-Jul-09 10:11 AM
Windows Memory Diagnostic	14-Jul-09 10:11 AM
Windows PowerShell Modules	14-Jul-09 10:22 AM

Data Sources (ODBC) Date modified: 14-Jul-09 10:11 AM
Shortcut Size: 1.24 KB

ODBC Data Source Administrator

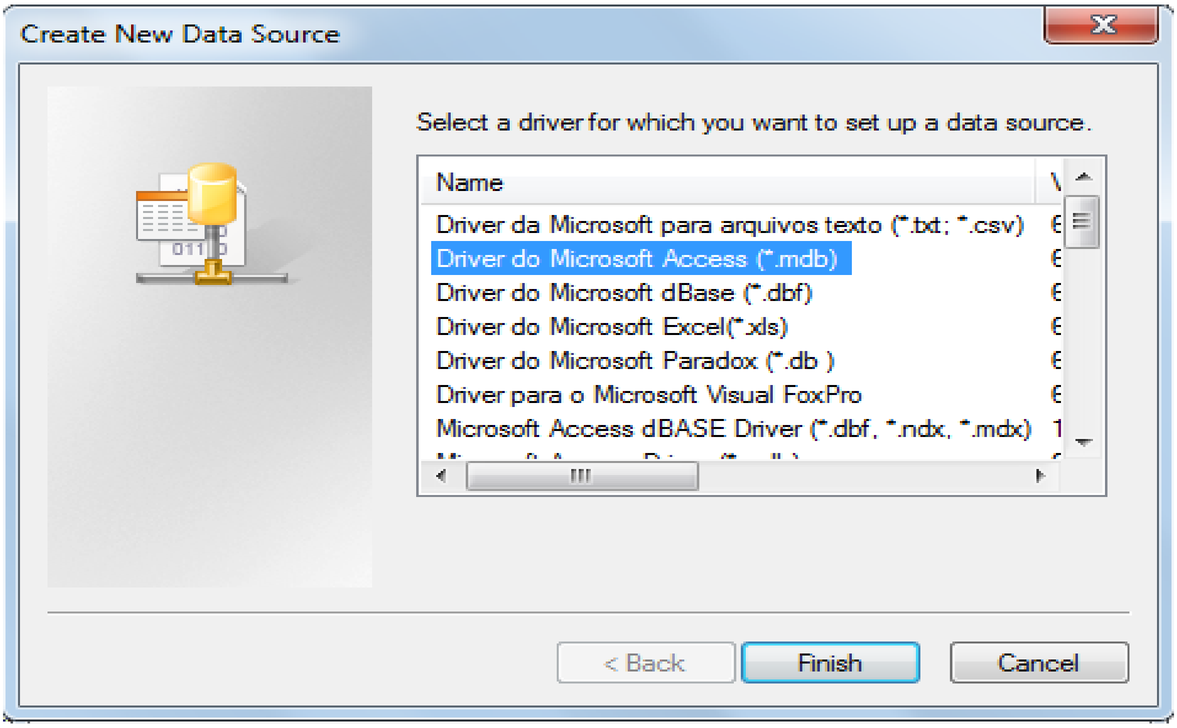


System Data Sources:

Name	Driver

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Creating Microsoft Access Driver(*.mdb)



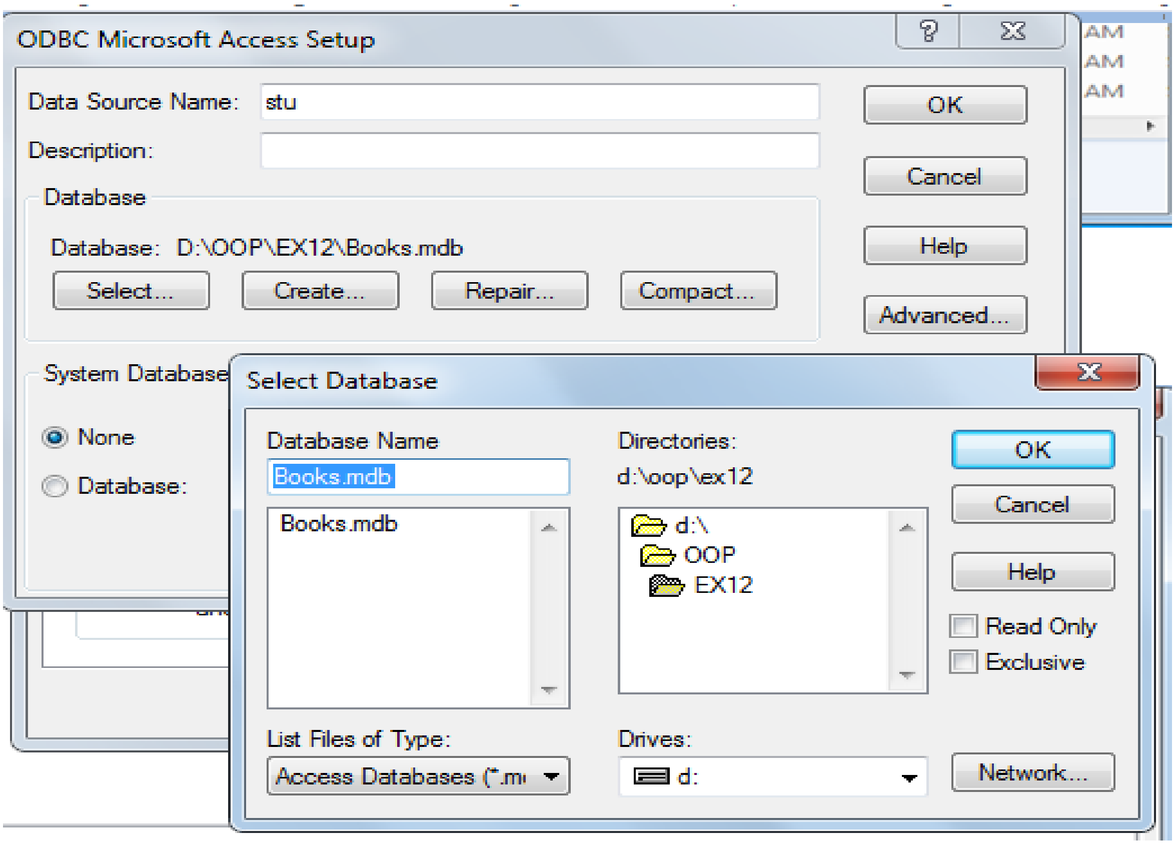
Create New Data Source

Select a driver for which you want to set up a data source.

Name

Driver da Microsoft para arquivos texto (*.txt; *.csv)
Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)
Driver do Microsoft Paradox (*.db)
Driver para o Microsoft Visual FoxPro
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)

ODBC Microsoft Access Setup



ODBC Microsoft Access Setup

Data Source Name: stu

Description:

Database

Database: D:\OOP\EX12\Books.mdb

System Database

● None
○ Database:

Select Database

Database Name: Books.mdb

Books.mdb

Directories: d:\oop\ex12

d:\
OOP
EX12

List Files of Type: Access Databases (*.mı

Drives: d:

□ Read Only
□ Exclusive

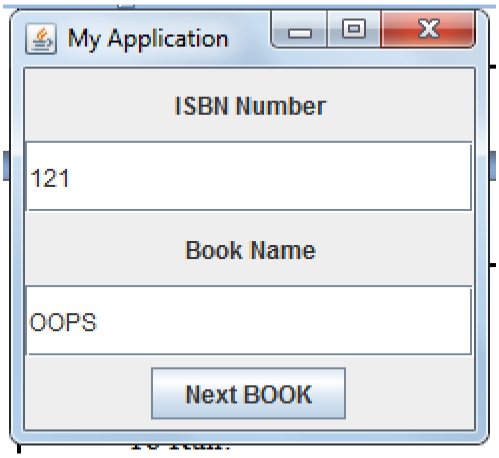
OUTPUT:

To Compile:

```
javac Data.java
```

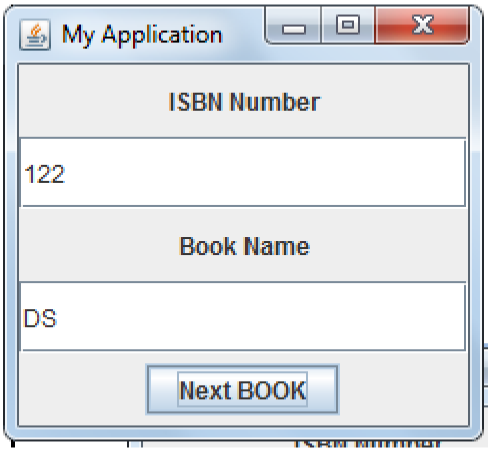
To Run:

```
java Data
```



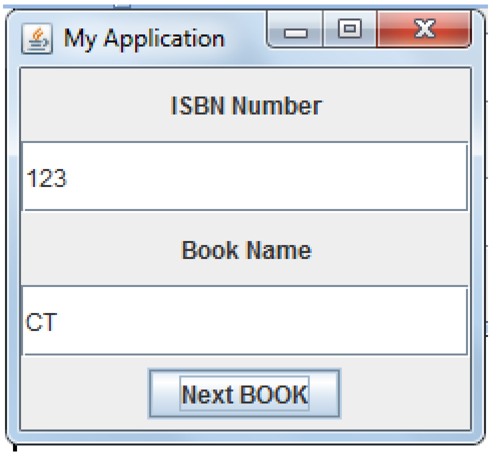
ISBN Number: 121

Book Name: OOPS



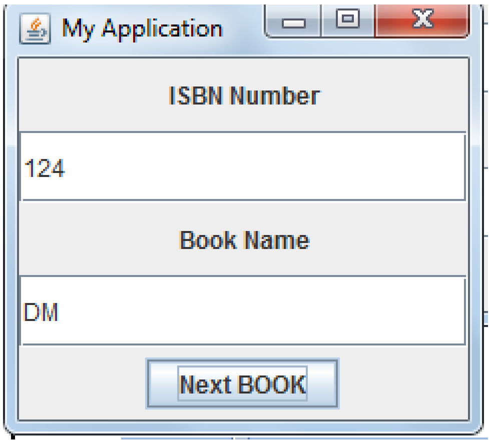
ISBN Number: 122

Book Name: DS



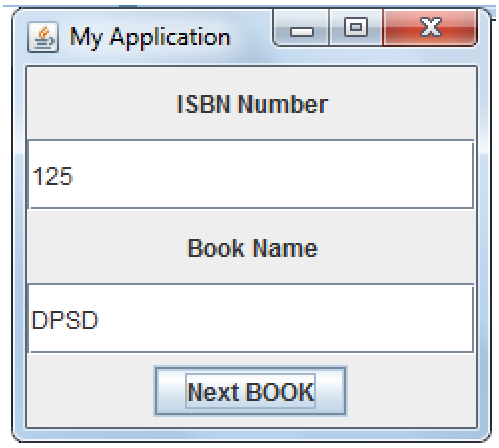
ISBN Number: 123

Book Name: CT



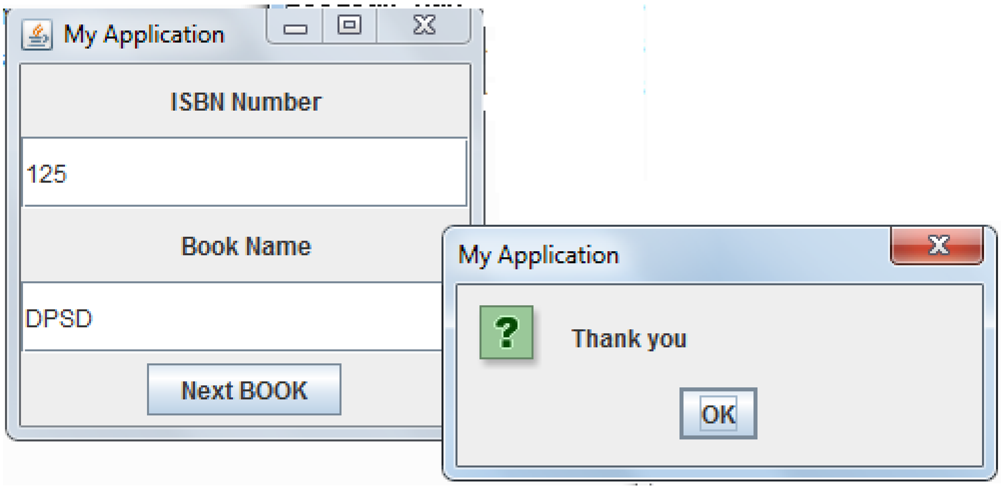
ISBN Number: 124

Book Name: DM



ISBN Number: 125

Book Name: DPSD



125

DPSD

Thank you

RESULT:

Thus the program to develop the simple OPAC for the libraries is executed successfully.

Viva questions:

1.How will you Import the awt,swing packages.?
2. How will you Extend the JFrame?
3. How will you Create the textfield?
4. How will you Create object?
5. How will you Assign the length and breadth value for the layout?