ARUNAI ENGINEERING COLLEGE

(Affiliated to Anna University)

Velu Nagar, Thiruvannamalai-606 603www.arunai.org

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

BACHELOR OF ENGINEERING

2021 - 2022

THIRD SEMESTER

CS8383 – OBJECT ORIENTED PROGRAMMING LAB

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, array list, 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, 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

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 Array List. 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

PROGRAMME OUTCOMES (POs)

After going through the four years of study, computer science & engineering graduates will exhibit :

	Graduate Attribute	Programme Outcome
1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
2	Problem analysis	Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understanding of the limitations.
6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10	Communication	Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

By the completion of Information Technology program the student will have following Programspecific outcomes.

1. Design secured database applications involving planning, development and maintenance usingstate of the art methodologies based on ethical values.
2. Design and develop solutions for modern business environments coherent with the advancedtechnologies and tools.
3. Design, plan and setting up the network that is helpful for contemporary business environmentsusing latest hardware components.
4. Planning and defining test activities by preparing test cases that can predict and correct errorsensuring a socially transformed product catering all technological needs.

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:

SOFTWARE:

- JAVA

HARDWARE:

Standalone desktops - 30 Nos. (or) Server supporting 30 terminals or more.

OUTCOMES:

At the end of the course, the students will be able to:

Course Outcomes	Description	Level in Bloom's Taxonomy
C204.1	Interpret Java programs using Object Oriented Programming principles	K2
C204.2	Explain Java programs with the concepts inheritance and interfaces	K2
C204.3	Contrast Java applications using exceptions and I/O streams	K2
C204.4	Relate Java applications with threads and generics classes	K2
C204.5	Develop interactive Java programs using swings	K2
C204.6	Demonstrate simple Graphical User Interfaces	K2

CO - PO MATRIX

Course Outcomes	Programme Outcome (POs)											
	K3	K4	K4	K5	K3,K4, K5	A3	A2	A3	A3	A3	A3	A2
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	-	-	-	-	-	-	-	-	-
CO2	2	1	1	-	-	-	-	-	-	-	-	-
CO3	2	1	1	-	-	-	-	-	-	-	-	-
CO4	2	1	1	-	-	-	-	-	-	-	-	-
CO5	2	1	1	-	-	-	-	-	-	-	-	-
CO6	2	1	1	-	-	-	-	-	-	-	-	-
	2	1	1	-	-	-	-	-	-	-	-	-

CO - PSO MATRIX

	PSO1	PSO2	PSO3
CO1	2	2	-
CO2	2	2	-
CO3	2	2	-
CO4	2	2	-
CO5	2	2	-
CO6	2	2	-
	2	2	-

MODE OF ASSESSMENT

EVALUATION PROCEDURE FOR EACH EXPERIMENT

S.No	Description	Mark
1.	Aim & Pre-Lab discussion	20
2.	Observation	20
3.	Conduction and Execution	30
4.	Output & Result	10
5.	Viva	20
Total		**100**

INTERNAL ASSESSMENT FOR LABORATORY

S.No	Description	Mark
1.	Observation	05
2.	Performance	05
3.	Viva voce	05
4.	Record	05
Total		**20**

ABOUT THE SOFTWARE

- Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable.
- Java Platform is a collection of programs that help programmers to develop and run Java programming applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications.

Java Development kit (JDK)

- JDK is a software development environment used for making applets and Java applications. The full form of JDK is Java Development Kit.
- Java developers can use it on Windows, macOS, Solaris, and Linux. JDK helps them to code and run Java programs. It is possible to install more than one JDK version on the same computer.

Java Virtual Machine (JVM):

- Java Virtual Machine (JVM) is an engine that provides a runtime environment to drive the Java Code or applications. It converts Java byte code into machine language. JVM is a part of the Java Run Environment (JRE).
- In other programming languages, the compiler produces machine code for a particular system. However, the Java compiler produces code for a Virtual Machine known as Java Virtual Machine.

Java Runtime Environment (JRE)

- JRE is a piece of software that is designed to run other software. It contains the class libraries, loader class, and JVM.
- In simple terms, if you want to run a Java program, you need JRE. If you are not a programmer, you don't need to install JDK, but just JRE to run Java programs.

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 uni t♣♣> 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 write a java program to develop a Java application to generate Electricity bill.

ALGORITHM:

1. Start the program.
2. Import scanner class to get input.
3. Create main class with the name ElectBill to develop java application for electricity bill generation.
4. Declare variables a, cno, cpr, ccr, cname, uc, amt to represent appropriate values.
5. Using Scanner class methods like nextDouble(), nextInt() and next() to get the input details from the user.
6. Calculate consumed units by subtracting current reading from previous reading.
7. Generate bill amount based on domestic or commercial using switch statement.
8. Finally print the consumer details with bill amount.
9. Stop the program.

PROGRAM:

```
//Java Program for Electricity Bill Generation
//import Scanner as we require it.
import java.util.Scanner;
//The name of our class its public
public class ElectBill
{
//void main
public static void main (String[] args)
{
//declare double
double uc, amt=0.01f;
int a,cno,cpr,ccr;
String cname;
//Declare input as scanner
Scanner input = new Scanner(System.in);
System.out.println("Enter Consumer no");
cno=input.nextInt();
System.out.println("Enter consumer name");
cname=input.next();
System.out.println("Enter previous month reading");
cpr=input.nextInt();
System.out.println("Enter current month reading");
ccr=input.nextInt();
uc=ccr-cpr;
System.out.println("Enter \n 1. Domestic \n 2. Commercial");
a = input.nextInt();
```

```
switch(a)
{
//Domestic EB Bill Calcualtion
//Fixed charges 100-200 units = 20rs , 200-500=30rs and above 500units = 50 has to
be //add     with amount
case 1:
//calculate
if(uc>=0 && uc<=100)
{
amt=uc*1;
}
else
{
if(uc>100 && uc<=200)
{
amt=((uc-100)*2.50)+100+20;
}
else
{
if(uc>200 && uc<=500)
{
amt=((uc-200)*4)+350+30;
}
else
{
amt=((uc-500)*6)+1550+50;
}
```

```
}
}
break;
//Commercial EB Bill Calculation
//Fixed charges 100-200 units = 20rs , 200-500=30rs and above 500units = 50 has to
//be add with amount
case 2:
if(uc>=0 && uc<=100)
{
amt=uc*2;
}
else
{
if(uc>100 && uc<=200)
{
amt=((uc-100)*4.50)+200+20;
}
else
{
if(uc>200 && uc<=500)
{
amt=((uc-200)*6)+650+30;
}
else
{
amt=((uc-500)*7)+2450+50;
}
```

```
}
}
break;
default:
System.out.println("invalid input");
break;
}
//print
System.out.println("----------------------");
System.out.println("Electricity Bill");
System.out.println("----------------------");
System.out.println("Consumer no \t" + cno);
System.out.println("----------------------");
System.out.println("consumer name\t" + cname);
System.out.println("----------------------");
System.out.println("previous month reading\t" + cpr);
System.out.println("----------------------");
System.out.println("current month reading\t"+ ccr);
System.out.println("----------------------");
System.out.println("Total bill amount \t"+ amt);
}
}
```

OUTPUT:

Enter Consumer no 101

Enter consumer name XX

Enter previous month reading 200

Enter current month reading 310

Enter

1. Domestic
2. Commercial

1

Electricity Bill

Consumer no 101

---

consumer name XX

---

previous month reading 200

---

current month reading 310

Total bill amount 145.0

Result

Thus the java application program to generate electricity generator has been executed successfully.

2. Develop a javaL 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 develop a Java application to implement currency converter , distance converter and time converter using packages.

ALGORITHM:

1. Start the program
2. Create three packages for currency converter , distance converter and time converter.
3. Create corresponding code for conversion
4. Print the converted value.
5. Stop the program

PROGRAM:

```
package allclass;
public class cur
{
double d,i,e,i1,y,i2,i3,y1;
public double dtoi(double d)
{
i=(d*68.44);
return i;
}
public double etoi(double e)
{
i1=(e*80.10);
return i1;
}
public double ytoi(double y)
{
i2=(y*0.61);
return (double)i2;
}
public double itoy(double i3)
{
y1=(i3*1.61);
return y1;
}
}
```

```
package allclass;
public class dis
{
double km,m,mi1,km1,km2,mi2;

public double mtok(double m)
{
km=(m*0.001);
return km;
}
public double mi1tokm(double mi1)
{
km1=(mi1*1.6093);
return km1;
}
public double kmtomi(double km2)
{
mi2=(km2*0.62);
return mi2;
}
}
package allclass;
public class tim
{
double h,mi,s,mi1,mi2,s1;
public double htom(double h)
{
```

```
mi=(h*60);
return mi;
}
public double stom(double s)
{
mi1=(s*0.016);
return mi1;
}
public double mitos(double mi2)
{
s1=(mi2*60);
return s1;
}
}
import java.io.*;
import java.util.Scanner;
import allclass.*;
public class converterpgm
{
public static void main(String args[])
{
//char ch;
int choice,c3,c1,c2,ch;
double
d1,inr1,e1,inr2,y2,inr3,i4,inr4,h1,mi3,mi4,s1,mi5,s2,m1,km3,mi8,km4,km5,mi7;
 cur c=new cur();
tim t=new tim();
```

```
dis d=new dis();
do
{
Scanner input=new Scanner(System.in);
System.out.println(" 1. Currency converter \n 2. Time converter \n 3. Distance
conveter \n");
choice=input.nextInt();
switch(choice)
{
case 1:
System.out.println(" 1. dollar to inr \n 2. euro to inr \n 3. yen to inr \n 4. inr to yen
\n");
c3=input.nextInt();
switch(c3)
{
case 1:
System.out.println("Enter the dollar value \n");
d1=input.nextDouble();
inr1=c.dtoi(d1);
System.out.println("INR Value" + inr1);
break;
case 2:
System.out.println("Enter the euro value \n");
e1=input.nextDouble();
inr2=c.etoi(e1);
System.out.println("INR Value" + inr2);
break;
```

```
case 3:
System.out.println("Enter the yen value \n");
y2=input.nextDouble();
inr3=c.ytoi(y2);
System.out.println("INR Value" + inr3);
break;
case 4:
System.out.println("Enter the inr value \n");
i4=input.nextDouble();
inr4=c.itoy(i4);
System.out.println("YEN Value" + inr4);
break;
default:
System.out.println("Invalid input");
break;
}
break;
case 2:
System.out.println(" 1. hour to minuntes \n 2. seconds to minutes \n 3. minutes to
seconds \n");
c1=input.nextInt();
switch(c1)
{
case 1:
System.out.println("Enter the hour value \n");
h1=input.nextDouble();
mi3=t.htom(h1);
```

```
System.out.println("Minutes value:"+mi3);
break;
case 2:
System.out.println("Enter the seconds value \n");
s1=input.nextDouble();
mi4=t.stom(s1);
System.out.println("Minutes value:"+mi4);
break;
case 3:
System.out.println("Enter the minutes value \n");
mi5=input.nextDouble();
s2=t.mitos(mi5);
System.out.println("Seconds value:"+s2);
break;
default:
System.out.println("Invalid input");
break;
}
break;
case 3:
System.out.println(" 1. meter to km \n 2. miles to km \n 3. km to miles \n");
c2=input.nextInt();
switch(c2)
{
case 1:
System.out.println("Enter the meter value \n");
m1=input.nextDouble();
```

```
km3=d.mtok(m1);
System.out.println("KM Value"+ km3);
break;
case 2:
System.out.println("Enter the miles value \n");
mi8=input.nextDouble();
km4=d.mtok(mi8);
System.out.println("KM Value"+ km4);
break;
case 3:
System.out.println("Enter the KM value \n");
km5=input.nextDouble();
mi7=d.mtok(km5);
System.out.println("Miles Value"+ mi7);
break;
default:
System.out.println("Invalid input");
break;
}
break;
}
System.out.println("Do you want to continue (yes(1)/no(0))");
ch=input.nextInt();
}while(ch==1);
}
}
```

OUTPUT:

```
C:\jdk1.6.0\bin>javac converterpgm.java
C:\jdk1.6.0\bin>java converterpgm
 1. Currency converter
 2. Time converter
 3. Distance conveter
2
 1. hour to minuntes
 2. seconds to minutes
 3. minutes to seconds
1
Enter the hour value
1
Minutes value:60.0
Do you want to continue (yes(1)/no(0))
1
 1. Currency converter
 2. Time converter
 3. Distance conveter
1
 1. dollar to inr
 2. euro to inr
 3. yen to inr
 4. inr to yen
1
```

Enter the dollar value

50

INR Value3422.0

Do you want to continue (yes(1)/no(0))

0

C:\jdk1.6.0\bin>

Result

Thus the java application to implement currency converter, distance convertor, time convertor using package has been done successfully.

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

AIM:

To 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% ofBP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary

ALGORITHM:

1. Start the program
2. Create Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members.
3. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class.
4. 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.
5. Generate pay slips for the employees with their gross and net salary.
6. Stop the program

PROGRAM:

```
import java.io.*;
import java.util.Scanner;
class Employee
{
int Emp_id,Mobile_no;
String Emp_name,address,Mail_id;
Scanner get=new Scanner(System.in);
Employee()
{
System.out.println("Enter Name of the Employee:");
Emp_name=get.nextLine();
System.out.println("Enter Mail_id of the Employee:");
Mail_id=get.nextLine();
System.out.println("Enter Address of the Employee:");
address=get.nextLine();
System.out.println("Enter Emp_id:");
Emp_id=get.nextInt();
System.out.println("Enter Mobile_no:");
Mobile_no=get.nextInt();
}
void display()
{
System.out.println("Employee Id:"+Emp_id);
System.out.println("Employee name:"+Emp_name);
System.out.println("Phone Number:"+Mobile_no);
System.out.println("Mail_id:"+Mail_id);
```

```
System.out.println("Address:"+address);
}
}

class Programmer extends Employee
{
float bp,salary,hra,da,swf,pf;
Programmer()
{
System.out.println("Enter Basic Pay:");
bp=get.nextFloat();
hra=bp*10/100;
da=bp*97/100;
pf=bp*12/100;
swf=bp*0.1/100;
salary=bp+da+hra-pf-swf;
}
void display()
{
System.out.println("Programmer Details"+"\n");
super.display();
System.out.println("Basic Payy:"+bp);
System.out.println("HRA:"+hra);
System.out.println("DA:"+da);
System.out.println("Staff welfare amounts:"+smf);
System.out.println("PF:"+pf);
System.out.println("Salary:"+salary);
```

```
}
}
class AssistProf extands Employee
{
float bp,salary,hra,da,swf,pf;
AssistProf()
{
System.out.println("Enter Basic Pay:");
bp=get.nextFloat();
hra=bp*10/100;
da=bp*97/100;
pf=bp*12/100;
swf=bp*0.1/100;
salary=bp+da+hra-pf-swf;
}
void display()
{
System.out.println("Assistant Professor Details"+"\n");
super.display();
System.out.println("Basic Payy:"+bp);
System.out.println("HRA:"+hra);
System.out.println("DA:"+da);
System.out.println("Staff welfare amounts:"+smf);
System.out.println("PF:"+pf);
System.out.println("Salary:"+salary);
}
}
```

```
class AssoProf extands Employee
{
float bp,salary,hra,da,swf,pf;
AssoProf()
{
System.out.println("Enter Basic Pay:");
bp=get.nextFloat();
hra=bp*10/100;
da=bp*97/100;
pf=bp*12/100;
swf=bp*0.1/100;
salary=bp+da+hra-pf-swf;
}
void display()
{
System.out.println("Assistant Professor Details"+"\n");
super.display();
System.out.println("Basic Payy:"+bp);
System.out.println("HRA:"+hra);
System.out.println("DA:"+da);
System.out.println("Staff welfare amounts:"+smf);
System.out.println("PF:"+pf);
System.out.println("Salary:"+salary);
}
}
class Professor extands Employee
{
```

```
float bp,salary,hra,da,swf,pf;
Professor()
{
System.out.println("Enter Basic Pay:");
bp=get.nextFloat();
hra=bp*10/100;
da=bp*97/100;
pf=bp*12/100;
swf=bp*0.1/100;
salary=bp+da+hra-pf-swf;
}
void display()
{
System.out.println("Professor Details"+"\n");
super.display();
System.out.println("Basic Payy:"+bp);
System.out.println("HRA:"+hra);
System.out.println("DA:"+da);
System.out.println("Staff welfare amounts:"+smf);
System.out.println("PF:"+pf);
System.out.println("Salary:"+salary);
}
}
class Employees
{
public static void main(String args[])
{
```

```
System.out.println("Enter Employee Details"+"\n");
Employee e=new Employee();
e.display;
Programmer obj1=new Programmer();
obj1.display();
AssistProf obj2=new AssistProf();
obj2.display();
AssoProf obj3=new AssoProf();
obj3.display();
Professor obj4=new Professor();
obj4.display();
}
}
class Employee
{
String Emp_name, Address, Mail_id;
long Emp_id, Mobile_no;
}
class Programmer extends Employee
{
int BP;
double grosssalary,deductions,netsalary;
void displaydetail(String n,long id,String ad,String mid, long m)
{
Emp_name=n;
Address=ad;
Mail_id=mid;
```

```
Emp_id=id;
Mobile_no=m;
System.out.println(" Employee name \t"+ Emp_name);
System.out.println(" Employee Id \t"+ Emp_id);
System.out.println(" Employee Address \t"+ Address);
System.out.println(" Employee Mail id \t"+ Mail_id);
System.out.println(" Employee Mobile no \t"+ Mobile_no);
}
void calculate(int BP1)
{
BP=BP1;
grosssalary=BP+(0.97*BP)+(1.07*BP);
deductions=(0.12*BP)+(0.001*BP);
netsalary=grosssalary-deductions;
System.out.println("Gross Salary \t"+grosssalary+"\n"+"Net Salary\t"+netsalary);
}
}
class AssistantProfessor extends Employee
{
int BP;
double grosssalary,deductions,netsalary;
void displaydetail(String n,long id,String ad,String mid,long m)
{
Emp_name=n;
Address=ad;
Mail_id=mid;
Emp_id=id;
```

```
Mobile_no=m;
System.out.println(" Employee name \t"+ Emp_name);
System.out.println(" Employee Id \t"+ Emp_id);
System.out.println(" Employee Address \t"+ Address);
System.out.println(" Employee Mail id \t"+ Mail_id);
System.out.println(" Employee Mobile no \t"+ Mobile_no);
}
void calculate(int BP1)
{
BP=BP1;
grosssalary=BP+(0.97*BP)+(1.07*BP);
deductions=(0.12*BP)+(0.001*BP);
netsalary=grosssalary-deductions;
System.out.println("Gross Salary \t"+grosssalary+"\n"+"Net Salary\t"+netsalary);
}
}
class AssociateProfessor extends Employee
{
int BP;
double grosssalary,deductions,netsalary;
void displaydetail(String n,long id,String ad,String mid, long m)
{
Emp_name=n;
Address=ad;
Mail_id=mid;
Emp_id=id;
Mobile_no=m;
```

```
System.out.println(" Employee name \t"+ Emp_name);
System.out.println(" Employee Id \t"+ Emp_id);
System.out.println(" Employee Address \t"+ Address);
System.out.println(" Employee Mail id \t"+ Mail_id);
System.out.println(" Employee Mobile no \t"+ Mobile_no);
}
void calculate(int BP1)
{
BP=BP1;
grosssalary=BP+(0.97*BP)+(1.07*BP);
deductions=(0.12*BP)+(0.001*BP);
netsalary=grosssalary-deductions;
System.out.println("Gross Salary \t"+grosssalary+"\n"+"Net Salary\t"+netsalary);
}
}

class Professor extends Employee
{
int BP;
double grosssalary,deductions,netsalary;
void displaydetail(String n,long id,String ad,String mid, long m)
{
Emp_name=n;
Address=ad;
Mail_id=mid;
Emp_id=id;
Mobile_no=m;
```

```
System.out.println(" Employee name \t"+ Emp_name);
System.out.println(" Employee Id \t"+ Emp_id);
System.out.println(" Employee Address \t"+ Address);
System.out.println(" Employee Mail id \t"+ Mail_id);
System.out.println(" Employee Mobile no \t"+ Mobile_no);
}
void calculate(int BP1)
{
BP=BP1;
grosssalary=BP+(0.97*BP)+(1.07*BP);
deductions=(0.12*BP)+(0.001*BP);
netsalary=grosssalary-deductions;
System.out.println("Gross Salary \t"+grosssalary+"\n"+"Net Salary\t"+netsalary);
}
}

class employeepayslip
{
public static void main(String args[])
{
int bp,a,a1;
String name,address,mailid;
long mobno,id;
Programmer p=new Programmer();
AssistantProfessor aa=new AssistantProfessor();
AssociateProfessor aa1=new AssociateProfessor();
Professor p1=new Professor();
```

```
Scanner s=new Scanner(System.in);
do
{
System.out.println("Enter employee name");
name=s.next();
System.out.println("Enter employee id");
id=s.nextInt();
System.out.println("Enter employee address");
address=s.next();
System.out.println("Enter employee mailid");
mailid=s.next();
System.out.println("Enter employee mob no");
mobno=s.nextLong();
System.out.println(" \n Calculate payslip for \n 1. Programmer \n 2. Assistant
Professor \n 3. Associate Professor \n 4. Professor \n");
a=s.nextInt();
switch(a)
{
case 1:
System.out.println("Enter the basic salary");
bp=s.nextInt();
p.displaydetail(name,id,address,mailid,mobno);
p.calculate(bp);
break;
case 2:
System.out.println("Enter the basic salary");
bp=s.nextInt();
```

```
aa.displaydetail(name,id,address,mailid,mobno);
aa.calculate(bp);
break;
case 3:
System.out.println("Enter the basic salary");
bp=s.nextInt();
aa1.displaydetail(name,id,address,mailid,mobno);
aa1.calculate(bp);
break;
case 4:
System.out.println("Enter the basic salary");
bp=s.nextInt();
p1.displaydetail(name,id,address,mailid,mobno);
p1.calculate(bp);
break;
}
System.out.println(" \n Enter \n 1 to exit \n 0 to continue");
a1=s.nextInt();
}while(a1==0);
}
}
```

OUTPUT:

C:\Program Files\jdk1.6.0\bin>javac employeepayslip.java

C:\Program Files\jdk1.6.0\bin>java employeepayslip
Enter employee name
XX
Enter employee id
2525
Enter employee address
Gingee
Enter employee mailid
XX@gmail.com
Enter employee mob no
9894501242

Calculate payslip for
1. Programmer
2. Assistant Professor
3. Associate Professor
4. Proffessor

3
Enter the basic salary
40000
Employee name XX
Employee Id 2525

Employee Address Gingee

Employee Mail id XX@gmail.com

Employee Mobile no 9894501242

Gross Salary 121600.0

Net Salary 116760.0

Enter

1 to exit

0 to continue

1

C:\Program Files\jdk1.6.0\bin>

Result

Thus the java application with employee class with employee name, employee id and employee address has been done successfully.

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

AIM:

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

ALGORITHM:

1. Start the program
2. Define the interface.
3. Read the elements using array.
4. Initialize stackTop pointer as zero,
5. Define and use the method Push() to insert the elements into the stack with 'STACK OVERFLOW' condition.
6. Define and use the method pop() to remove an element from an array with 'STACK UNDERFLOW' condition
7. Display the output.

PROGRAM:

```
import java.io.*;
public interface Stack
{
public void push(int x) throws StackException;
public int pop() throws StackException;
}
public class ArrayStack implements Stack
{
public int[] item;
public int stackTop;
public ArrayStack(int size)
{
item=new int [size];  //Make array
stackTop=0;
}
public void push(int x) throws StackException
{
if(stackTop==item.length)
{
throw new StackException("Stack overview");
}
item[stackTop]=x;   //Store x in next slot
stackTop++;         //Advance one slot location
}
public int pop() throws StackException
{
```

```
int returnItem;
if(stackTop==0)
{
throw new StackException("Stack underflow");
}
returnItem=item[stackTop-1];//Get last stored item
stackTop--;                 //Back up one slot location
return returnItem;
}
public static void main(String[] args)
{
int x;
Stack s;
s=new ArrayStack(6);     //Will cause underflow
try
{
x=4;s.push(x);System.out.println("push("+x+");");
x=7;s.push(x);System.out.println("push("+x+");");
x=8;s.push(x);System.out.println("push("+x+");");
x=9;s.push(x);System.out.println("push("+x+");");
x=s.pop(x);System.out.println("pop()->"+x);
x=s.pop(x);System.out.println("pop()->"+x);
x=s.pop(x);System.out.println("pop()->"+x);
x=s.pop(x);System.out.println("pop()->"+x);
x=s.pop(x);System.out.println("pop()->"+x);
x=s.pop(x);System.out.println("pop()->"+x);
}
```

```
catch(StackException e)
{
System.out.println("Error detected:"+e.getMessage());
System.exit(1);
}
}
}
```

OUTPUT:

push(4);

push(7);

push(8);

push(9);

pop()->9

pop()->8

pop()->7

pop()->4

Error detected:Stack underflow

Result

To develop the java application to implement the interface using array has been done successfully.

5. Write a program to perform string operations using Array List. Write function 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 write a program to perform string operations using Array List. Write function for the following

a. Append - add at end
b. Insert - add at particular index
c. Search
d. List all string starts with given letter

ALGORITHM:

1. Start the program
2. Add the String as an object to List.
3. Get the choice from the user and do according to the choice
   a. Append-add at end
   b. Insert-add at particular index
   c. Search
   d. List all string starts with given letter.
4. Display the result
5. Stop the program.

PROGRAM:

```
import java.io.*;
import java.util.*;
public class ArrayListExample
{
ArrayList<String>list=new ArrayList<String>();
BufferedReaderbr=new BufferedReader(new InputStreamReader(System.in));
String item="";
public ArrayListExample() {
list.add("Rajkumar");
list.add("Ravi");
list.add("Kumar");
list.add("Karan");
list.add("Suresh");
list.add("Ram");
list.add("Sivan");
list.add("Lorenz");
}
void display main Menu()
throws Exception
{
intyesorno, choice =0;
do{
System.out.println("Main Menu\n1)Append String at end\n2)Insert String at
particular index \n3)Search String\n 4)List all string starts with given letter \n5)
Listall strings\n 6)Exit \n");
```

```
System.out.print("Enter your choice[1- 6]:");
choice=Integer.parseInt(br.readLine());
switch(choice) {
case1:
System.out.print("Enter the string to be inserted:");
item=br.readLine();
appendAtEnd(list,item);
break;
case2:
System.out.print("Enter the string to be inserted:");
item=br.readLine();
System.out.print("Enter the location wheretoinsert:");
int index=Integer.parseInt(br.readLine());
insertAtIndex(item,index);
break;
case3:
System.out.print("Enterthestringtobesearched:");
item=br.readLine();
search(list,item);
break;
case4:
System.out.print("Enter the first letter of string to be searched:");
item=br.readLine();
list AllStringBeginsWithLetter(list,item);
break;
case5:
list AllString(list);
```

```
break;
 case6:
 System.exit(0);
 default:
 System.out.println("Enter correct choice");
 }
 System.out.print("Want to continue Main Menu?[Press1forYes/0forNo]:");
 yesorno=Integer.parseInt(br.readLine());
 }while(yesorno==1);
 }
 void search(ArrayListlist,Stringitem)
 {
 if(list.contains(item))
 {
 System.out.println("Item presentin the list");
 } else{
 System.out.println("Item not present in the list");
 }
 }
 ArrayListinsertAtIndex(Stringitem,int  index)
 {
 list.add(index,item);
  returnlist;
  }
  voidlistAllString(ArrayListlist)
  {
```

```
   if(list.isEmpty()){
  System.out.println("Noiteminthelist");
  }
   else{
  System.out.println(list);
  }
  }
void list AllStringBeginsWithLetter(ArrayListlist,String ch)
{
ArrayList<String>temp=newArrayList();
for (int i =0; i< list.size(); i++)
{
item=list.get(i).toString();
if(item.startsWith(ch))
{
 temp.add(item);
 }
 }
 if(temp.isEmpty()){
 System.out.println("No item begins with this letter");
 } else{
 System.out.println(temp);
 }
  }
ArrayListappendAtEnd(ArrayListlist,Stringitem) throws Exception
{
```

```
list.add(item);
  returnlist;
  }
  public static void main(String args[]) throws Exception
  {
  ArrayListExample ale=new ArrayListExample();
  ale.display MainMenu();
  }
  }
```

OUTPUT:

MainMenu

a. Append String at end
b. Insert String at particular index
c. Search String
d. List all string starts with given letter
e. List all strings
f. Exit

Enter your choice [1-6]:5

[Raj kumar,Ravi,Kumar,Karan,Suresh,Ram,Sivan,Lorenz]

Want to continue Main Menu?[Press1 for Yes /0 for No]:

1

Main Menu

a. Append String at end
b. Insert String at particular index
c. Search String
d. List all string starts with given letter
e. List all strings
f. Exit

Enter your choice[1-6]:1

Enter the string to be inserted: Bill

Want to continue Main Menu?[Press1for Yes /0 for No]:

1

MainMenu

a. Append String at end

b. Insert String at particular index
c. Search String
d. List all string starts with given letter
e. List all strings
f. Exit

Enter your choice[1-6]:5

[Raj kumar,Ravi, Kumar,Karan,Suresh,Ram,Sivan,

Lorenz,Bill]

Want to continue Main Menu?[Press1forYes/0forNo]: 1

Main Menu

a. Append String at end
b. Insert String at particular index
c. Search String
d. List all string starts with given letter
e. List all strings
f. Exit

Enter your choice [1-6]:2

Enter the string to be inserted: Sudhakar

Enterthelocationwheretoinsert:2

Want to continue Main Menu?[Press1 for Yes/ 0 for No]: 1

Main Menu

a. Append String at end
b. Insert String at particular index
c. Search String
d. List all string starts with given letter
e. List all strings

f. Exit

Enter your choice [1-6]:5

[Raj kumar,Ravi, Sudhakar,Kumar,Karan,Suresh, Ram,Siva,

Lorenz,Bill]

Want to continue Main Menu? [Press1forYes/0forNo]: 1

Main Menu

a. Append String at end

b. Insert String at particular index

c. Search String

d. List all string starts with given letter

e. List all strings

f. Exit

Enter your choice [1-6]:3

Enter the string to be searched: Lorenz Item

Present in the list

Want to continue Main Menu?[Press1forYes/0forNo]: 1

Main Menu

a. Append String at end

b. Insert String at particular index

c. Search String

d. List all string starts with given letter

e. List all strings

f. Exit

Enter your choice [1-6]:3

Enter the string to be searched:Gates Item not present in the list

Want to continue Main Menu?[Press1forYes/0forNo]: 1

Main Menu

a. Append String at end

b. Insert String at particular index

c. Search String

d. List all string starts with given letter

e. List all strings

f. Exit

Enter your choice [1-6]:4

Enter the first letter of string to be searched: L

[Lorenz]

Want to continue Main Menu?[Press1forYes/0forNo]:0

```
import java.io.*;
import java.util.ArrayList;
public class MyBasicArrayList
{
public static void main(String args[])
{
ArrayList<String>a1=new ArrayList<>();
//add elements to the ArrayList
a1.add("JAVA");
a1.add("C++");
a1.add("PERL");
a1.add("PHP");
System.out.println(a1);
//Insert elements at a specific index
a1.add(2,"PLAY");
System.out.println(a1);
//System element by index
System.out.println("Element at index 1:"+a1.get(1));
//List  string starts with given letter
System.out.println("Does list contains JAVA?"+a1.contains("JAVA"));
System.out.println("Is array empty?"+a1.isEmpty());
System.out.println("Index of PERL is"+a1.indexOf("PERL"));
System.out.println("Size of the arraylist is:"+a1.size());
}
}
```

OUTPUT:

JAVA, C++, PERL, PHP]

[JAVA, C++, PLAY, PERL, PHP]

Element at index 1:C++

Does list contains JAVA? true

Is array list empty? false

Index of PERL is 3

Size of the array list is: 5

Result

Thus the java application program for string operation using array list, append-add at end, insert-add at particular index,search,list all string start with given letter has been executed successfully.

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 contain only the method print Area() that prints the area of the given shape.

AIM:

To 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 contain only the method print Area() that prints the area of the given shape.

ALGORITHM:

1. Start the program
2. Define the abstract class shape.
3. Define the class Rectangle with PrintArea() method that extends(makes use of) Shape.
4. Define the class Triangle with PrintArea() method that extends(makes use of) Shape.
5. Define the class Circle with PrintArea() method that extends(makes use of) Shape.
6. Print the area of the Rectangle,Triangle and Circle .
7. Stop the Program.

PROGRAM:

```
import java.io.*;
abstract class shape
{
int a=3,b=4;
abstract public void print_area();
}
class rectangle extends shape
{
public int area_rect;
public void print_area()
{
area_rect=a*b;
System.out.println("The area of rectangle is:"+area_rect);
}
}
class triangle extands shape
{
int area_tri;
public void print_area()
{
area_tri=(0.5*a*b);
System.out.println("The area of triangle is:"+area_tri);
}
}
class circle extands shape
{       int area_circle;
```

```
public void print_area()
{
area_circle=(int)(3.14*a*b);
System.out.println("The area of circle is:"+area_circle);
}
}
public class Main
{
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:

The area of rectangle is:12

The area of triangle is:6

The area of circle is:28

Result

Thus the java application program to create an abstract class named shape that contains two integers has verified successfully.

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

AIM:

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

ALGORITHM:

1. Start the program
2. Define the exception for getting a number from the user.
3. If the number is positive print the number as such.
4. If the number is negative throw the exception to the user as 'Number must be positive'.
5. Stop the Program.

PROGRAM:

```
import java.io.*;
class MyException extends Exception
{
public MyException(String s)
{
super(s);
}
}
public class Main
{
public static void main(String args[])
{
int no;
BufferedReader b=new BufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("Enter a positive number");
no=Integer.parseInt(b.readLine());
if(no<0)
{
throw new MyException("exception occurred");
}
else
{
System.out.println("Number"+ no);
}
}
```

```
catch(MyException ex)
{
System.out.println("Caught");
System.out.println(ex.getMessage());
}
catch(Exception e)
{}
}
}
```

OUTPUT:

```
C:\jdk1.6.0\bin>javac Main.java
C:\jdk1.6.0\bin>java Main
Enter a positive number
-10
Caught
exception occurred
```


Result

Thus the java application program to implement user defined exception handling has been executed successfully.

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

AIM:

To 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.

ALGORITHM:

1. Start the program
2. Read the filename from the user.
3. Use getName() Method to display the filename.
4. Use getPath() Method to display the path of the file.
5. Use getParent() Method to display its parent's information.
6. Use exists() Method to display whether the file exist or not
7. Use isFile() and isDirectory() Methods to display whether the file is file or directory.
8. Use canRead() and canWrite) methods to display whether the file is readable or writable.
9. Use lastModified() Method to display the modified information.
10. Use length() method to display the size of the file.
11. Use isHiddden() Method to display whether the file is hidden or not.

PROGRAM:

```
import java.io.*;
import java.util.Scanner;
class FileDemo
{
public static void main(String args[])
{
Scanner s=new Scanner(System.in);
String filename;
System.out.println("Enter the file name");
filename=s.next();
File f1=new File(filename);
System.out.println("file name"+f1.getName());
System.out.println("path"+f1.getPath());
System.out.println("parent"+f1.getParent());
System.out.println(f1.exists());
System.out.println(f1.canRead());
System.out.println(f1.canWrite());
System.out.println(f1.isDirectory());
System.out.println(f1.isFile());
System.out.println(f1.lastModified());
System.out.println(f1.length());
System.out.println(f1.isHidden());
}
}
```

OUTPUT:

```
C:\jdk1.6.0\bin>javac FileDemo.java
C:\jdk1.6.0\bin>java FileDemo
Enter the file name
FileDemo.java
file name:FileDemo.java
path:FileDemo.java
parent:null
File exists :true
Readable:true
Writable:true
Directory:false
File or not:true
Last Modified:1536136980469
Length:791
Hidden:false
```


Result

Thus the java application program for file information has been executed successfully.

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

AIM:

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

ALGORITHM:

1. Start the program
2. Design the first thread that generates a random integer for every 1 second .
3. If the first thread value is even, design the second thread as the square of the number and then print it.
4. If the first thread value is odd, then third thread will print the value of cube of the number.
5. Stop the program.

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 JavaProgram5
{
public static void main(String[] args)
{
A a = new A();
a.start();
}
}
```

OUTPUT:

```
D:\JP>java JavaProgram5
Main Thread and Generated Number is 10
New Thread 10 is EVEN and Square of 10 is: 100
-------------------------------------
Main Thread and Generated Number is 14
New Thread 14 is EVEN and Square of 14 is: 196
-------------------------------------
Main Thread and Generated Number is 83
New Thread 83 is ODD and Cube of 83 is: 571787
-------------------------------------
Main Thread and Generated Number is 1
New Thread 1 is ODD and Cube of 1 is: 1
-------------------------------------
Main Thread and Generated Number is 20
New Thread 20 is EVEN and Square of 20 is: 400
-------------------------------------
```


Result

Thus the java application to implement multithread application has been executed successfully.

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

AIM:

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

ALGORITHM:

1. Start the program
2. Define the array with the elements
3. Sets the first value in the array as the current maximum
4. Find the maximum value by comparing each elements of the array
5. Display the maximum value 6. Stop the program

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:

Max of 3, 4 and 5 is 5

Max of 6.6,8.8 and 7.7 is 8.8

Max of pear, apple and orange is pear

```
import java.io.*;
public class Max
{
public static void main(String[] args)
{
Integer[] list=new Integer[10];
for (int i=0;i<list.length;i++ )
{
list[i]=i;
}
System.out.println("Max="+max(list));
}
public static<E extends Comparable<E>>E max(E[]list)
{
E max=list[0];
for(int i=1;i<list.length;i++)
{
E element=list[i];
if(element.compareTo(max)>0)
{
max=element;
}}
return max;
```

```
}
}
```

OUTPUT:

Max=9

Result

Thus the java application to find maximum value of the given elements has been executed successfully.

11. Design calculatr using even-driven programming paradigm of java with the folloing options

a.Decimal manipulations

b.scientific manipulations

AIM:

To Design calculatr using even-driven programming paradigm of java java with the folloing options

a.Decimal manipulations

b.scientific manipulations

ALGORITHM:

1. Start the program
2. Using the swing components design the buttons of the calculator
3. Use key events and key listener to listen the events of the calculator.
4. Do the necessary manipulations.
5. Stop the program.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class Calculator extends JFrame {
private final Font BIGGER_FONT = new Font("monspaced",Font.PLAIN, 20);
private JTextField textfield;
private boolean number = true;
private String equalOp = "=";
private CalculatorOp op = new CalculatorOp();
public Calculator() {
textfield = new JTextField("", 12);
textfield.setHorizontalAlignment(JTextField.RIGHT);
textfield.setFont(BIGGER_FONT);
ActionListener numberListener = new NumberListener();
String buttonOrder = "1234567890 ";
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new GridLayout(4, 4, 4, 4));
for (int i = 0; i < buttonOrder.length(); i++) {
String key = buttonOrder.substring(i, i+1);
if (key.equals(" ")) {
buttonPanel.add(new JLabel(""));
} else {
JButton button = new JButton(key);
button.addActionListener(numberListener);
button.setFont(BIGGER_FONT);
buttonPanel.add(button);
```

```
}
}
ActionListener operatorListener = new OperatorListener();
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(4, 4, 4, 4));
String[] opOrder = {"+", "-", "*", "/","=","C","sin","cos","log"};
for (int i = 0; i < opOrder.length; i++) {
JButton button = new JButton(opOrder[i]);
button.addActionListener(operatorListener);
button.setFont(BIGGER_FONT);
panel.add(button);
}
JPanel pan = new JPanel();
pan.setLayout(new BorderLayout(4, 4));
pan.add(textfield, BorderLayout.NORTH );
pan.add(buttonPanel , BorderLayout.CENTER);
pan.add(panel , BorderLayout.EAST);
this.setContentPane(pan);
this.pack();
this.setTitle("Calculator");
this.setResizable(false);
}
private void action() {
number = true;
textfield.setText("");
equalOp = "=";
op.setTotal("");
}
```

```
class OperatorListener implements ActionListener {
public void actionPerformed(ActionEvent e) {
String displayText = textfield.getText();
if (e.getActionCommand().equals("sin"))
{
textfield.setText("" + Math.sin(Double.valueOf(displayText).doubleValue()));

 }else
 if (e.getActionCommand().equals("cos"))
{
textfield.setText("" +
Math.cos(Double.valueOf(displayText).doubleValue()));

 }
 else
if (e.getActionCommand().equals("log"))
 {
textfield.setText("" +
Math.log(Double.valueOf(displayText).doubleValue()));

 }
 else if (e.getActionCommand().equals("C"))
{
 textfield.setText("");
}
else
{
if (number)
```

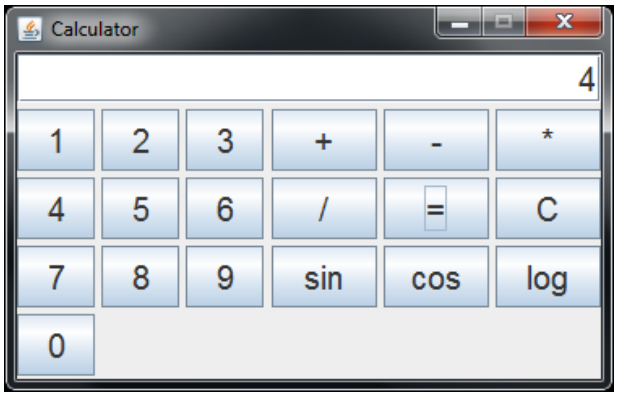
```
{
action();
textfield.setText("");
}
else
{
number = true;
if (equalOp.equals("="))
{
 op.setTotal(displayText);
 }else
 if (equalOp.equals("+"))
 {
 op.add(displayText);
 }
else if (equalOp.equals("-"))
 {
op.subtract(displayText);
 }
 else if (equalOp.equals("*"))
 {
 op.multiply(displayText);
}
else if (equalOp.equals("/")
 {
 op.divide(displayText);
   }
```

```
 textfield.setText("" + op.getTotalString());
 equalOp = e.getActionCommand();
}
}
}
}
class NumberListener implements ActionListener {
public void actionPerformed(ActionEvent event) {
String digit = event.getActionCommand();
if (number) {
textfield.setText(digit);
number = false;
 } else
{
  textfield.setText(textfield.getText() + digit);
  }
 }
 }
  public class CalculatorOp
{
  private int total;
  public CalculatorOp()
{
  total = 0;
  }
  public String getTotalString()
{
  return ""+total;
```

```
 }
  public void setTotal(String n)
{
  total = convertToNumber(n);
  }
public void add(String n) {
total += convertToNumber(n);
}
public void subtract(String n)
 {
total -= convertToNumber(n);
}
public void multiply(String n)
{
total *= convertToNumber(n);
}
public void divide(String n)
{
total /= convertToNumber(n);
}
private int convertToNumber(String n)
{
return Integer.parseInt(n);
 }
 }
}
class SwingCalculator
{
```

```
public static void main(String[] args)
{
JFrame frame = new Calculator();
frame.set Default CloseOperation(JFrame.EXIT_ON_CLOSE);
frame.set Visible(true);
 }
}
```

OUTPUT:



Result

Thus the java application to design calculator using even driven programming paradigm has been executed successfully