SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur-603203

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR: 2020-2021

ODD SEMESTER

LAB MANUAL

(REGULATION - 2019)

1904306- OBJECT ORIENTED PROGRAMMING LAB

THIRD SEMESTER

B.Tech - Information Technology

Prepared By

Ms. R. Thenmozhi, Assistant Professor (Sel.G)/IT

Ms. R. Saranya, Assistant Professor (O.G)/IT

TABLE OF CONTENTS

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. To afford the necessary background in the field of Information Technology to deal with engineering problems to excel as engineering professionals in industries.
2. To improve the qualities like creativity, leadership, teamwork and skill thus contributing towards the growth and development of society.
3. To develop ability among students towards innovation and entrepreneurship that caters to the needs of Industry and society.
4. To inculcate and attitude for life-long learning process through the use of information technology sources.
5. To prepare then to be innovative and ethical leaders, both in their chosen profession and in other activities.

PROGRAMME OUTCOMES (POs)

After going through the four years of study, Information Technology Graduates will exhibit ability to:

PO#	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 Program specific outcomes

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

1904306	OBJECT ORIENTED PROGRAMMING LAB	L T P C
		0 0 4 2

OBJECTIVES

- To apply the concepts of classes.
- To understand and implement packages and interfaces.
- To handle I/O and exception handling.
- To understand file processing operations.
- To develop applications using event handling

LIST OF EXPERIMENTS

1. Write a java program to illustrate the concept of class and object creation.
2. Write a java program to implement constructors.
3. Write a java program to implement abstract class and abstract method.
4. Write a java program to implement Inheritance.
5. Write a java program to implement I/O, Throwing and Catching exceptions.
6. Write a java program to implement Designing Packages.
7. Write a java program to implement Interfaces in Java.
8. Write a java program to manipulate file operations.
9. Write a java program to create multithreads in Java applications.
10. Write a java program to implement Graphics classes
11. Write a java program to implement Event driven programming.

TOTAL: 60 PERIODS

LIST OF EQUIPMENTS FOR A BATCH OF 30 STUDENTS:

Software:

C++, Java

Hardware:

Standalone desktops - 30 Nos.

OUTCOMES

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

- Build software development skills using java programming for real-world applications
- Develop and implement Java programs for simple applications that make use of classes, packages and interfaces.
- Develop and implement Java programs using Inheritance and Interfaces.
- Develop and implement Java programs using array list, exception handling and multithreading.
- Design applications using file processing, generic programming and event handling.

COURSE OUTCOMES:

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

1904306.1	Build software development skills using java programming for real-world applications
1904306.2	Develop and implement Java programs for simple applications that make use of classes, packages and interfaces.
1904306.3	Develop and implement Java programs using Inheritance and Interfaces.
1904306.4	Develop and implement Java programs using array list, exception handling and multithreading.
1904306.5	Design applications using file processing, generic programming and event handling.

CO-PO Matrix:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1904306.1	3	-	-	-	-	-	2	1	-	-	1	2
1904306.2	-	3	3	3	-	1	-	-	-	-	-	-
1904306.3	2	-	-	-	2	-	-	-	2	-	-	-
1904306.4	-	-	2	1	-	-	-	-	2	2	-	-
1904306.5	1	-	3	1	3	2	-	-	3	3	-	-
1904306	2	3	3	2	3	2	2	1	2	3	1	2

CO-PSO Matrix:

Course	PSO 1	PSO 2
1904306.1	3	2
1904306.2	3	2
1904306.3	3	-
1904306.4	3	2
1904306.5	3	2
1904306	3	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.	Conduction & Execution of Experiment	25
2.	Record	10
3.	Model Test	15
Total		**50**

INTRODUCTION / DESCRIPTION OF MAJOR SOFTWARE USED

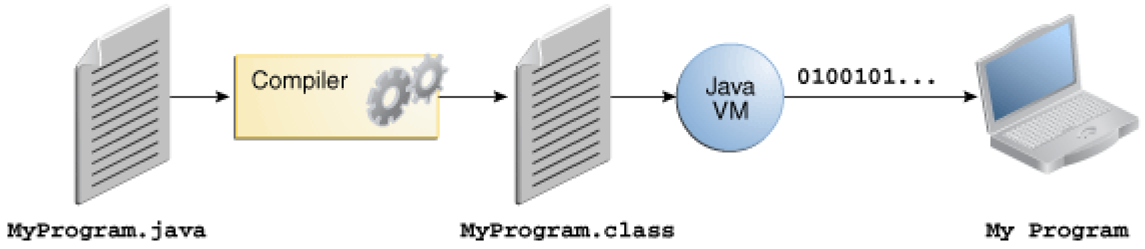
JAVA:

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters , game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere. Java is free to download.

The latest Java version contains important enhancements to improve performance, stability and security of the Java applications that run on your machine. Installing this free update will ensure that your Java applications continue to run safely and efficiently.

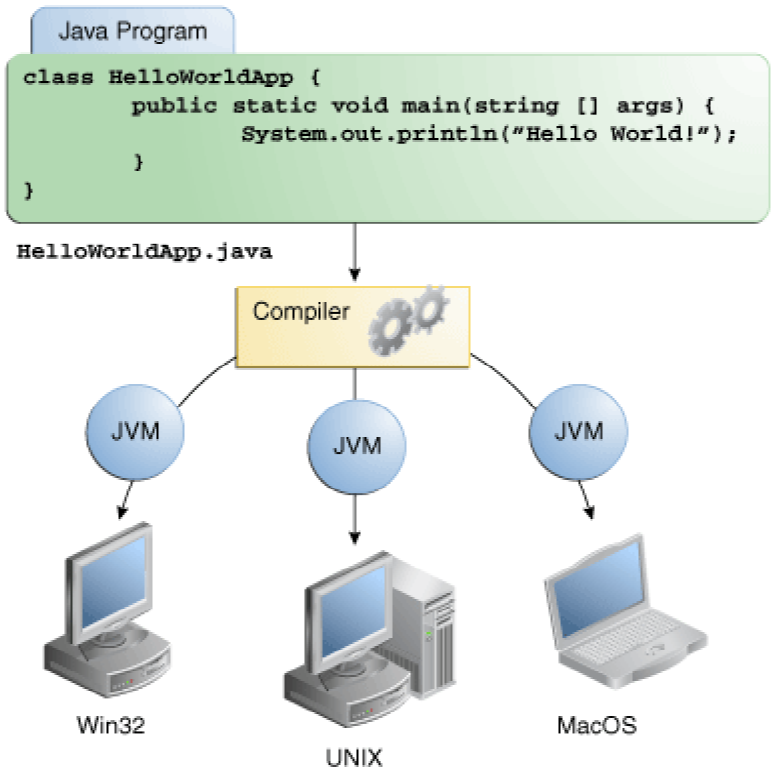
The Java Runtime Environment (JRE) is what you get when you download Java software. The JRE consists of the Java Virtual Machine (JVM), Java platform core classes, and supporting Java platform libraries. The JRE is the runtime portion of Java software, which is all you need to run it in your Web browser.
The Java Plug-in software is a component of the Java Runtime Environment (JRE). The JRE allows applets written in the Java programming language to run inside various browsers. The Java Plug-in software is not a standalone program and cannot be installed separately.
The Java Virtual Machine is only one aspect of Java software that is involved in web interaction. The Java Virtual Machine is built right into your Java software download, and helps run Java applications.

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains *bytecodes* — the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.



An overview of the software development process.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java SE HotSpot at a Glance, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code.



Through the Java VM, the same application is capable of running on multiple platforms.

The Java Platform

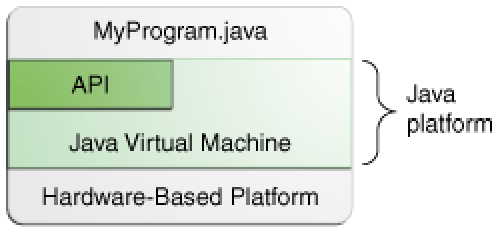
A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine
- The Java Application Programming Interface (API)

Java Virtual Machine; it's the base for the Java platform and is ported onto various hardware-based platforms.

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.



The API and Java Virtual Machine insulate the program from the underlying hardware.

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

Key advantages of Java Programming

Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

EX. NO.1

Implementation of Class and Object creation

Aim:

To implement class and object creation by developing a java programme for generating Electricity bill.

Generate Electricity bill by creating 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) and 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

PRELAB DISCUSSION:

Java is an Object - Oriented Language. As a language that has the Object - Oriented feature, Java supports the following fundamental concepts: Polymorphism, Inheritance, Encapsulation, Abstraction, Classes, Objects, Instance, Method, and Message Parsing.

- Object - Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors – wagging the tail , barking, eating. An object is an instance of a class.
- Class - A class can be defined as a template/blue print that describes the behavior /state that the object of its type support.
- Method - A method is a group of instructions that is given a name and can be called up at any point in a program simply by quoting that name. For instance, we met an instruction in the last lesson that draws a straight line on the screen. We could use this instruction three times to draw a simple triangle
- Messages - A single object alone is generally not very useful and usually appears as a component of a larger program or application that contains many other objects. Through the interaction of these objects, programmers achieve higher order functionality and more complex behavior.
- Abstraction : the virtue by which the data from a traditional process-oriented program can be transformed into its component objects.
- Encapsulation : the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.
- Inheritance : the process by which one object acquires the properties of another object.
- Polymorphism(Many Forms) : A feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation.

ALGORITHM:

1. Start the program
2. Create a class with the name ElectricityBill.

3. Get the details of the user and type of the connection.
4. Calculate the tariff based on the type of connection.
5. Compile and Execute the class "ElectricityBill", and generate the appropriate output.
6. Stop the program

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C: \set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac ElectricityBill.java
3.C:\java ElectricityBill
```

SOURCE CODE:

```
import java.util.*;
import java.io.*;
class ElectricityBill{
  public static void main(String args[])  {
   int choice;
   long units, consumer_no, current_reading, previous_reading;
   String consumer_name;
   Scanner sc=new Scanner(System.in);
   System.out.println("Enter Consumer Number:");
   consumer_no=sc.nextLong();
   System.out.println("Enter Consumer Name");
   consumer_name=sc.next();
   System.out.println("Enter Current Reading");
   current_reading=sc.nextLong();
   System.out.println("Enter Previous Reading");
   previous_reading=sc.nextLong();
   System.out.println("Type of EB connection");
   System.out.println("1. Domestic");
   System.out.println("2. Commercial");
   System.out.println("Select the EB connection");
   choice=sc.nextInt();
   double billpay=0;
   units=current_reading-previous_reading;
   switch(choice)
    {
        case 1 :
           if(units<100)
             billpay=units*1;
             else if(units<200)
               billpay=100*1+(units-100)*2.50;
               else if(units<500)
                 billpay=100*1+ 100*2.50+(units-200)*4;
               else if(units>500)
                 billpay=100*1+100*2.50+300*4+(units-500)*6;
          System.out.println("Consumer Number :" +consumer_no+"\nConsumer Name : "+consumer_name+
"\nPrevious Reading : " +previous_reading+"\nCurrent Reading : "+current_reading);
          System.out.println("\nTotal units consumed : "+units);
          System.out.println("Bill to pay : " + billpay);
           break;
        case 2 :
          if(units<100)
             billpay=units*2;
             else if(units<200)
               billpay=100*2+(units-100)*4.50;
               else if(units<500)
                 billpay=100*2+ 100*4.50+(units-200)*6;
               else if(units>500)
                 billpay=100*2+100*4.50+300*6+(units-500)*7;
          System.out.println("Consumer Number :" +consumer_no+"\nConsumer Name : "+consumer_name+
"\nPrevious Reading : " +previous_reading+"\nCurrent Reading : "+current_reading);
          System.out.println("\nTotal units consumed : "+units);
          System.out.println("Bill to pay : " + billpay);
           break;
        default :
           System.out.println("wrong reading");
    }
  }
}
```

INPUT AND OUTPUT:

```
Enter Consumer Number: 1098
Enter Consumer Name:  Mahesh
Enter Current Reading:  3750
Enter Previous Reading :  3350
Type of EB connection
1. Domestic
2. Commercial
Select the EB connection : 1
Consumer Number :1098
Consumer Name : Mahesh
Previous Reading : 3350
Current Reading : 3750
Total units consumed : 400
Bill to pay : 1150.0
```

VIVA QUESTIONS (PRELAB & POSTLAB):

1. Explain how to create instance of a class by giving an example
2. What is the purpose of the Runtime class?
3. What is the default value of an object reference declared as an instance variable?
4. Can a top-level class be private or protected?

Result:

Thus the class and Object creation are implemented using Electricity bill generation.

Ex.No. 2

Implementation of Constructors

AIM

To implement Constructors with parameters and without parameters by developing a java programme to calculate area.

PRELAB DISCUSSION:

A constructor is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes. The constructor name must match the class name, and it cannot have a return type. All classes have constructors by default: if you do not create a class constructor yourself, Java creates one. However, initial values for object attributes cannot be set.

ALGORITHM:

(i)
1. Create a class A.
2. Create a constructor without parameter.
3. Using that constructor calculate the area
4. Stop the program execution.

(ii)
1. Create a class A
2. Create a constructor with Parameter and follow the above procedure
3. Using that constructor calculate the area
4. Stop the program execution

PROCEDURE:

To execute a java program we require setting a class path:

(i)
```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac conswithP.java
3.C:\java conswithP
```
(ii)
```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac conswoP.java
3.C:\java conswoP
```

SOURCE CODE:

(i) A constructor with no parameters:

```
class A
{
int l,b;
A()
{
l=10;
b=20;
}
int area()
{
return l*b;
}
}
class constructordemo
{
public static void main(String args[])
{
A a1=new A();
int r=a1.area();
System.out.println("The area is: "+r);
}
}
```

OUTPUT:

```
The area is:200
```

(ii)A constructor with parameters

```
class A
{
int l,b;
A(int u,int v)
{
l=u;
b=v;
}
int area()
{
return l*b;
}
}
class constructordemo
{
public static void main(String args[])
{
A a1=new A(10,20);
int r=a1.area();
System.out.println("The area is: "+r);
}
}
```

OUTPUT:

```
The area is: 200
```

VIVA QUESTIONS (PRELAB & POSTLAB)

1. What is Constructor Chaining ?
2. What happens if you keep a return type for a constructor?
3. How a no – argument constructor is different from default Constructor**?**
4. When do we need Constructor Overloading?
5. Why return type is not allowed for constructor?

RESULT:

Thus the concept of constructor with argument and without argument is implemented.

Ex. No. 3

Implementation of Abstract classes and Abstract methods

AIM

To implement Abstract classes and Abstract methods by creating an abstract class named shape that contains two integers and an empty method named printArea by providing three classes named Rectangle, Triangle and Circle subclass that each one of the classes extends the Class Shape. Each one of the classes contains only the method printArea() that prints the area of Shape.

PRELAB DISCUSSION:

A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods (method with body). Abstraction is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only important things to the user and hides the internal details for example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery. Abstraction lets you focus on what the object does instead of how it does it. There are two ways to achieve abstraction in java 1. Abstract class 2.Interface.

ALGORITHM:

1. Start the program
2. Create a class with the name Shape.
3. Create the Rectangle,Triangle,Circle extends Shape.
4. Create the objects to the individual classes.
5. Call the PrintArea() on individual object.
6. Stop the program.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C: \set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac Abstex.java
3.C:\java Abstex
```

SOURCE CODE:

```
abstract class shape {
public int x, y;
public abstract void printArea();
}
class Rectangle extends shape {
public void printArea() {
System.out.println("Area of Rectangle is " + x * y);
}
}
class Triangle extends shape {
public void printArea() {
System.out.println("Area of Triangle is " + (x * y) / 2);
}
}
class Circle extends shape {
```

```
public void printArea() {
System.out.println("Area of Circle is " + (22 * x * x) / 7);
}
}
public class Abstex {
public static void main(String[] args) {
Rectangle r = new Rectangle();
r.x = 10;
r.y = 20;
r.printArea();
System.out.println("-------------------------------------");
Triangle t = new Triangle();
t.x = 30;
t.y = 35;
t.printArea();
System.out.println("-------------------------------------");
Circle c = new Circle();
c.x = 2;
c.printArea();
System.out.println("-------------------------------------");
}
}
```

INPUT AND OUTPUT:

```
Area of Rectangle is 56.0
Area of Triangle is 28.0
Area of Circle is 154.0
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1.What is difference between abstract class and interface ?
2.Can a abstract class be defined without any abstract methods?
3.Can an interface extend another interface?
4.Can an abstract class extend another interface?
5.What is the difference between abstract class and concrete class?

RESULT

Thus an abstract class and methods were implemented with name shape that contains two integers and an empty method named printArea by providing three classes named Rectangle, Triangle and Circle subclass that each one of the classes extends the Class Shape and each one of the classes contains only the method printArea() that prints the area of Shape.

Ex. No. 4

Implementation of Inheritance

AIM

To implement Inheritance by developing a java programme 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.

PRELAB DISCUSSION:

Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object. It is an important part of OPPs(Object Oriented programming system). The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also. Inheritance represents the IS-A relationship, also known as *parent-child* relationship.

Terms used in Inheritance

- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in previous class.

ALGORITHM:

1. Start the program.
2. Create a class called Employee with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members and compute the payslip
3. Extends the Employee class for further classes called Programmer, Assistant Professor, Associate Professor and Professor
4. Create class called Pay slip and generate Pay slip according to the position of the Employee.
5. Stop the program.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C: \set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac Employee.java
3.C:\java Employee
```

SOURCE CODE:

```
// Payslip.java
```

```
import java.util.Scanner;
class Employee
{
   String Emp_name,Mail_id,Address,Emp_id, Mobile_no;
   double BP,GP,NP,DA,HRA,PF,CF;
   Scanner get = new Scanner(System.in);
   Employee()
   {
      System.out.println("Enter Name of the Employee:");
      Emp_name = get.nextLine();
      System.out.println("Enter Address of the Employee:");
      Address = get.nextLine();
      System.out.println("Enter ID of the Employee:");
      Emp_id = get.nextLine();
      System.out.println("Enter Mobile Number:");
      Mobile_no = get.nextLine();
         System.out.println("Enter E-Mail ID of the Employee :");
         Mail_id = get.nextLine();
   }
    void display()
   {
      System.out.println("Employee Name: "+Emp_name);
      System.out.println("Employee Address: "+Address);
      System.out.println("Employee ID: "+Emp_id);
      System.out.println("Employee Mobile Number: "+Mobile_no);
System.out.println("Employee E-Mail ID"+Mail_id);
         DA=BP*0.97;
         HRA=BP*0.10;
         PF=BP*0.12;
         CF=BP*0.01;
         GP=BP+DA+HRA+PF;
         NP=GP-PF-CF;

      System.out.println("Basic Pay :"+BP);
      System.out.println("Dearness Allowance : "+DA);
      System.out.println("House Rent Allowance :"+HRA);
      System.out.println("Provident Fund :"+PF);
      System.out.println("Club Fund :"+CF);

      System.out.println("Gross Pay :"+GP);
      System.out.println("Net Pay :"+NP);
    }
}
 class Programmer extends Employee
{
    Programmer()
    {
        System.out.println("Enter Basic pay of the Programmer:");
        BP = get.nextFloat();
    }
    void display()
    {
        System.out.println("=============================="+"\n"+"Programmar Pay
Slip"+"\n"+"=============================="+"\n");
        super.display();
    }
}
 class Assistant_Professor extends Employee
{
    Assistant_Professor()
    {
        System.out.println("Enter Basic pay of the Assistant Professor:");
        BP = get.nextFloat();
    }
    void display()
    {
        System.out.println("=============================="+"\n"+"Assistant Professor
                                        Pay Slip"+"\n"+"=============================="+"\n");
        super.display();
    }
}

class Associate_Professor extends Employee
{
    Associate_Professor()
    {
        System.out.println("Enter Basic pay of the Professor:");
        BP = get.nextFloat();
    }
    void display()
    {
        System.out.println("================================"+"\n"+"Associate Professor Pay
Slip"+"\n"+"================================"+"\n");
        super.display();
    }
}

class Professor extends Employee
{
    Professor()
    {
        System.out.println("Enter Basic pay of the Professor:");
        BP = get.nextFloat();
    }
    void display()
    {
        System.out.println("================================"+"\n"+"Professor Pay
Slip"+"\n"+"================================"+"\n");
        super.display();
    }
}

class Payslip
{
    public static void main(String args[])
    {
            String pos;
            Scanner get = new Scanner(System.in);
            System.out.println("\nEnter Employee Position :");
            pos=get.nextLine();


            if(pos.equals("Programmer")||pos.equals("programmer"))
            {
               Programmer p=new Programmer();
               p.display();
            }
            if(pos.equals("AP")||pos.equals("ap"))
            {
                        Assistant_Professor AP=new Assistant_Professor();
                        AP.display();
            }
        if(pos.equals("ASSOCIATE")||pos.equals("associate"))
        {
              Associate_Professor ASP=new Associate_Professor();
                 ASP.display();
        }
        if(pos.equals("PROFESSOR")||pos.equals("professor"))
        {
                 Professor PR=new Professor();
                 PR.display();
        }
    }
}
```

INPUT AND OUTPUT:

```
Enter Employee Position :professor
Enter Name of the Employee:raj
Enter Address of the Employee:127 RR street chennai
Enter ID of the Employee:345
Enter Mobile Number:9797342543
Enter E-Mail ID of the Employee :raj@gmail.com
Enter Basic pay of the Professor:3000
================================
Professor Pay Slip
================================

Employee Name: raj
Employee Address: 127 RR street chennai
Employee ID: 345
Employee Mobile Number: 9797342543
Employee E-Mail IDraj@gmail.com
Basic Pay :3000.0
Dearness Allowance : 2910.0
House Rent Allowance :300.0
Provident Fund :360.0
Club Fund :30.0
Gross Pay :6570.0
Net Pay :6180.0
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1. Can you instantiate an object for a class that has a subclass overriding all the constructors of its base class?
2. Can a base-type access properties in its sub-types?
3. What is method overloading and method overriding?

4. What is the difference between Polymorphism and Inheritance?
5. What is meant by reusability?

RESULT:

Thus the Inheritance concept was implemented by creating Employee class and the Programmer, Assistant Professor, Associate Professor and Professor are inherited to the employee class and pay slip was generated.

Ex. No. 5

Implementation of Exception Handling

AIM

To write a Java program to implement Exception Handling.

PRELAB DISCUSSION:

In java we have already defined, exception classes such as ArithmeticException, NullPointerException etc. These exceptions are already set to trigger on pre-defined conditions such as when you divide a number by zero it triggers ArithmeticException, In the last tutorial we learnt how to throw these exceptions explicitly based on your conditions using throw keyword. In java we can create our own exception class and throw that exception using throw keyword. These exceptions are known as user-defined or custom exceptions. In this tutorial we will see how to create your own custom exception and throw it on a particular condition.

ALGORITHM:

```
1. Start the program
2. Create a class with the name “MyException extends Exception‟ to handle exception
3. Create the class userdef and get the age details
4. If the age is not in the specified range throw the exception
5. Print the message and stop the program.
```

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac userdef.java
3.C:\java userdef
```

SOURCE CODE:

```
import java.lang.Exception;
import java.lang.*;
import java.lang.Exception;
import java.io.DataInputStream;
class MyException extends Exception
{
MyException(String message)
{
super(message);
}
}
class userdef
{
public static void main(String a[])
{
int age;
DataInputStream ds=new DataInputStream(System.in);
try
{
System.out.println("Enter the age (above 15 and below 25) :");
age=Integer.parseInt(ds.readLine());
if(age<15 || age> 25)
{
throw new MyException("Number not in range");
}
System.out.println(" the number is :" +age);
}
catch(MyException e)
{
System.out.println("Caught MyException");
System.out.println(e.getMessage());
}
catch(Exception e){ System.out.println(e); }
}
}
```

INPUT AND OUTPUT:

```
c:\jdk1.6.0_26\bin>java userdef
Enter the age (above 15 and below 25) : 6
Caught MyException
Number not in range
```

VIVA QUESTIONS (PRELAB & POSTLAB):

1. What is checked exception?
2. What is an event and what are the models available for event handling?
3. What is difference between ClassNotFoundException and NoClassDefFoundError?
4. What is use of throws keyword?

RESULT:

Thus the user defined exception handling was implemented.

Ex. No: 6

Implementation of Packages

AIM:

To implement packages by writing a java programme for 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) .

PRELAB DISCUSSION:

A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.Here, we will have the detailed learning of creating and using user-defined packages. **Package = directory**. Java classes can be grouped together in *packages*. A package name is the same as the directory (folder) name which contains the .java files. You declare packages when you define your Java program, and you name the packages you want to use from other libraries in an *import* statement.

Package declaration

The first statement, other than comments, in a Java source file, must be the package declaration. Following the optional *package* declaration, you can have *import* statements, which allow you to specify classes from other packages that can be referenced without qualifying them with their package.

Default package. Altho all Java classes are in a directory, it's possible to omit the package declaration. For small programs it's common to omit it, in which case Java creates what it calls a *default* package. Sun recommends that you do not use default packages.

Package declaration syntax

The statement order is as follows. Comments can go anywhere.

1. Package statment (optional).
2. Imports (optional).
3. Class or interface definitions.

Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.
2) Java package provides access protection.
3) Java package removes naming collision.

ALGORITHM:

1. Start the program
2. Create a class with the name Conversion
3. Create a Package with the name mypack1 and import the mypack1 in the conversion program
4. Create a class currencyconverter, distanceconverter, timeconverter in the package calledmypack1.
5. Call the currencyconverter, distanceconverter, timeconverter from the conversion class.
6. Get the details of the user for the corresponding conversion and print it.
7.Stop the Program.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C: \set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac converter.java
3.C:\java converter
```

SOURCE CODE:

Main Program

```
package converter;
import mypack1.*;
import java.util.Scanner;
class converter
{
  public static void main(String args[])
  {
        Scanner in = new Scanner(System.in);
        System.out.println("1 Currency Converter");
        System.out.println("2 Distance Converter");
        System.out.println("3 Time Converter");
        System.out.println("3 Exit");
        System.out.print("Enter your choice: ");
        int choice=in.nextInt();
        switch(choice)
        {
        case 1:
           currency obj1 = new currency();
           obj1.convertcurrency();
        break;
        case 2:
           distanceconverter obj2=new distanceconverter();
           obj2.convertDistance();
        break;
        case 3:
           timeconverter obj3=new timeconverter();
           obj3.convertime();
        break;
        case 4:
           System.exit(0);
        break;
        }
  }
}
```

Currency Conversion

```
package mypack1;
import java.util.Scanner;
public class currency
{
    public void convertcurrency()
    {
        char us_dollar_sym = 36;
        char pound_sym = 163;
        char yen_sym = 165;
        char euro_sym = 8364;
        char inr_sym = 8377;
        String us_dollar = "Dollars";
        String pound = "Pounds";
        String yen = "Yen";
        String euro = "Euros";
        String rupee = "Rupees";
        double rate = 0;

        // Interface
        System.out.println("Welcome to the A Cool Currency Converter \n");
        System.out.println("Use the following codes to input your currency choices: \n 1 - US dollars \n 2 –
                                        Euros \n 3 - British Pounds \n 4 - Japanese Yen \n 5 - Indian Rupee \n");
        System.out.println("Please choose the input currency:");
        Scanner in = new Scanner(System.in);
        int choice = in.nextInt();
        String inType = null;
        switch (choice) {
                    case 1:
                    inType = "US Dollars >> " + us_dollar_sym;
                    break;
                    case 2:
                    inType = "Euros >> " + euro_sym;
                    break;
                    case 3:
                    inType = "British Pounds >> " + pound_sym;
                    break;
                    case 4:
                    inType = "Japanese Yen >> " + yen_sym;
                    break;
           case 5:
                    inType = "Indian Rupees >> " + inr_sym;
                    break;
        default:
           System.out.println("Please restart the program & enter a number from the list.");
           return;
        }
        System.out.println("Please choose the output currency");
        int output = in.nextInt();
        System.out.printf("Now enter the input in " + inType);
        double input = in.nextDouble();
        if (choice == output)
           System.out.println("Same currency no need to convert");
        if (choice == 1 && output == 2) {
           double dollar_euro_rate = 0.78391;
           rate = input * dollar_euro_rate;
           System.out.printf("%s" + input + " at a conversion rate of "  + dollar_euro_rate + " Dollars to %s =
                                                                            %.2f\n", (char) us_dollar_sym, euro, rate);
        } else if (choice == 1 && output == 3) {
            double dollar_pound_rate = 0.621484;
            rate = input * dollar_pound_rate;
            System.out.printf("%s" + input + " at a conversion rate of "+ dollar_pound_rate + " Dollars to %s =
                                                                            %.2f\n",  (char) us_dollar_sym, pound, rate);
        } else if (choice == 1 && output == 4) {
            double dollar_yen_rate = 107.174;
            rate = input * dollar_yen_rate;
            System.out.printf("%s" + input + " at a conversion rate of " + dollar_yen_rate + " Dollars to %s =
                                                                            %.2f\n", (char) us_dollar_sym, yen, rate);
        }
        else if (choice == 1 && output == 5) {
            double dollar_rupee_rate = 67.061;
            rate = input * dollar_rupee_rate;
            System.out.printf("%s" + input + " at a conversion rate of "+ dollar_rupee_rate + " Dollars to %s =
                                                                          %.2f\n",     (char) us_dollar_sym, rupee, rate);
        }
        if (choice == 2 && output == 1) {
            double euro_dollar_rate = 1.27579;
            rate = input * euro_dollar_rate;
            System.out.printf("%s" + input + " at a conversion rate of " + euro_dollar_rate + " Euros to %s =
                                                                          %.2f\n",     (char) euro_sym, us_dollar, rate);
        } else if (choice == 2 && output == 3) {
            double euro_pound_rate = 0.792648;
            rate = input * euro_pound_rate;
            System.out.printf("%s" + input + " at a conversion rate of " + euro_pound_rate + " Euros to %s =
                                                                          %.2f\n",     (char) euro_sym, pound, rate);
        } else if (choice == 2 && output == 4) {
            double euro_yen_rate = 136.708;
            rate = input * euro_yen_rate;
            System.out.printf("%s" + input + " at a conversion rate of " + euro_yen_rate + " Euros to %s =
                                                                            %.2f\n", (char) euro_sym,  yen, rate);
        }
        else if (choice == 2 && output == 5) {
            double euro_rupee_rate = 79.26;
            rate = input * euro_rupee_rate;

            System.out.printf("%s" + input + " at a conversion rate of "  + euro_rupee_rate + " Euros to %s =
                                                                                %.2f\n", (char) euro_sym, rupee, rate);
        }
        if (choice == 3 && output == 1) {
            double pound_dollar_rate = 1.60972;
            System.out.printf("%s" + input + " at a conversion rate of "+ pound_dollar_rate + " Pounds to %s =
                                                                            %.2f\n", (char) pound_sym, us_dollar, rate);
        } else if (choice == 3 && output == 2) {
            double pound_euro_rate = 1.26161;
```

```
            System.out.printf("%s" + input + " at a conversion rate of "+ pound_euro_rate + " Pounds to %s =
                                                                              %.2f\n", (char) pound_sym, euro, rate);
        } else if (choice == 3 && output == 4) {
            double pound_yen_rate = 172.511;
            System.out.printf("%s" + input + " at a conversion rate of "  + pound_yen_rate + " Pounds to %s =
                                                                                %.2f\n", (char) pound_sym, yen, rate);
        }
        else if (choice == 3 && output == 5) {
            double pound_rupee_rate = 90.231;
            System.out.printf("%s" + input + " at a conversion rate of "+ pound_rupee_rate + " Pounds to %s =
                                                                               %.2f\n",  (char) pound_sym, rupee, rate);
        }
        if (choice == 4 && output == 1) {
            double yen_dollar_rate = 0.00932574;
            System.out.printf("%s" + input + " at a conversion rate of " + yen_dollar_rate + " Yen to %s =
                                                                               %.2f\n", (char) yen_sym, us_dollar, rate);
        } else if (choice == 4 && output == 2) {
            double yen_euro_rate = 0.00730615;
            System.out.printf("%s" + input + " at a conversion rate of " + yen_euro_rate + " Yen to %s = %.2f\n",
                                                                                           (char) yen_sym, euro, rate);
        } else if (choice == 4 && output == 3) {
            double yen_pound_rate = 0.00579135;
            System.out.printf("%s" + input + " at a conversion rate of "+ yen_pound_rate + " Yen to %s =
                                                                                %.2f\n", (char) yen_sym, pound, rate);
        }
       else if (choice == 4 && output == 5) {
            double yen_rupee_rate = 0.61135;
            System.out.printf("%s" + input + " at a conversion rate of " + yen_rupee_rate + " Yen to %s = %.2f\n",
                                                                                          (char) yen_sym,  rupee, rate);
        }
        if (choice == 5 && output == 1) {
            double inr_dollar_rate = 0.0155555555555;
            System.out.printf("%s" + input + " at a conversion rate of " + inr_dollar_rate + " Rupees to %s =
                                                                              %.2f\n", (char) inr_sym,  us_dollar, rate);
        } else if (choice == 5 && output == 2) {
            double inr_euro_rate = 0.0139386;

    System.out.printf("%s" + input + " at a conversion rate of "+ inr_euro_rate + " Rupees to %s =
                                                                                   %.2f\n", (char) inr_sym, euro, rate);
        } else if (choice == 5 && output == 3) {
            double inr_pound_rate = 0.0119942;
            System.out.printf("%s" + input + " at a conversion rate of "+ inr_pound_rate + " Rupees to %s =
                                                                                   %.2f\n", (char) inr_sym,  pound, rate);
        } else if (choice == 5 && output == 4) {
            double inr_yen_rate = 1.72887;
            rate = input * inr_yen_rate;
            System.out.printf("%s" + input + " at a conversion rate of "+ inr_yen_rate + " Rupees to %s = %.2f\n",
                                                                    (char) inr_sym, yen, rate);
        }
        System.out.println("Thanks for using the currency converter!");
    }}
```

Distance Conversion

```
package mypack1;
import java.util.*;
public class distanceconverter
{
public void convertDistance()
{
System.out.println("1 KM to Miles");
System.out.println("2 Miles to KM");
System.out.println("3 Meter to KM");
System.out.println("4 KM to Meter");
System.out.println("5 Exit");
 float meter, kilometer;
Scanner input=new Scanner(System.in);
System.out.print("Enter your choice: ");
int choice=input.nextInt();
switch(choice)
{
    case 1:
        int km[]=new int[1];
        double miles[]=new double[1];
        System.out.println("Enter distance in KM: ");
        for(int i=0;i<km.length;i++){
            km[i]=input.nextInt();
            miles[i] = km[i] * 0.621;
        }
        System.out.println("KMS \t Miles");
        for(int i=0;i<km.length;i++){
        System.out.println(km[i]+"\t"+miles[i]);
        }
        break;

    case 2:
        int mile[]=new int[1];
        double kms[]=new double[1];
        System.out.println("Enter distance in Miles: ");
        for(int i=0;i<mile.length;i++){
            mile[i]=input.nextInt();
                    kms[i] = mile[i] / 0.621;
        }
        System.out.println("Miles \t KMS");
        for(int i=0;i<mile.length;i++){
        System.out.println(mile[i]+"\t"+kms[i]);
        }
        break;
    case 3:
            System.out.println("Enter meter:");
            Scanner input1 = new Scanner(System.in);
            meter = input1.nextFloat();
            kilometer = meter/1000;
            System.out.println("Length in kilometer = "+kilometer);
        break;
    case 4:
         meter = 1000;
        System.out.println("Enter Kilometer: ");
        Scanner reader = new Scanner(System.in);
        kilometer = reader.nextFloat();
        double kmtometer=kilometer*meter;
        System.out.println( kilometer+ " kilometers is " +kmtometer+ " in meters");
        break;

    case 5:
        System.exit(0);
        break;
}
}
}
```

Time Conversion

```
package mypack1;
import java.util.Scanner;
public class timeconverter
{
    public void convertime()
    {
        Scanner in = new Scanner(System.in);
        System.out.println("1 Seconds to Hours, Minutes, Seconds");
        System.out.println("2 Hours, Minutes, Seconds to seconds");
        System.out.println("3 Exit");
        System.out.print("Enter your choice: ");
        int choice=in.nextInt();
        switch(choice)
        {
        case 1:
            System.out.print("Enter seconds : ");
            int seconds = in.nextInt();
            int p1 = seconds % 60;
            int p2 = seconds / 60;
            int p3 = p2 % 60;
            p2 = p2 / 60;
            System.out.print("HH:MM:SS - " +p2 + ":" + p3 + ":" + p1);
            System.out.print("\n");
        break;
        case 2:
            System.out.print(" Enter the Hours Minutes and Seconds(HH:MM:SS) - " );
            String h=in.nextLine();
            String[] h1=h.split(":");
            int hour=Integer.parseInt(h1[0]);
            int minute=Integer.parseInt(h1[1]);
            int second=Integer.parseInt(h1[2]);
            int temp;
            temp = second + (60 * minute) + (3600 * hour);
            System.out.println("secondsss"+temp);
        break;
        case 3:
            System.out.exit(0);
        break;
        }
    }
}
```

INPUT AND OUTPUT:

```
1 Currency Converter
2 Distance Converter
3 Time Converter
3 Exit
Enter your choice: 1
Welcome to the A Cool Currency Converter

Use the following codes to input your currency choices:
 1 - US dollars
 2 - Euros
 3 - British Pounds
 4 - Japanese Yen
 5 - Indian Rupee

Please choose the input currency:
1
Please choose the output currency
5
Now enter the input in US Dollars >> $1
$1.0 at a conversion rate of 67.06 Dollars to Rupees = 67.06
Thanks for using the currency converter!

1 Currency Converter
2 Distance Converter
3 Time Converter
```

```
3 Exit
Enter your choice: 2
1 KM to Miles
2 Miles to KM
3 Meter to KM
4 KM to Meter
5 Exit
Enter your choice: 1
Enter distance in KM:
12
KMS         Miles
12       7.452

1 Currency Converter
2 Distance Converter
3 Time Converter
4 Exit
Enter your choice: 3
1 Seconds to Hours, Minutes, Seconds
2 Hours, Minutes, Seconds to seconds
3 Exit
Enter your choice: 1
Enter seconds : 12345
HH:MM:SS - 3:25:45
```

VIVA QUESTIONS (PRELAB & POSTLAB):

1. How to send the class file to another directory or drive?
2. How to put two public classes in a package?
3. What is static import feature of Java5?
4. Can we import same package/class twice? Will the JVM load the package twice at runtime?
5. Which mechanism used for naming and visibility control of a class and its content?

RESULT

Thus the packages were implemented in currency, distance, and time conversion java Program.

Ex. No: 7

Implementation of Interfaces

AIM

To implement a Java interface for ADT Stack by implementing this interface using array and to provide necessary exception handling in both the implementations.

PRELAB DISCUSSION:

An interface in java is a blueprint of a class. It has static constants and abstract methods. The interface in java is a mechanism to achieve abstraction. There can be only abstract methods in the java interface not method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have methods and variables but the methods declared in interface contain only method signature, not body. Java Interface also represents IS-A relationship. It cannot be instantiated just like abstract class. Interface is declared by using interface keyword. It provides total abstraction; means all the methods in interface are declared with empty body and are public and all fields are public, static and final by default. A class that implement interface must implement all the methods declared in the interface

ALGORITHM:

1: Start the program

2: Create an interface which consists of three methods namely PUSH, POP and DISPLAY

3**:** Create a class which implements the above interface to implement the concept of stack through Array

4**:** Define all the methods of the interface to push any element, to pop the element and to display the elements present in the stack.

5**:** Create another class which implements the same interface to implement the concept of stack through linked list.

6**:** Repeat STEP 4 for the above said class also.

7**:** In the main class, get the choice from the user to choose whether array implementation or linked list implementation of the stack.

8**:** Call the methods appropriately according to the choices made by the user in the previous step.

9**:** Repeat step 6 and step 7 until the user stops his/her execution

10: Stop the program

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac StackADT.java
3.C:\java StackADT
```

SOURCE CODE:

```
import java.io.*;
interface Mystack
{
   public void pop();
   public void push();
   public void display();
```

```
}
class Stack_array implements Mystack
{
    final static int n=5;
    int stack[]=new int[n];
    int top=-1;
    public void push()
    {
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            if(top==(n-1))
            {
                System.out.println(" Stack Overflow");
                return;
            }
            else
            {
                System.out.println("Enter the element");
                int ele=Integer.parseInt(br.readLine());
                stack[++top]=ele;
            }
        }
        catch(IOException 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 Link
{
    public int data;
    public Link nextLink;
    public Link(int d)
    {
        data= d;
        nextLink=null;
    }
    public void printLink()
    {
        System.out.print(" --> "+data);
    }
}
class Stack_List  implements Mystack
{
    private Link first;
    public Stack_List()
    {
        first = null;
    }
    public boolean isEmpty()
    {
        return first == null;
    }

    public void push()
    {
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the element");
            int ele=Integer.parseInt(br.readLine());
            Link link = new Link(ele);
            link.nextLink = first;
            first = link;
        }
        catch(IOException e)
        {
            System.err.println(e);
        }
    }
    public Link delete()
    {
        Link temp = first;
        try
        {
            first = first.nextLink;
        }
        catch(NullPointerException e)
        {
            throw e;
        }
        return temp;
    }

    public void pop()
    {
        try
        {
            Link deletedLink = delete();
            System.out.println("Popped: "+deletedLink.data);
        }
        catch(NullPointerException e)
        {
            throw e;
        }
    }
    public void display()
    {
        if(first==null)
            System.out.println("Stack is empty");
        else
        {
            Link currentLink = first;
            System.out.print("Elements are: ");
            while(currentLink != null)
            {
                currentLink.printLink();
                currentLink = currentLink.nextLink;
            }
            System.out.println("");
        }
    }
}
class StackADT
{
    public static void main(String arg[])throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Implementation of Stack using Array");
        Stack_array stk=new Stack_array();
        int ch=0;
        do
        {
            System.out.println("1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List");
            System.out.println("Enter your choice:");
            ch=Integer.parseInt(br.readLine());
            switch(ch)
            {
            case 1:
                stk.push();
                break;
            case 2:
                stk.pop();
                break;
            case 3:
                stk.display();
                break;
            case 4:
                System.exit(0);
            }
        }
        while(ch<5);
        System.out.println("Implementation of Stack using Linked List");
        Stack_List stk1=new Stack_List();
        ch=0;

        do
        {
            System.out.println("1.Push 2.Pop 3.Display 4.Exit");
            System.out.println("Enter your choice:");
            ch=Integer.parseInt(br.readLine());
            switch(ch)
            {
            case 1:
                stk1.push();
                break;
            case 2:
                try
                {
                   stk1.pop();
                }
                catch(NullPointerException e)
                {
                   System.out.println("Stack underflown");
                }
                break;
             case 3:
                stk1.display();
                break;
             default:
                System.exit(0);

             }
          }
          while(ch<5);
      }
}
```

INPUT AND OUTPUT:

```
Implementation of Stack using Array
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
1
Enter the element
10
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
1
Enter the element
15
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
1
Enter the element
25
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
3
Elements are:   10 <--  15 <--  25 <--
1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List
Enter your choice:
5
Implementation of Stack using Linked List
```

```
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
15
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:
3
Elements are:  --> 20 --> 15 --> 10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
2
Popped: 20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
3
Elements are:  --> 15 --> 10
1.Push 2.Pop 3.Display 4.Exit
Enter your choice:
4
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1. What will happen if we define a concrete method in an interface?
2. What is Nested Interface in Java?
3. Can we create non static variables in an interface? Can we declare interface as final?
4. When we need to use extends and implements?
5. What will happen if we are not implementing all the methods of an interface in class which implements an interface?

RESULT

Thus the Java interface concept is implemented for ADT Stack implementation.

Ex. No: 8

Implementation of File Handling

AIM

To implement file handling by creating 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.

PRELAB DISCUSSION:

File Handling in Java programming Language: FileWriter and FileReader classes are very frequently used to write and read data from text files (they are Character Stream classes). For any Byte stream classes, if you want to read and write them it is not wise and recommended to use FileInputStream.Java FileWriter is very useful in creating a file writing characters. This class inherits from the OutputStream class.The constructors of the class FileWriter usually assume that the byte-buffer size and default character encoding is acceptable. To declare them by oneself we need to construct OutputStreamWriter on a FileOutputStream.Java FileWriter is meant for writing streams of characters.

ALGORITHM:

1. Create a class by name AboutFile.
2. Inside the main method prompt the user to enter file name.
3. Provide the file name to the File object constructor to create an object of File.
4. Use the methods provided by the File class to know information such as file exists, readable, writable, length and provide the appropriate output.

PROCEDURE:

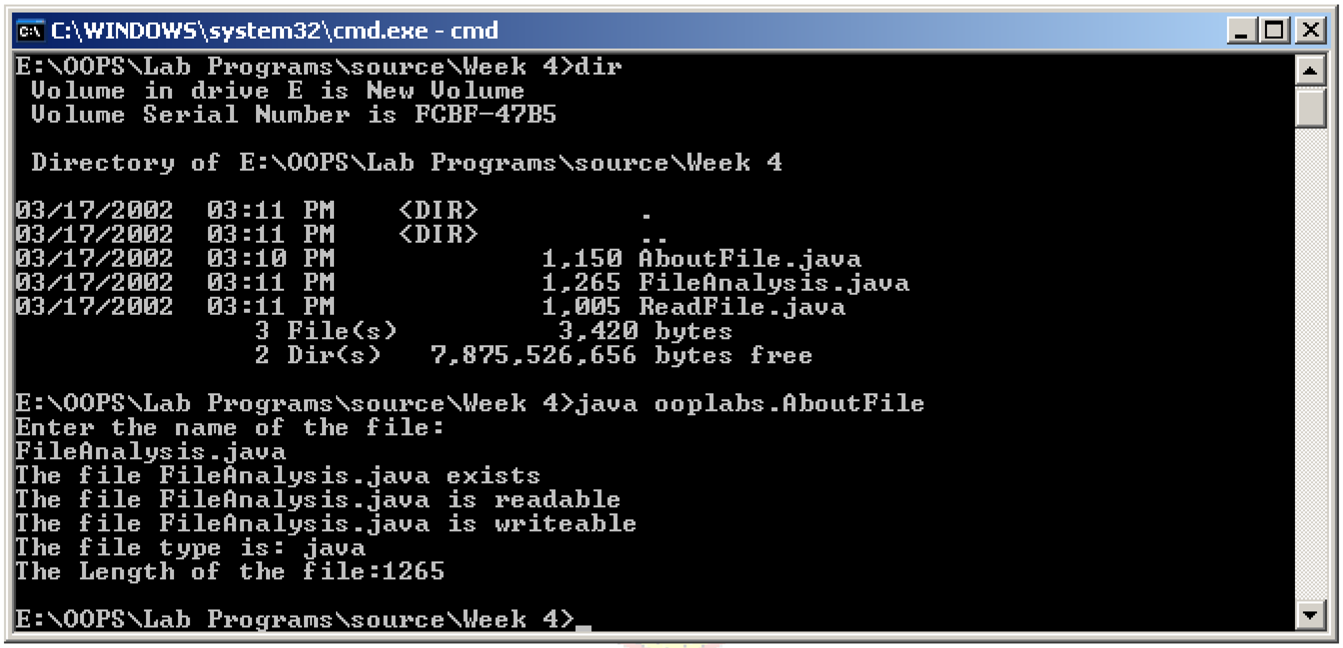
To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac AboutFile.java
3.C:\java AboutFile
```

SOURCE CODE:

```
package ooplabs;
import java.io.*;
import java.util.*;
class AboutFile{
public static void main(String[] args){
Scanner input = new Scanner(System.in);
System.out.println("Enter the name of the file:");
String file_name = input.nextLine();
File f = new File(file_name);
if(f.exists())
System.out.println("The file " +file_name+ " exists");
else
System.out.println("The file " +file_name+ " does not exist");
if(f.exists()){
if(f.canRead())
System.out.println("The file " +file_name+ " is readable");
else
System.out.println("The file " +file_name+ " is not readable");
 if(f.canWrite())
System.out.println("The file " +file_name+ " is writeable");
else
System.out.println("The file " +file_name+ " is not writeable");
System.out.println("The file type is: " +file_name.substring(file_name.indexOf('.')+1));
System.out.println("The Length of the file:" +f.length());
}
}
}
```

INPUT AND OUTPUT:



```
E:\OOPS\Lab Programs\source\Week 4>dir
 Volume in drive E is New Volume
 Volume Serial Number is FCBF-47B5

 Directory of E:\OOPS\Lab Programs\source\Week 4

03/17/2002  03:11 PM    <DIR>          .
03/17/2002  03:11 PM    <DIR>          ..
03/17/2002  03:10 PM             1,150 AboutFile.java
03/17/2002  03:11 PM             1,265 FileAnalysis.java
03/17/2002  03:11 PM             1,005 ReadFile.java
               3 File(s)          3,420 bytes
               2 Dir(s)   7,875,526,656 bytes free

E:\OOPS\Lab Programs\source\Week 4>java ooplabs.AboutFile
Enter the name of the file:
FileAnalysis.java
The file FileAnalysis.java exists
The file FileAnalysis.java is readable
The file FileAnalysis.java is writeable
The file type is: java
The Length of the file:1265

E:\OOPS\Lab Programs\source\Week 4>_
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1. How to compress a File in GZip format
2. How to Copy a File to another File in Java
3. How to get the last modified date of a file in java
4. How to make a File Read Only in Java
5. What is File Handling in Java

RESULT: Thus the file handling was implemented by creating a Java program to read 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.

Ex. No: 9

Implementation of MultiThreading

AIM

To implements a multithread application that has three threads, First thread generates random integer for every second and if the value is even,second thread computes the square of number and prints and If the value is odd,the third thread will print the value of cube of number

PRELAB DISCUSSION:

Threads enable a single program to run multiple parts of itself at the same time. For example, one part of a program can display an animation on the screen while another part builds the next animation to be displayed. Threads are a lightweight version of a process. Threads are similar to processes in that they can be executed independently and simultaneously, but are different in that they do not have all the overhead that a process does. Threads do not make copies of the entire parent process. Instead, only the code needed is run in parallel. This means that threads can be started quickly because they don't have all of the overhead of a complete process. They do, however, have complete access to all data of the parent process. Threads can read and/or write data that any other thread can access. This makes interthread communication simpler, but can lead to multiple threads modifying data in an unpredictable manner.

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.Threads can be created by using two mechanisms :1. Extending the Thread class 2. Implementing the Runnable Interface.

ALGORITHM:

1. Start the program
2. Create a class with the name "even implements Runnable"and "odd implements Runnable".
3. Create thread objects and Random class object.
4.Pass the objects of our class to thread class.
5. Call the start method.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac Mthread.java
3.C:\java Mthread
```

SOURCE CODE:

```
import java.util.*;
class even implements Runnable {
public int x;
public even(int x) {
this.x = x;
}
public
void run() {
```

```
System.out.println("Thread Name:Even Thread and " + x + "is even Number 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("Thread Name:ODD Thread and " + x + " is odd number and Cube of " + x + " is: " + x * x
* x);
}
}
class A extends Thread {
public String tname;
public Random r;
public Thread t1, t2;
public A(String s) {
tname = s;
}
public void run() { int num= 0;
r = new Random();
try {
for (int i = 0; i < 50; i++) {
num = r.nextInt(100);
System.out.println("Main Thread and Generated Number is " + num);
if (num % 2 == 0) {
t1 = new Thread(new even(num));
t1.start();
} else {
t2 = new Thread(new odd(
num));
t2.start();
}
Thread.sleep(1000);
System.out.println("--------------------------------------");
}
}
catch (Exception ex)
{
System.out.println(ex.getMessage());
}
}
}
public class Mthread
{
```

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

INPUT AND OUTPUT:

```
Main Thread and Generated Number is 92
Thread Name:Even Thread and 92is even Number and Square of 92 is: 8464
--------------------------------------
Main Thread and Generated Number is 58
Thread Name:Even Thread and 58is even Number and Square of 58 is: 3364
--------------------------------------
Main Thread and Generated Number is 46
Thread Name:Even Thread and 46is even Number and Square of 46 is: 2116
--------------------------------------
Main Thread and Generated Number is 63
Thread Name:ODD Thread and 63 is odd number and Cube of 63 is: 250047
--------------------------------------
Main Thread and Generated Number is 67
Thread Name:ODD Thread and 67 is odd number and Cube of 67 is: 300763
--------------------------------------
Main Thread and Generated Number is 57
Thread Name:ODD Thread and 57 is odd number and Cube of 57 is: 185193
--------------------------------------
Main Thread and Generated Number is 77
Thread Name:ODD Thread and 77 is odd number and Cube of 77 is: 456533
--------------------------------------
Main Thread and Generated Number is 58
Thread Name:Even Thread and 58is even Number and Square of 58 is: 3364
--------------------------------------
Main Thread and Generated Number is 38
Thread Name:Even Thread and 38is even Number and Square of 38 is: 1444
--------------------------------------
Main Thread and Generated Number is 30
Thread Name:Even Thread and 30is even Number and Square of 30 is: 900
--------------------------------------
Main Thread and Generated Number is 88
Thread Name:Even Thread and 88is even Number and Square of 88 is: 7744
--------------------------------------
Main Thread and Generated Number is 88
Thread Name:Even Thread and 88is even Number and Square of 88 is: 7744
--------------------------------------
Main Thread and Generated Number is 96
Thread Name:Even Thread and 96is even Number and Square of 96 is: 9216
--------------------------------------
```

```
Main Thread and Generated Number is 94
Thread Name:Even Thread and 94is even Number and Square of 94 is: 8836
--------------------------------------
Main Thread and Generated Number is 22
Thread Name:Even Thread and 22is even Number and Square of 22 is: 484
--------------------------------------
Main Thread and Generated Number is 51
Thread Name:ODD Thread and 51 is odd number and Cube of 51 is: 132651
--------------------------------------
Main Thread and Generated Number is 9
Thread Name:ODD Thread and 9 is odd number and Cube of 9 is: 729
--------------------------------------
Main Thread and Generated Number is 38
Thread Name:Even Thread and 38is even Number and Square of 38 is: 1444
--------------------------------------
Main Thread and Generated Number is 25
Thread Name:ODD Thread and 25 is odd number and Cube of 25 is: 15625
--------------------------------------
Main Thread and Generated Number is 54
Thread Name:Even Thread and 54is even Number and Square of 54 is: 2916
--------------------------------------
Main Thread and Generated Number is 20
Thread Name:Even Thread and 20is even Number and Square of 20 is: 400
--------------------------------------
Main Thread and Generated Number is 52
Thread Name:Even Thread and 52is even Number and Square of 52 is: 2704
--------------------------------------
Main Thread and Generated Number is 17
Thread Name:ODD Thread and 17 is odd number and Cube of 17 is: 4913
--------------------------------------
Main Thread and Generated Number is 99
Thread Name:ODD Thread and 99 is odd number and Cube of 99 is: 970299
--------------------------------------
Main Thread and Generated Number is 84
Thread Name:Even Thread and 84is even Number and Square of 84 is: 7056
--------------------------------------
Main Thread and Generated Number is 58
Thread Name:Even Thread and 58is even Number and Square of 58 is: 3364
--------------------------------------
Main Thread and Generated Number is 45
Thread Name:ODD Thread and 45 is odd number and Cube of 45 is: 91125
--------------------------------------
Main Thread and Generated Number is 69
Thread Name:ODD Thread and 69 is odd number and Cube of 69 is: 328509
--------------------------------------
Main Thread and Generated Number is 16
Thread Name:Even Thread and 16is even Number and Square of 16 is: 256
```

```
--------------------------------------
Main Thread and Generated Number is 61
Thread Name:ODD Thread and 61 is odd number and Cube of 61 is: 226981
--------------------------------------
Main Thread and Generated Number is 61
Thread Name:ODD Thread and 61 is odd number and Cube of 61 is: 226981
--------------------------------------
Main Thread and Generated Number is 65
Thread Name:ODD Thread and 65 is odd number and Cube of 65 is: 274625
--------------------------------------
Main Thread and Generated Number is 67
Thread Name:ODD Thread and 67 is odd number and Cube of 67 is: 300763
--------------------------------------
Main Thread and Generated Number is 43
Thread Name:ODD Thread and 43 is odd number and Cube of 43 is: 79507
--------------------------------------
Main Thread and Generated Number is 12
Thread Name:Even Thread and 12is even Number and Square of 12 is: 144
--------------------------------------
Main Thread and Generated Number is 29
Thread Name:ODD Thread and 29 is odd number and Cube of 29 is: 24389
--------------------------------------
Main Thread and Generated Number is 12
Thread Name:Even Thread and 12is even Number and Square of 12 is: 144
--------------------------------------
Main Thread and Generated Number is 21
Thread Name:ODD Thread and 21 is odd number and Cube of 21 is: 9261
--------------------------------------
Main Thread and Generated Number is 49
Thread Name:ODD Thread and 49 is odd number and Cube of 49 is: 117649
--------------------------------------
Main Thread and Generated Number is 18
Thread Name:Even Thread and 18is even Number and Square of 18 is: 324
--------------------------------------
Main Thread and Generated Number is 79
Thread Name:ODD Thread and 79 is odd number and Cube of 79 is: 493039
--------------------------------------
Main Thread and Generated Number is 61
Thread Name:ODD Thread and 61 is odd number and Cube of 61 is: 226981
--------------------------------------
Main Thread and Generated Number is 39
Thread Name:ODD Thread and 39 is odd number and Cube of 39 is: 59319
--------------------------------------
Main Thread and Generated Number is 29
Thread Name:ODD Thread and 29 is odd number and Cube of 29 is: 24389
--------------------------------------
Main Thread and Generated Number is 60
```

```
Thread Name:Even Thread and 60 is even Number and Square of 60 is: 3600
--------------------------------------
Main Thread and Generated Number is 48
Thread Name:Even Thread and 48is even Number and Square of 48 is: 2304
--------------------------------------
Main Thread and Generated Number is 80
Thread Name:Even Thread and 80is even Number and Square of 80 is: 6400
--------------------------------------
Main Thread and Generated Number is 14
Thread Name:Even Thread and 14is even Number and Square of 14 is: 196
--------------------------------------
Main Thread and Generated Number is 89
Thread Name:ODD Thread and 89 is odd number and Cube of 89 is: 704969
--------------------------------------
Main Thread and Generated Number is 27
Thread Name:ODD Thread and 27 is odd number and Cube of 27 is: 19683
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1.What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?
2. When you will synchronize a piece of your code?
3. What is daemon thread and which method is used to create the daemon thread?
4. What is deadlock?
5. What are the states associated in the thread?

RESULT

The multithread concept was implemented with the following three threads, First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints and if the value is odd, the third thread will print the value of cube of number.

Ex. No: 10

Implementation of Graphic Classes

AIM: To write a JAVA program to create different geometric shapes and fill colors.

PRELAB DISCUSSION:

The Graphics class is the abstract super class for all graphics contexts which allow an application to draw onto components that can be realized on various devices or onto off-screen images as well. A Graphics object encapsulates all state information required for the basic rendering operations that Java supports. State information includes the following properties. The Component object on which to draw. A translation origin for rendering and clipping coordinates. The current clip. The current color. The current font. The current logical pixel operation function.The current XOR alternation color.

ALGORITHM:

1. Start the program
2. Create a class with the name "graphicsdemo ".
3. Initialize the X and Y values.
4. Using the inbuilt methods Different shapes are drawn.
5. Stop the program.

PROCEDURE:

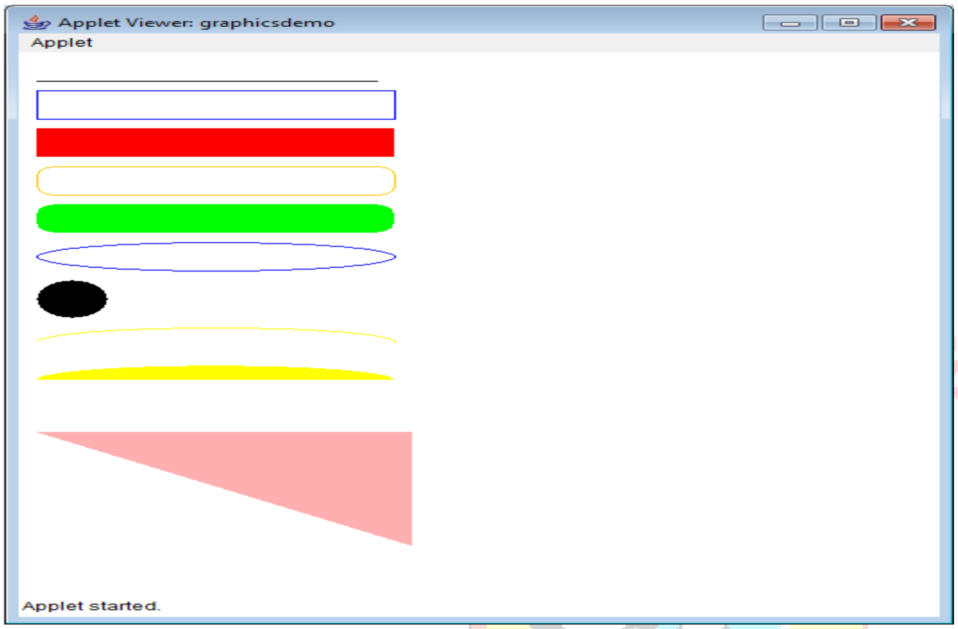
To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac shapes.java
3.C:\java shapes
```

SOURCE CODE:

```
import java.awt.*;
import java.applet.*;
//<applet code="graphicsdemo" width="400" height="400"></applet>
public class graphicsdemo extends Applet
{
public void paint(Graphics g)
{
int x[]={10,220,220};
int y[]={400,400,520};
int n=3;
g.drawLine(10,30,200,30);
g.setColor(Color.blue); g.drawRect(10,40,200,30);
g.setColor(Color.red); g.fillRect(10,80,200,30);
g.setColor(Color.orange); g.drawRoundRect(10,120,200,30,20,20);
g.setColor(Color.green); g.fillRoundRect(10,160,200,30,20,20);
g.setColor(Color.blue); g.drawOval(10,200,200,30);
g.setColor(Color.black); g.fillOval(10,240,40,40);
g.setColor(Color.yellow); g.drawArc(10,290,200,30,0,180);
g.setColor(Color.yellow); g.fillArc(10,330,200,30,0,180);
g.setColor(Color.pink); g.fillPolygon(x,y,n);
}
}
```

OUTPUT:



VIVA QUESTIONS(PRELAB & POSTLAB):

1. What is graphics API?
2. What is the purpose of graphics class?
3. What is AWT in Java?
4. Which graphics API is best?
5. What is fillRect in Java?

RESULT

Thus the Graphic Classes are implemented and different shapes are created.

Ex. No: 11

Implementation of Event-Driven Programming

AIM

To design a calculator by implementing event-driven programming paradigm of Java with the following Decimal and Scientific manipulations

PRELAB DISCUSSION:

The basic idea of event-driven programming is simply to create objects with methods that handle the appropriate events or circumstances, without explicit attention to how or when these methods will be called. These helper methods provide answers to questions of the form, "What should I do when xxx happens?" Because xxx is a "thing that happens", or an event, these methods are sometimes called event handlers. As the writer of event handler methods, you expect that the event handlers will somehow (automatically) be invoked whenever the appropriate thing needs dealing with, i.e., whenever the appropriate event arises. he result of this transformation is that your code focuses on the occasions when something of interest happens -- instead of the times when nothing much is going on -- and on how it should respond to these circumstances. An event is, after all, simply something (significant) that happens. This style of programming is called event-driven because the methods that you write -- the event handlers -- are the instructions for how to respond to events. The dispatcher -- whether central control loop or otherwise -- is a part of the background; the event handlers drive the code.

ALGORITHM:

1. Start the program
2. Create a class with the name "Scientific Calculator " extends JFrame implements ActionListener.
3. Declare the JButton ,JTextField ,JPanel variables.
4. Add listener to each button and to the layout.
5. Add Text Field to display the result.
6. Handle any Exceptions like divide by zero.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac ScientificCalculator.java
3.C:\java ScientificCalculator
```

SOURCE CODE:

```
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, pow3, exp, fac, plus, min, div, log, rec,
                                        mul, eq, addSub, dot, mr, mc, mp, mm, sqrt, sin, cos, tan;
Container cont;
JPanel textPanel, buttonpanel;

ScientificCalculator() {
        cont = getContentPane();
        cont.setLayout(new BorderLayout());
        JPanel textpanel = new JPanel();
        tfield = new JTextField(25);
        tfield.setHorizontalAlignment(SwingConstants.RIGHT);
        tfield.addKeyListener(new KeyAdapter() {
                public void keyTyped(KeyEvent keyevent) {
                        char c = keyevent.getKeyChar();
                        if (c >= '0' && c <= '9') {
                        } else {
                                keyevent.consume();
                        }
                }
        });
        textpanel.add(tfield);
        buttonpanel = new JPanel();
        buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
        boolean t = true;
        mr = new JButton("MR");
        buttonpanel.add(mr);
        mr.addActionListener(this);
        mc = new JButton("MC");
        buttonpanel.add(mc);
        mc.addActionListener(this);

        mp = new JButton("M+");
        buttonpanel.add(mp);
        mp.addActionListener(this);

        mm = new JButton("M-");
        buttonpanel.add(mm);
        mm.addActionListener(this);

        b1 = new JButton("1");
        buttonpanel.add(b1);
        b1.addActionListener(this);
        b2 = new JButton("2");
        buttonpanel.add(b2);
        b2.addActionListener(this);
        b3 = new JButton("3");
```

```
buttonpanel.add(b3);
b3.addActionListener(this);

b4 = new JButton("4");
buttonpanel.add(b4);
b4.addActionListener(this);
b5 = new JButton("5");
buttonpanel.add(b5);
b5.addActionListener(this);
b6 = new JButton("6");
buttonpanel.add(b6);
b6.addActionListener(this);

b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);

zero = new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);

plus = new JButton("+");
buttonpanel.add(plus);
plus.addActionListener(this);

min = new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);

mul = new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);

div = new JButton("/");
div.addActionListener(this);
buttonpanel.add(div);

addSub = new JButton("+/-");
buttonpanel.add(addSub);
addSub.addActionListener(this);
```

```
}

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

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

                rec = new JButton("1/x");
                buttonpanel.add(rec);
                rec.addActionListener(this);
                sqrt = new JButton("Sqrt");
                buttonpanel.add(sqrt);
                sqrt.addActionListener(this);
                log = new JButton("log");
                buttonpanel.add(log);
                log.addActionListener(this);

                sin = new JButton("SIN");
                buttonpanel.add(sin);
                sin.addActionListener(this);
                cos = new JButton("COS");
                buttonpanel.add(cos);
                cos.addActionListener(this);
                tan = new JButton("TAN");
                buttonpanel.add(tan);
                tan.addActionListener(this);
                pow2 = new JButton("x^2");
                buttonpanel.add(pow2);
                pow2.addActionListener(this);
                pow3 = new JButton("x^3");
                buttonpanel.add(pow3);
                pow3.addActionListener(this);
                exp = new JButton("Exp");
                exp.addActionListener(this);
                buttonpanel.add(exp);
                fac = new JButton("n!");
                fac.addActionListener(this);
                buttonpanel.add(fac);

                clr = new JButton("AC");
                buttonpanel.add(clr);
                clr.addActionListener(this);
                cont.add("Center", buttonpanel);
                cont.add("North", textpanel);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
```

```
public void actionPerformed(ActionEvent e) {
        String s = e.getActionCommand();
        if (s.equals("1")) {
                if (z == 0) {
                        tfield.setText(tfield.getText() + "1");
                } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "1");
                        z = 0;
                }
        }
        if (s.equals("2")) {
                if (z == 0) {
                        tfield.setText(tfield.getText() + "2");
                } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "2");
                        z = 0;
                }
        }
        if (s.equals("3")) {
                if (z == 0) {
                        tfield.setText(tfield.getText() + "3");
                } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "3");
                        z = 0;
                }
        }
        if (s.equals("4")) {
                if (z == 0) {
                        tfield.setText(tfield.getText() + "4");
                } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "4");
                        z = 0;
                }
        }
        if (s.equals("5")) {
                if (z == 0) {
                        tfield.setText(tfield.getText() + "5");
                } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "5");
                        z = 0;
                }
```

```
}
if (s.equals("6")) {
            if (z == 0) {
                        tfield.setText(tfield.getText() + "6");
            } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "6");
                        z = 0;
            }
}
if (s.equals("7")) {
            if (z == 0) {
                        tfield.setText(tfield.getText() + "7");
            } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "7");
                        z = 0;
            }
}
if (s.equals("8")) {
            if (z == 0) {
                        tfield.setText(tfield.getText() + "8");
            } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "8");
                        z = 0;
            }
}
if (s.equals("9")) {
            if (z == 0) {
                        tfield.setText(tfield.getText() + "9");
            } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "9");
                        z = 0;
            }
}
if (s.equals("0")) {
            if (z == 0) {
                        tfield.setText(tfield.getText() + "0");
            } else {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + "0");
                        z = 0;
            }
}
if (s.equals("AC")) {
```

```
                tfield.setText("");
                x = 0;
                y = 0;
                z = 0;
        }
        if (s.equals("log")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.log(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("1/x")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = 1 / Double.parseDouble(tfield.getText());
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("Exp")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.exp(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("x^2")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.pow(Double.parseDouble(tfield.getText()), 2);
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("x^3")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.pow(Double.parseDouble(tfield.getText()), 3);
                        tfield.setText("");
```

```
				tfield.setText(tfield.getText() + a);
			}
		}
		if (s.equals("+/-")) {
			if (x == 0) {
				tfield.setText("-" + tfield.getText());
				x = 1;
			} else {
				tfield.setText(tfield.getText());
			}
		}
		if (s.equals(".")) {
			if (y == 0) {
				tfield.setText(tfield.getText() + ".");
				y = 1;
			} else {
				tfield.setText(tfield.getText());
			}
		}
		if (s.equals("+")) {
			if (tfield.getText().equals("")) {
				tfield.setText("");
				temp = 0;
				ch = '+';
			} else {
				temp = Double.parseDouble(tfield.getText());
				tfield.setText("");
				ch = '+';
				y = 0;
				x = 0;
			}
			tfield.requestFocus();
		}
		if (s.equals("-")) {
			if (tfield.getText().equals("")) {
				tfield.setText("");
				temp = 0;
				ch = '-';
			} else {
				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("MC")) {
        m1 = 0;
        tfield.setText("");
}
if (s.equals("MR")) {
        tfield.setText("");
        tfield.setText(tfield.getText() + m1);
}
if (s.equals("M+")) {
        if (k == 1) {
                m1 = Double.parseDouble(tfield.getText());
                k++;
        } else {
                m1 += Double.parseDouble(tfield.getText());
                tfield.setText("" + m1);
        }
}
if (s.equals("M-")) {
        if (k == 1) {
```

```
                        m1 = Double.parseDouble(tfield.getText());
                        k++;
                } else {
                        m1 -= Double.parseDouble(tfield.getText());
                        tfield.setText("" + m1);
                }
        }
        if (s.equals("Sqrt")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.sqrt(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("SIN")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.sin(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("COS")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.cos(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("TAN")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.tan(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("=")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
```

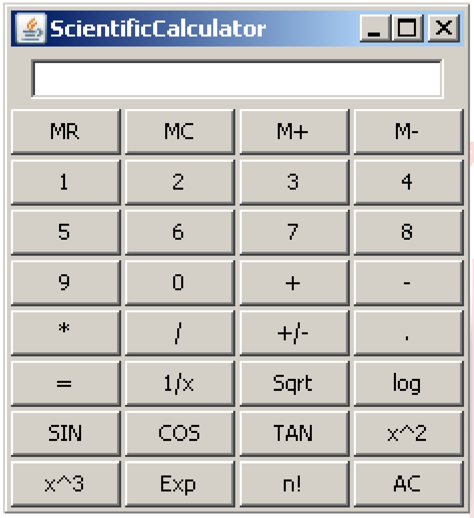
```
                                temp1 = Double.parseDouble(tfield.getText());
                                switch (ch) {
                                case '+':
                                        result = temp + temp1;
                                        break;
                    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

        double fact(double x) {
                int er = 0;
                if (x < 0) {
                        er = 20;
                        return 0;
                }
                double i, s = 1;
                for (i = 2; i <= x; i += 1.0)
                        s *= i;
                return s;
        }

        public static void main(String 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);
          }
}
```

INPUT AND OUTPUT:



VIVA QUESTIONS(PRELAB & POSTLAB):

1.How to convert the string data into long data?
2.Which containers use a Border layout as their default layout?
3.What is an event and what are the models available for event handling?
4.Which listeners are implemented for JButton ?
5. What is awt and swing?

RESULT

The calculator using event-driven programming paradigm of Java with the following Decimal and Scientific manipulations was implemented.

TOPIC BEYOND SYLLABUS

Ex. No:12 Implementation of String Operation using Array List

AIM

To perform string operations using Array List with the following functions for Append - add at end, Insert – add at particular index, Search and list all string starts with given letter.

PRELAB DISCUSSION:

Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.The Java platform provides the String class to create and manipulate strings.The most direct way to create a string is to write −`String greeting = "Hello world!";`Whenever it encounters a string literal in your code, the compiler creates a String object with its value in this case, "Hello world!'.As with any other object, you can create String objects by using the new keyword and a constructor. The String class has 11 constructors that allow you to provide the initial value of the string using different sources, such as an array of characters.

ALGORITHM:

5. Create an ArrayList named obj.
6. Add elements to the existing ArrayList at a specific position.
7. Insert the String elements at last one by one and display the Arraylist obj elements.
8. To search a specific element,get the input of the element to search.
9. Get the element starting letter from the user and find the position then store in the new ArrayList
10. Display the new Arraylist.
11. Stop the program execution.

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac stringoperation.java
3.C:\java stringoperation
```

SOURCE CODE:

```
import java.util.*;
import java.util.ArrayList;
public class stringoperation
{
  public static void main(String args[])
  {
     int choice, i,size;
     Scanner sc=new Scanner(System.in);
     ArrayList<String> obj = new ArrayList<String>();
     System.out.println("Enter the total number of strings to be entered into the arraylist");
     size=sc.nextInt();
     for (int j=0;j<size;j++)
```

```
        {
          System.out.println("Enter the string"+j);
          obj.add(sc.nextLine());
        }
        System.out.println("Currently the array list has following elements:"+obj);
        System.out.println("Type of String Operation");
        System.out.println("1. Append - add at end  \n2. Insert – add at particular index \n3. Search \n4. List all
string starts with given letter");
        System.out.println("Select the Operation");
        choice=sc.nextInt();
        switch(choice)
        {
          case 1 :
                Scanner scan=new Scanner(System.in);
                System.out.println("Enter the String to Append at the end:");
                String append=scan.nextLine();
                obj.add(append);
                System.out.println("Current array list is:"+obj);
             break;
          case 2 :
                Scanner scan1=new Scanner(System.in);
                System.out.println("Enter the String to Insert at a particular index:");
                String b =scan1.nextLine();
                System.out.println("Enter the index:");
                i =sc.nextInt();
                obj.add(i, b);
                System.out.println("Current array list is:"+obj);
         break;
          case 3 :
                Scanner scan2=new Scanner(System.in);
                System.out.println("Enter the String to search:");
                String st1 =scan2.nextLine();
                if (obj.contains(st1))
                {
                     System.out.println("Found the String");
                }
                else
                {
                     System.out.println("There is no such String");
                }
             break;
          case 4 :
                Scanner scan3=new Scanner(System.in);
                System.out.println("Enter the character to List all string starts with given letter");
                String startString =scan3.nextLine();
                for (String str : obj)
                {
                    if(str.startsWith(startString))
                        System.out.println(str);
                               }
              System.out.println("Current array list is:"+obj);
   }
  }
}
```

INPUT AND OUTPUT:

```
Enter the total number of strings to be entered into the arraylist : 5
Enter the string1: Hari
Enter the string2: ram
Enter the string3: prakash
Enter the string4: sanjeve
Enter the string5: harray
Currently the array list has following elements:[, Hari, ram, prakash, sanjeve, harray]
Type of String Operation
1. Append - add at end
2. Insert – add at particular index
3. Search
4. List all string starts with given letter
Select the Operation:: 1
Enter the String to Append at the end:raj
Current array list is:[, Hari, ram, prakash, sanjeve, harray, raj]
```

VIVA QUESTIONS (PRELAB & POSTLAB)

1. Can we use String in switch case?
2. How to convert String to byte array and vice versa?
3. Why String is popular HashMap key in Java?
4. Why String is immutable or final in Java
5. What is String in Java? String is a data type?

RESULT

Thus the string operations using Array List with the following functions for Append - add at end, Insert – add at particular index, Search and list all string starts with given letter.

TOPIC BEYOND SYLLABUS

Ex. No: 13 Implementation of Generic Function

AIM:

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

PRELAB DISCUSSION:

Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods, or with a single class declaration, a set of related types, respectively. Generics also provide compile-time type safety that allows programmers to catch invalid types at compile time. Using Java Generic concept, we might write a generic method for sorting an array of objects, then invoke the generic method with Integer arrays, Double arrays, String arrays and so on, to sort the array elements. You can write a single generic method declaration that can be called with arguments of different types. Based on the types of the arguments passed to the generic method, the compiler handles each method call appropriately. Following are the rules to define Generic Methods −

- All generic method declarations have a type parameter section delimited by angle brackets (< and >) that precedes the method's return type (< E > in the next example).
- Each type parameter section contains one or more type parameters separated by commas. A type parameter, also known as a type variable, is an identifier that specifies a generic type name.
- The type parameters can be used to declare the return type and act as placeholders for the types of the arguments passed to the generic method, which are known as actual type arguments.
- A generic method's body is declared like that of any other method. Note that type parameters can represent only reference types, not primitive types (like int, double and char)

ALGORITHM:

```
1. Start the Process
2. Create a array of number and array of strings and pass it to generic function.
3. If the array is Integer type
       3.1 Assume first element as MAX
       3.2 Compare [Numeric Perspetive] this element with MAX
       3.3 If it is greater than MAX then store current element as MAX
       3.4 Else do nothing
       3.5 Goto step 3.1 until all the elements has been processed.
4. If the array is String type
       4.1 Assume first element as MAX
       4.2 Compare [Dictionary Perspective] this element with MAX
       a.  If it is greater than MAX then store current element as MAX
       4.4 Else do nothing
       4.5 Goto step 3.1 until all the elements has been processed.
5.     Stop the Process
```

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac Genericfunction.java
3.C:\java Genericfunction
```

SOURCE CODE:

```
class GenericMax {
        public <T extends Comparable<T>> void maxFinder (T[] array){ T max
                = array[0];
                for(T element: array){
                        System.out.println(element);
                        if(element.compareTo(max) > 0)
                                max = element;
                }
                System.out.println("Max is : "+max);
        }
}

public class Main {

        public static void main(String[] args) { GenericMax
                max = new GenericMax(); Integer[] numbers
                = {14,3,42,5,6,10}; String[] strings =
                {"R","Ra","Raj"}; max.maxFinder(numbers);
                max.maxFinder(strings);
        }

}
```

INPUT AND OUTPUT:

```
14
3
42
5
10
Max is 42
Valliammai
Engineering
College
Max is : Engineering
```

VIVA QUESTIONS(PRELAB & POSTLAB):

1. How to write parametrized class in Java using Generics
2. Can you pass List<String> to a method which accepts List<Object>
3. Can we use Generics with Array?
4. Is it Possible to Declared a Multiple Bounded Type Parameter?

5. What is a Generic Type Parameter

RESULT:

The Java program to find maximum value from the given type of elements using generic functions was implemented.

TOPIC BEYOND SYLLABUS

Ex.No.14 Implementation of JDBC

AIM

To develop a simple OPAC system for library using event-driven and concurrent programming paradigms of Java. Use JDBC to connect to a back-end database.

PRELAB DISCUSSION:

The JDBC (Java Database Connectivity) API defines interfaces and classes for writing database applications in Java by making database connections. Using JDBC you can send SQL, PL/SQL statements to almost any relational database. JDBC is a Java API for executing SQL statements and supports basic SQL functionality. It provides RDBMS access by allowing you to embed SQL inside Java code. Because Java can run on a thin client, applets embedded in Web pages can contain downloadable JDBC code to enable remote database access. You will learn how to create a table, insert values into it, query the table, retrieve results, and update the table with the help of a JDBC Program example. Although JDBC was designed specifically to provide a Java interface to relational databases, you may find that you need to write Java code to access non-relational databases as well. Java application calls the JDBC library. JDBC loads a driver which talks to the database In general, to process any SQL statement with JDBC, you follow these steps: ① Establishing a connection. ② Create a statement. ③ Execute the query. ④ Process the ResultSet object. ⑤ Close the connection.

JDBC-ODBC Bridge

As its name JDBC-ODBC bridge, it acts like a bridge between the Java Programming Language and the ODBC to use the JDBC API. To use the JDBC API with the existing ODBC Sun Microsystems (Now Oracle Corporation) provides the driver named JdbcOdbcDriver. Full name of this class is sun.jdbc.odbc.JdbcOdbcDriver.

JDBC ODBC Connection

To Connect the JDBC and ODBC we should have a database. Then we would be required to create a DSN to use JDBC ODBC Bridge driver. The DSN (Data Source Name) specifies the connection of an ODBC to a specific server.

How To Create DSN In Java.

To create a DSN we would be required to follow some basic steps. These steps are as follows : Here I am using the MS-Access as database system. Before creating DSN to use MS-Access with JDBC technology the database file should be created in advance.

- First step is to go through the Control Panel -> Administrative Tools -> Data Sources (ODBC).
- Go to the tab System DSN in the ODBC Data Source Administrator dialog box then, Click on Add button -> select MS Access Driver (*.mdb) -> click on Finish button.
- In the next step a dialog box ODBC Microsoft Access Setup will be opened then provide the field's values corresponding to the field's label (i.e. provide the DSN name as you wish) and then click on the "Select" button.
- In the next step a new dialog box "Select database" will be opened. Here you have to select the directory where you have stored your file and provide the name in the place of the Database Name and then press ok -> ok -> ok.

ALGORITHM:

1. Create a Database
2. Create a table employee in Database created.
3. Insert the fields into the employee table with the following fields: name, rollno, course, subject, marks.
4. Create another panel consisting of buttons UserID, BookID, Return/Renewal,Fine.
5. Associate these buttons with listeners(with Transaction Database).

PROCEDURE:

To execute a java program we require setting a class path:

```
1.C:\set path= C:\Program Files\Java\jdk1.6.0\bin;.;
2.C:\javac OpacSystem.java
3.C:\javac OpacSystem
```

SOURCE CODE:

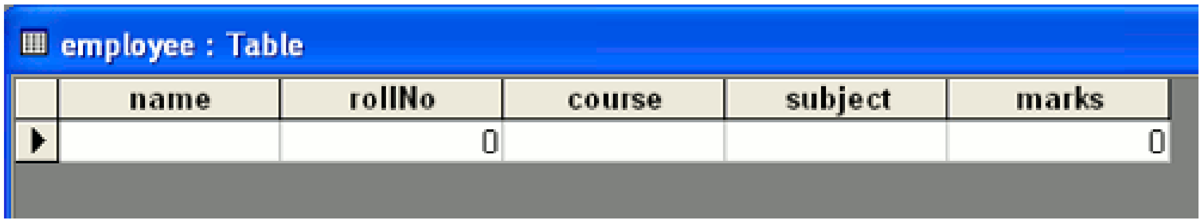
```
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
public class JdbcOdbcExample
 {
        public static void main(String args[])
        {
        Connection con = null;
        PreparedStatement stmt = null;
        try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection("jdbc:odbc:swing");

        String sql ="INSERT INTO employee (name, rollNo, course, subject, marks) VALUES" +
                        "('Deepak', 10, 'MCA', 'Computer Science', 85)";
        stmt = con.prepareStatement(sql);
        int i = stmt.executeUpdate();
        if(i > 0 )
        {
                System.out.println("Record Added Into Table Successfully.");
        }
        } catch (SQLException sqle) {
                System.out.println(sqle.getNextException());
        } catch (ClassNotFoundException e) {
                System.out.println(e.getException());
        } finally {
        try {
        if (stmt != null) {
        stmt.close();
```

```
            stmt = null;
            }
            if (con != null) {
            con.close();
            con = null;
            }
            } catch (Exception e) {
            System.out.println(e);
            }
            }
            }
}
```

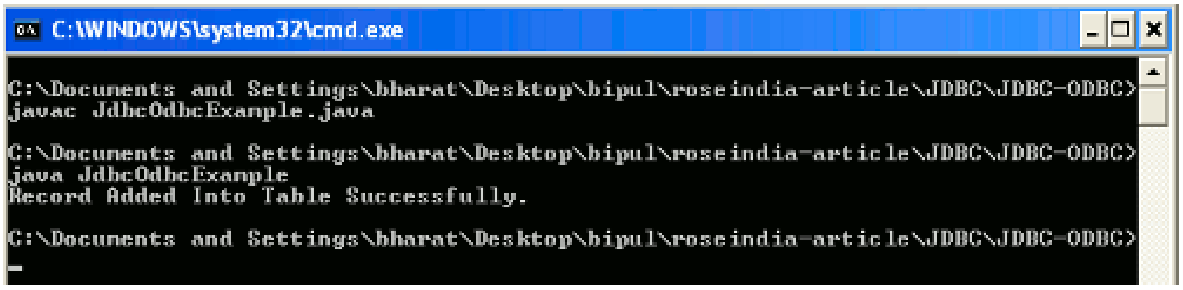
INPUT AND OUTPUT:

Table Before Inserting the record into it.


employee : Table

	name	rollNo	course	subject	marks
▶		0			0

When you will execute the above example you will get the output at console as follows :

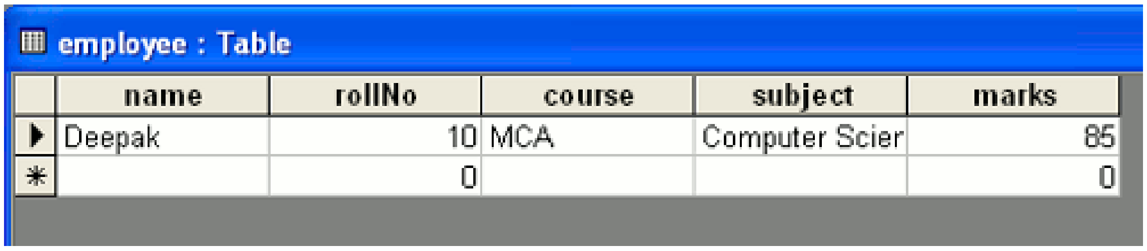


```
C:\Documents and Settings\bharat\Desktop\bipul\roseindia-article\JDBC\JDBC-ODBC>
javac JdbcOdbcExample.java

C:\Documents and Settings\bharat\Desktop\bipul\roseindia-article\JDBC\JDBC-ODBC>
java JdbcOdbcExample
Record Added Into Table Successfully.

C:\Documents and Settings\bharat\Desktop\bipul\roseindia-article\JDBC\JDBC-ODBC>
_
```

And then the Table after inserting record will be seen as follows :


employee : Table

	name	rollNo	course	subject	marks
▶	Deepak	10	MCA	Computer Scier	85
*		0			0

VIVA QUESTIONS(PRELAB & POSTLAB):

1. What are different types of Statement?
2. What is connection pooling?
3. Does the JDBC-ODBC Bridge support multiple concurrent open statements per connection?
4. What are the locking system in JDBC
5. What are the main steps in java to make JDBC connectivity?

RESULT:

A simple OPAC system for library using event-driven and concurrent programming paradigms of Java with the use JDBC to connect to a back-end database was implemented.