ANNA UNIVERSITY

JAYA ENGINEERING COLLEGE

THIRUNINRAVUR-602024

(An ISO 9001:2000 Certified Institution)

Email :info@jec.ac.inWebsite:www.jec.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

PRACTICAL RECORD

CS3381 OBJECT ORIENTED PROGRAMMING LABORATORY

NAME : ______________________________

REGNO. : ______________________________

YEAR : ______________________________

SEM : ______________________________

ANNA UNIVERSITY

JAYA ENGINEERING COLLEGE

THIRUNINRAVUR-602 024
(An ISO 9001:2000 Certified Institution)

Email :info@jec.ac.in

Website: www.jec.ac.in

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of work done by

Mr./Ms.________________________________ Reg.No.____________________________

of **B.TECH / Information Technology** in **CS3381 OBJECT ORIENTED PROGRAMMING LABORATORY** in the **III** semester during **AUGUST 2022 to JANUARY 2023.**

Head of the Department Staff-In-Charge

Submitted For Practical Examination Held On ___________________

Internal Examiner **External Examiner**

VISION AND MISSION OF THE INSTITUTE

Vision of the Institute

To Achieve Excellence in Technical Education through Innovative Teaching and Multidisciplinary Research with Professionalism to serve the Global Society.

Mission of the Institute

Jaya Engineering College(JEC)will Endeavor

M1-To provide state of art teaching and learning for engineering and Technology, Research and management studies.

M2-To provide quality education, self discipline and ethical values.

M3-To associate with R&D and industries to have connectivity with the society.

M4-To impart knowledge to become empowered professionals in the field of Engineering and Management.

VISION AND MISSION OF THE DEPARTMENT

Vision of the department

To bridge the gap between the academician and the industry.

To impart quality education through innovative teaching and learning method.

Mission of the department

The Information Technology Department is committed to:

To develop employable software developing graduates with knowledge, skills and ethics; provide them with the professional and soft skills necessary to lead a successful career.

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1

To ensure graduates will be proficient in utilizing the fundamental knowledge of basic sciences, mathematics and Information Technology for the applications relevant to various streams of Engineering and Technology.

PEO 2

To enrich graduates with the core competencies necessary for applying knowledge of computers and telecommunications equipment to store, retrieve, transmit, manipulate and analyze data in the context of business enterprise.

PEO 3

To enable graduates to think logically, pursue lifelong learning and will have the capacity to understand technical issues related to computing systems and to design optimal solutions.

PEO 4

To enable graduates to develop hardware and software systems by understanding the importance of social, business and environmental needs in the human context.

PEO 5

To enable graduates to gain employment in organizations and establish themselves as professionals by applying their technical skills to solve real world problems and meet the diversified needs of industry, academia and research.

PROGRAMME OUTCOMES

ENGINEERING GRADUATES WILL BE ABLE TO:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: 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 the public health and safety, and the cultural, societal, and environmental considerations.

PO4: 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.

PO5: 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.

PO6: 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.

PO7: 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.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with 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.

PO11: 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.

PO12: 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 OBJECTIVES (PSOs)

1. To create, select, and apply appropriate techniques, resources, modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

2. To manage complex IT projects with consideration of the human, financial, ethical and environmental factors and an understanding of risk management processes, and operational and policy implications.

SYLLABUS

CS3381 OBJECT ORIENTED PROGRAMMING LABORATORY L T P C 0 0 3 1.5

COURSE OBJECTIVES:

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

LISTOFEXERCISES:

1. Solve problems by using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)

2. Develop stack and queue data structures using classes and objects.

3. Develop a java application with an 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 funds. Generate pay slips for the employees with their gross and net salary.

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

5. Solve the above problem using an interface.

6. Implement exception handling and creation of user defined exceptions.

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

8. Write a program to perform file operations.

9. Develop applications to demonstrate the features of generics classes.

10. Develop applications using JavaFX controls, layouts and menus.

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

Total: 45 Periods

COURSE OUTCOME

On completion of this course, the students will be able to

CO1: Design and develop java programs using object-oriented programming concepts

CO2: Develop simple applications using object-oriented concepts such as package, exceptions

CO3: Implement multithreading, and generics concepts

CO4: Create GUIs and event driven programming applications for real world problems

CO5: Implement and deploy web applications using Java

CO's-PO's&PSO'sMAPPING

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	1	2	1	-	-	-	-	1	2	2	2			
2	2	1	3	1	-	-	-	-	2	3	3	2			
3	2	2	1	2	1	-	-	-	1	2	1	3			
4	2	2	1	3	-	-	-	-	3	1	1	1			
5	1	3	3	1	3	-	-	-	1	1	1	1			
AVg.	2	2	2	2	2	-	-	-	2	2	2	2			

INDEX

Ex.No: 1 **STUDY MATERIALS**

Date:

Aim:

To explore the applications of java and the features of java.

Introduction to Java

Java is a high-level, general-purpose, object-oriented, and secure programming language developed by James Gosling at Sun Microsystems, Inc. in 1991. It is formally known as OAK. In 1995, Sun Microsystem changed the name to Java. In 2009, Sun Microsystem takeover by Oracle Corporation.

Features of Java

- **Simple:** Java is a simple language because its syntax is simple, clean, and easy to understand. Complex and ambiguous concepts of C++ are either eliminated or re-implemented in Java. For example, pointer and operator overloading are not used in Java.
- **Object-Oriented:** In Java, everything is in the form of the object. It means it has some data and behavior. A program must have at least one class and object.
- **Robust:** Java makes an effort to check error at run time and compile time. It uses a strong memory management system called garbage collector. Exception handling and garbage collection features make it strong.
- **Secure:** Java is a secure programming language because it has no explicit pointer and programs runs in the virtual machine. Java contains a security manager that defines the access of Java classes.
- **Platform-Independent:** Java provides a guarantee that code writes once and run anywhere. This byte code is platform-independent and can be run on any machine.
- **Portable:** Java Byte code can be carried to any platform. No implementation-dependent features. Everything related to storage is predefined, for example, the size of primitive data types.
- **High Performance:** Java is an interpreted language. Java enables high performance with the use of the Just-In-Time compiler.
- **Distributed:** Java also has networking facilities. It is designed for the distributed environment of the internet because it supports TCP/IP protocol. It can run over the internet. EJB and RMI are used to create a distributed system.
- **Multi-threaded:** Java also supports multi-threading. It means to handle more than one job a time.

Application

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc.

Types of Java Applications

There are mainly 4 types of applications that can be created using Java programming:

1) Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2) Web Application

An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3) Enterprise Application

An application that is distributed in nature, such as banking applications, etc. is called an enterprise application. It has advantages like high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4) Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

Parameters used in First Java Program

Let's see what is the meaning of class, public, static, void, main, String[], System.out.println().

- **class** keyword is used to declare a class in Java.
- **public** keyword is an access modifier that represents visibility. It means it is visible to all.
- **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method. So, it saves memory.
- **void** is the return type of the method. It means it doesn't return any value.
- **main** represents the starting point of the program.
- **String[] args** or **String args[]** is used for command line argument. We will discuss it in coming section.
- **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class. We will discuss the internal working of System.out.println() statement in the coming section.

JDK, JRE, and JVM

1. A summary of JVM
2. Java Runtime Environment (JRE)
3. Java Development Kit (JDK)

JVM

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. There are three notions of the JVM: *specification*, *implementation*, and *instance*.

The JVM performs the following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.are available for many hardware and software platforms (i.e. JVM is platform dependent).A specification where working of Java Virtual Machine is specified.

The implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.

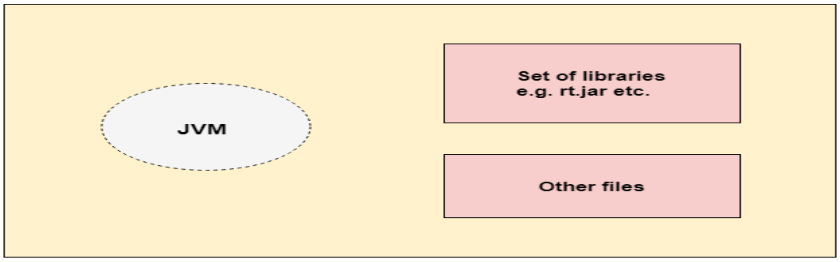
An implementation Its implementation is known as JRE (Java Runtime Environment).

Runtime Instance Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

JRE

It is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

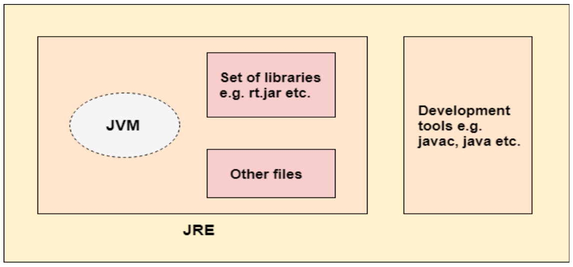
The implementation of JVM is also actively released by other companies besides Sun Micro Systems.



JRE

JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.



JDK

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform
- The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.

Ex. No:2a

Linear Search

Date:

Aim:

To perform linear search in an array, to find if the given element is present or not.

Algorithm:

1. Start the program.
2. Traverse the array
3. Match the key element with array element.
4. If key element is found, return the index position of the array element.
5. If key element is not found, return -1.
6. Stop.

Program:

```
class LinearSearch

{

    public static int LSearch(int arr[], int key)

    {

        for(int i=0;i<arr.length;i++)

        {

            if(arr[i]==key)

            {

                return i;

            }

        }

      return -1;
```

```
        }
        public static void main(String a[])
        {
            int a1[]={10,20,30,50,70,90};
            int key=50;
            System.out.println(key+" is found at index: "+LSearch(a1, key));
        }
    }
```

Output

```
50 is found at index: 3
```

Result

The linear search program has been executed successfully.

Ex. No:2b

Binary Search

Date:

Aim:

To perform binary search in an array, to check if the given element is present or not.

Algorithm:

1. Start the program.
2. Create a class and initialize a binary search function.
3. Search for the element by splitting the array.
4. If the value is found, print the index value, else print not found.
5. Create an array and the key value.
6. Pass the array and the key value to the function.
7. Stop.

Program

```
class BinarySearch

{

        public static void bin(int arr[], int first, int last, int key)

        {

                int mid = (first+last)/2;
                while(first<=last)

                {

                        if(arr[mid]<key)

                        {

                                first = mid+1;

                        }

                        else if(arr[mid]==key)
```

```
            {
                    System.out.println(“Element is found at index: “+mid);
                    break;

                                last = mid-1;

            }
            else
            {

            }
            mid = (first+last)/2;
        }
        if(first>last)
        {
            System.out.println("Element is not found!");
        }
    }
    public static void main(String args[])
    {
        int arr[]={10,20,30,40,50};
        int key = 30;
        int last=arr.length-1;
        bin(arr,0, last, key);
    }
}
```

Output

```
Element is found at index: 2
```

Result

The Binary search program has been executed successfully.

Ex. No: 2c

Bubble Sort

Date:

Aim:

To perform bubble sort in order to sort the elements in an array.

Algorithm:

1. Start the program.
2. Create a class and initialize a bubble sort function.
3. Sort the array by swapping values in the right order.
4. In the main function, create an array.
5. Print the values of the array before sorting
6. Pass the values to the function and print the sorted array.
7. Stop .

Program

```
class BubbleSort

{

    static void BSort(int arr[])

    {

        int n=arr.length; int
        temp=0;

        for(int i=0;i<n;i++)

        {

            for(int j=1;j<n;j++)

            {

                if(arr[j-1]>arr[j])

                {

                    temp=arr[j-1];
                    arr[j-1]=arr[j];
                    arr[j]=temp;

                }

            }

        }

    }

    public static void main(String args[])

    {

        int arr[]={3,60,35,2,45,320,5};
```

```
                System.out.println("Array before Bubble Sort: ");
                for(int i=0;i<arr.length;i++)
                {
                        System.out.print(arr[i]+"");
                }
                System.out.println();
                BSort(arr);
                System.out.println("Array after Bubble Sort: ");
                for(int i=0;i<arr.length;i++)
                {
                        System.out.print(arr[i]+"");
                }
        }
}
```

Output

```
Array before Bubble Sort:

3 60 35 2 45 320 5

Array after Bubble Sort:

2 3 5 35 45 60 320
```

Result

The Bubble sort has been performed on the array and the elements are sorted and executed successfully.

Ex. No:2d

Selection Sort

Date:

Aim:

To perform selection sort to sort the elements in an array

Algorithm:

1. Start the program.
2. Create a class and initialize a selection sort function.
3. Sort the array by checking current value and next value and swapping for correct order.
4. In the main function, create an array.
5. Print the values of the array before sorting.
6. Pass the values to the function and print the sorted array.
7. Stop.

Program

```
class SelectionSort
{
    public static void SSort(int arr[])
    {
        for(int i=0;i<arr.length;i++)
        {
            int index=i;
            for(int j=i+1;j<arr.length;j++)
            {
                if(arr[j]<arr[index])
                {
                    index=j;
                }
            }
            int smallerNumber=arr[index];
            arr[index]=arr[i];
            arr[i]=smallerNumber;
        }
    }
    public static void main(String a[])
    {
        int arr1[]={9,14,3,2,43,11,58,22};
        System.out.println("Before Selection Sort: ");
        for(int i:arr1)
        {
            System.out.print(i+"");
        }
        System.out.println();
        SSort(arr1);
        System.out.println("After Selection Sort: "); for(int i:arr1)
        {
            System.out.print(i+"");
        }
    }
}
```

Output

```
Before Selection Sort:

9 14 3 2 43 11 58 22

After Selection Sort:

2 3 9 11 14 22 43 58
```

Result

The program Selection sort to sort the elements in the array has been performed and executed successfully

Ex. No:2e

Insertion Sort

Date:

Aim:

To perform insertion sort, to sort the elements in an array.

Algorithm:

1. Start the program.
2. Create a class and initialize an insertion sort function.
3. Sort the array using iterations and swapping larger values with smaller values.
4. In the main function, create an array.
5. Print the values of the array before sorting.
6. Pass the values to the function and print the sorted array.
7. Stop.

Program

```
class InsertionSort

{

    public static void ins(int arr[])

    {

        int n=arr.length; for(int
        j=1;j<n;j++)

        {

            int key=arr[j]; int
            i=j-1;

            while((i>-1)&&(arr[i]>key))

            {

                arr[i+1]=arr[i]; i--;
            }
            arr[i+1]=key;
        }
    }
    public static void main(String a[])
    {
       int a1[]={9,14,3,2,43,11,58,22};
       System.out.println("Before Insertion Sort: "); for(int i:a1)
       {
        System.out.print(i+"");
       }
       System.out.println(); ins(a1);
       System.out.println("After Insertion Sort"); for(int
       i:a1)
       {
        System.out.print(i+"");
       }
    }
}
```

Output

```
Before Insertion Sort:

9 14 3 2 43 11 58 22

After Insertion Sort 2 3
9 11 14 22 43 58
```

Result

The program Insertion sort to sort the elements in an array has been executed successfully.

Ex. No:3a

Stack ADT

Date:

Aim:

To implement Stack ADT in Java

Algorithm:

1. Start the program.
2. Create a stack class using push() and pop().
3. Check if the stack is full, print stack overflow, otherwise if the stack is empty, then increment the top pointer to next empty space and add data element to the stack.
4. Check if the stack is empty, print the message stack underflow, otherwise if the stack is not empty access the data element at which top is pointing and pop the element.
5. Print the values after inserting and deleting.
6. Stop.

Program

```
class Stack

{

    private int arr[]; private
    int top;

    private int capacity;
    Stack(int size)

    {

        arr=new int[size];
        capacity=size; top=-
        1;

    }

    public void push(int x)

    {
        if(isFull())
        {
                System.out.println("Stack Overflow");
                System.exit(1);
        }
        System.out.println("Inserting "+x); arr[++top]=x;
}
public int pop()
{
        if(isEmpty())
        {
                System.out.println("Stack Empty"); System.exit(1);
        }
        return arr[top--];
}
public int getSize()
{
        return top+1;
}
public booleanisEmpty()
{
        return top==-1;
}
public booleanisFull()
{
        return top==capacity-1;
}
```

```
        public void printStack()

        {

                for(int i=0;i<=top;i++)

                {

                        System.out.print(arr[i]+",");

                }

        }

        public static void main(String args[])

        {

                Stack stack= new Stack(5);
                stack.push(1); stack.push(2);
                stack.push(3);
                System.out.print("Stack: ");
                stack.printStack(); stack.pop();

                System.out.println("\nAfter popping out: ");
                stack.printStack();

        }

}
```

Output

```
Inserting 1

Inserting 2

Inserting 3

Stack: 1,2,3,

After popping out:

1,2,
```

Result

The Program to implement Stack ADT in Java has executed successfully.

Ex. No:3b

Queue ADT

Date:

Aim:

To implement Queue ADT in Java.

Algorithm:

1. Start the program.
2. Create a class Queue to perform Queue ADT functions.
3. Create a method to check whether the Queue is full, and return true or false.
4. Create a method to check whether the Queue is empty, and return true or false.
5. Create a method to add elements to the Queue.
6. Create a method to delete elements from the Queue.
7. Create a method to display the elements present in the Queue.
8. In the main method, create an object to call the other methods.
9. Print the results.
10. Stop

Program

```
class Queue

{

    int size=5;

    int items[]= new int[size]; int
    front,rear;

    void queue()

    {

        front=-1;
        rear=-1;
    }
    booleanisFull()
    {
        if(front==0 && rear==size-1)
        {
            return true;
        }
        return false;
    }
    booleanisEmpty()
    {
        if(front==-1)
            return true;
        else
            return false;
    }
    void enQueue(int element)
    {
        if(isFull())
        {
            System.out.println("Queue is Full!");
        }
        else
        {
            if(front==-1)
                front=0;
            rear++;
            items[rear]=element;

            System.out.println("Insert "+element);

            }

    }

    int deQueue()

    {

            int element;
            if(isEmpty())

            {

                    System.out.println("Queue is Empty!");

                    return (-1);

            }else

            {

                    element=items[front];
                    if(front<=rear)

                    {

                                    front=-1; rear=-1;

                    }

                    else

                    front++;
                    System.out.println(element+" Deleted");

                     return(element);

            }

    }

    void display()

    {

            int i;

            if(isEmpty())

                    System.out.println("Empty Queue!");

                     else

            {

                    System.out.println("\nFront index ->"+front);
                    System.out.println("Items ->");
                    for(i=front;i<=rear;i++)

                            System.out.print(items[i]+"");

                            System.out.println("\nRear index ->"+rear);

            }

    }

    public static void main(String args[])

    {

            Queue q=new Queue();
            q.deQueue();

            for(int i=1;i<6;i++)

            {

                    q.enQueue(i);

            }

            q.enQueue(6);
```

```
            q.display();

            q.deQueue();

            q.display();

        }

    }
```

Output

```
0     Deleted
Insert 1

Insert 2

Insert 3

Insert 4

Insert 5

Queue is Full!


Front index -> 0
Items ->

1 2 3 4 5

Rear index -> 4


1     Deleted
Empty Queue!
```

Result

The Program to implement Queue ADT has been executed successfully.

Ex. No:4

Payrollof an Employee class using Inheritance

Date:

Aim:

To develop a Java program to generate pay slip for different category of employees using theconcept of inheritance

Algorithm:

1. Start the program.
2. Create a class employee with name, empid, address, mailed, mobile no as members
3. Add Basic Pay (BP) as the member of all the inherited classes.
4. Calculate DA as 97% of BP, HRA as 10% of BP, PF as 12% of BP, staff club fund as 0.1% of BP.
5. Calculate gross salary and net salary
6. Generate payslip for all categories of employees.
7. Create the objects for the inherited classes and invoke the necessary methods to display the Payslip.
8. Stop.

Program

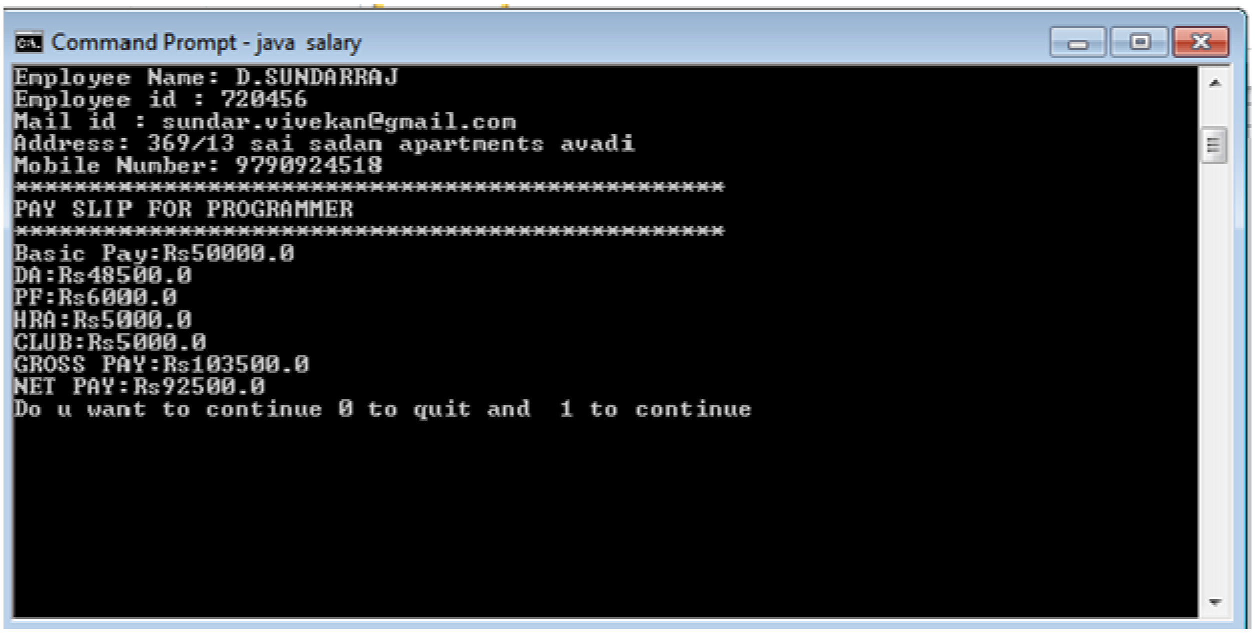
```
import java.util.*;
class employee
{
    int empid;
    long mobile;
   String name, address, mailid;
   Scanner get = new Scanner(System.in);
void getdata()
{
            System.out.println("Enter Name of the Employee");
             name = get.nextLine();
            System.out.println("Enter Mail id");
            mailid = get.nextLine();
            System.out.println("Enter Address of the Employee:");
```

```
             address = get.nextLine();
            System.out.println("Enter employee id ");
             empid = get.nextInt();
            System.out.println("Enter Mobile Number");
             mobile = get.nextLong();
 }
 void display()
 {
            System.out.println("Employee Name: "+name);
            System.out.println("Employee id : "+empid);
            System.out.println("Mail id : "+mailid);
            System.out.println("Address: "+address);
            System.out.println("Mobile Number: "+mobile);
 }
}
class programmer extends employee
{
            double salary,bp,da,hra,pf,club,net,gross;
            void getprogrammer()
            {
            System.out.println("Enter basic pay");
            bp = get.nextDouble();
            }
            void calculateprog()
            {
            da=(0.97*bp);
            hra=(0.10*bp);
            pf=(0.12*bp);
            club=(0.1*bp);
            gross=(bp+da+hra);
            net=(gross-pf-club);
            System.out.println("***********************************************");
            System.out.println("PAY SLIP FOR PROGRAMMER");
            System.out.println("***********************************************");
            System.out.println("Basic Pay:Rs"+bp);
            System.out.println("DA:Rs"+da);
            System.out.println("PF:Rs"+pf);
            System.out.println("HRA:Rs"+hra);
            System.out.println("CLUB:Rs"+club);
            System.out.println("GROSS PAY:Rs"+gross);
            System.out.println("NET PAY:Rs"+net);
            }
}
class asstprofessor extends employee
   {
            double salary,bp,da,hra,pf,club,net,gross;
            void getasst()
            {
```

```
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
        }
        void calculateasst()
        {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("************************************************");
        System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");
        System.out.println("************************************************");
        System.out.println("Basic Pay:Rs"+bp);
        System.out.println("DA:Rs"+da);
        System.out.println("HRA:Rs"+hra);
        System.out.println("PF:Rs"+pf);
        System.out.println("CLUB:Rs"+club);
        System.out.println("GROSS PAY:Rs"+gross);
        System.out.println("NET PAY:Rs"+net);
   }
  }
class associateprofessor extends employee
{
        double salary,bp,da,hra,pf,club,net,gross;
         void getassociate()
        {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
        }
        void calculateassociate()
        {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("************************************************");
        System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");
        System.out.println("************************************************");
        System.out.println("Basic Pay:Rs"+bp);
        System.out.println("DA:Rs"+da);
        System.out.println("HRA:Rs"+hra);
        System.out.println("PF:Rs"+pf);
        System.out.println("CLUB:Rs"+club);
        System.out.println("GROSS PAY:Rs"+gross);
        System.out.println("NET PAY:Rs"+net);
  }
}
class professor extends employee
{
        double salary,bp,da,hra,pf,club,net,gross;
        void getprofessor()
        {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
        }
        void calculateprofessor()
        {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("************************************************");
        System.out.println("PAY SLIP FOR PROFESSOR");
        System.out.println("************************************************");
        System.out.println("Basic Pay:Rs"+bp);
        System.out.println("DA:Rs"+da);
        System.out.println("HRA:Rs"+hra);
        System.out.println("PF:Rs"+pf);
        System.out.println("CLUB:Rs"+club);
        System.out.println("GROSS PAY:Rs"+gross);
        System.out.println("NET PAY:Rs"+net);
        }
  }
class salary
{
  public static void main(String args[])
  {
  int choice,cont;
  do
  {
  System.out.println("PAYROLL");
  System.out.println(" 1.PROGRAMMER \t 2.ASSISTANT PROFESSOR \t 3.ASSOCIATE
  PROFESSOR \t 4.PROFESSOR ");
  Scanner c = new Scanner(System.in);
  choice=c.nextInt();
  switch(choice)
  {
        case 1:
        {
        programmer p=new programmer();
```

```
                p.getdata();
                p.getprogrammer();
                p.display();
                p.calculateprog();
                break;
                }
                case 2:
                {
                asstprofessor asst=new asstprofessor();
                asst.getdata();
                asst.getasst();
                asst.display();
                asst.calculateasst();
                break;
            }
                case 3:
                {
                associateprofessorasso=new associateprofessor();
                asso.getdata();
                asso.getassociate();
                asso.display();
                asso.calculateassociate();
                break;
                }
               case 4:
                 {
                 professor prof=new professor();
                 prof.getdata();
                 prof.getprofessor();
                 prof.display();
                 prof.calculateprofessor();
                 break;
                 }
                 }
          System.out.println("Do u want to continue 0 to quit and  1 to continue ");
          cont=c.nextInt();
          }while(cont==1);
          }
     }
```

Output:



```
Employee Name: D.SUNDARRAJ
Employee id : 720456
Mail id : sundar.vivekan@gmail.com
Address: 369/13 sai sadan apartments avadi
Mobile Number: 9790924518
************************************************
PAY SLIP FOR PROGRAMMER
************************************************
Basic Pay:Rs50000.0
DA:Rs48500.0
PF:Rs6000.0
HRA:Rs5000.0
CLUB:Rs5000.0
GROSS PAY:Rs103500.0
NET PAY:Rs92500.0
Do u want to continue 0 to quit and  1 to continue
```

Result

The Java program to generate pay slip for different category of employees was executed successfully.

Ex. No:5

Shapes using Abstract Class

Date:

Aim:

To write a Java program to calculate the area of rectangle, circle and triangle using the concept of abstract class.

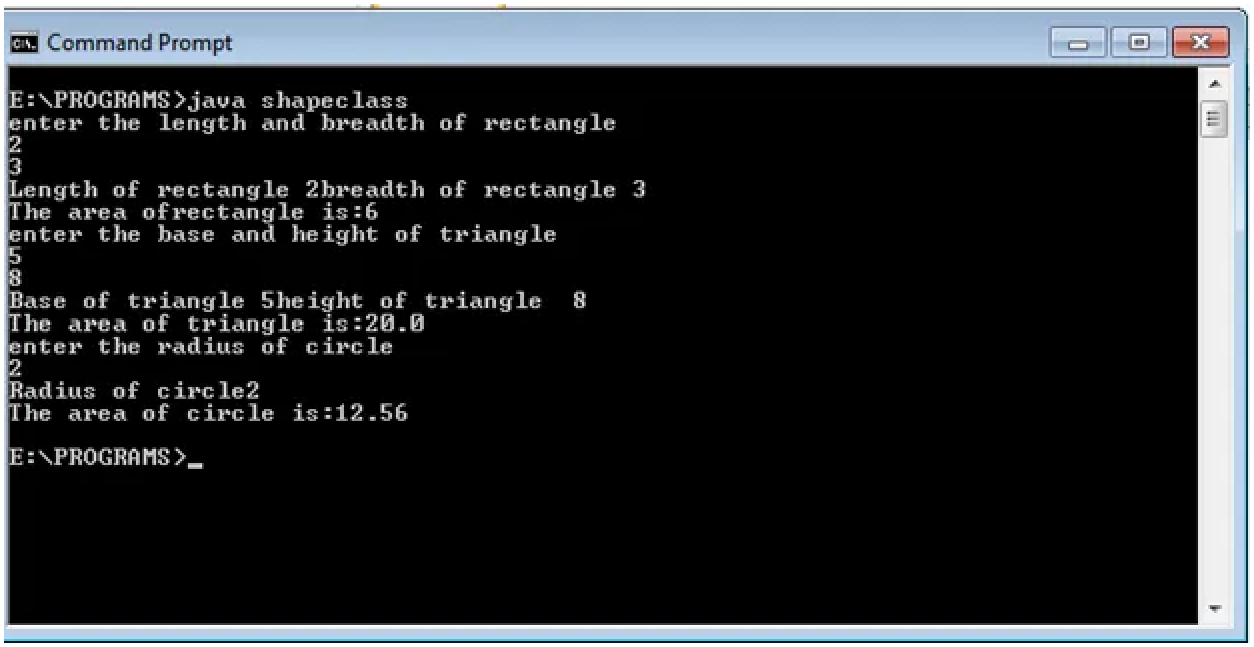
Algorithm:

1. Start the program.
2. Create an abstract class named shape that contains two integers and an empty method named printarea().
3. Provide three classes named rectangle, triangle and circle such that each one of the classes extends the class Shape.
4. Each of the inherited class from shape class should provide the implementation for the method printarea().
5. Get the input and calculate the area of rectangle, circle and triangle.
6. In the shape class, create the objects for the three inherited classes and invoke the methods and display the area values of the different shapes.
7. Stop.

Program:

```
import java.util.*;
abstract class shape
{
         int a,b;
         abstract public void printarea();
}
class rectangle extends shape
{
        public int area_rect;
        public void printarea()
        {
          Scanner s=new Scanner(System.in);
        System.out.println("enter the length and breadth of rectangle");
          a=s.nextInt();
          b=s.nextInt();
        area_rect=a*b;
        System.out.println("Length of rectangle "+a +"breadth of rectangle "+b);
        System.out.println("The area ofrectangle is:"+area_rect);
         }
 }
class triangle extends shape
{
         double area_tri;
         public void printarea()
        {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the base and height of triangle");
        a=s.nextInt();
        b=s.nextInt();
        System.out.println("Base of triangle "+a +"height of triangle  "+b);
        area_tri=(0.5*a*b);
        System.out.println("The area of triangle is:"+area_tri);
}
}
class circle extends shape
{
        double area_circle;
        public void printarea()
        {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the radius of circle");
        a=s.nextInt();
        area_circle=(3.14*a*a);
        System.out.println("Radius of circle"+a);
        System.out.println("The area of circle is:"+area_circle);
 }
}
public class shapeclass
{
 public static void main(String[] args)
{
            rectangle r=new rectangle();
            r.printarea();
             triangle t=new triangle();
            t.printarea();
            circle r1=new circle();
            r1.printarea();
}
}
```

Output:



```
E:\PROGRAMS>java shapeclass
enter the length and breadth of rectangle
2
3
Length of rectangle 2breadth of rectangle 3
The area ofrectangle is:6
enter the base and height of triangle
5
8
Base of triangle 5height of triangle  8
The area of triangle is:20.0
enter the radius of circle
2
Radius of circle2
The area of circle is:12.56

E:\PROGRAMS>_
```

Result

The Program to calculate the area of circle, rectangle and triangle has been executed successfully.

Ex. No:6

Calculating Area using Interface

Date:

Aim:

To write a Java program to calculate the area of rectangle, circle and triangle using the concept of abstract class.

Algorithm:

1. Start the program.
2. Declare three different classes for rectangle, square, and circle.
3. Declare the interface
4. Implement the interface methods.
5. Call the corresponding methods as per the number of arguments or their data types.
6. Display the result.
7. Stop

Program:

```
interface Area  //interface defined
{
   float pi=3.14F;
   float compute(float x, float y);
}

class Rectangle implements Area
{
public float compute (float x, float y)
{
return(x*y);
}

}

class Circle implements Area
{
public float compute(float x, float y)
{
return(pi*x*x);
}
}
```

```
class Triangle implements Area
{
public float compute (float x, float y)
{
return((x*y)/2);
}


class InterfaceTest2
{
public static void main(String args[])
{
Rectangle rect=new Rectangle();
Circle cir=new Circle();
Triangle tr= new Triangle();
Area a;
a=rect;

System.out.println("Area of rectangle : "+a.compute(5,10));
a=cir;
System.out.println("Area of circle : "+a.compute(5,0));
a=tr;
System.out.println("Area of Triangle : "+a.compute(5,7));

}
}
```

Output:

```
Area of rectangle : 50.0
Area of circle : 78.5
Area of Triangle:17.5
```

RESULT:

Thus the Java program to calculate the area of various shapes using interface is implemented and output was verified successfully

Ex. No:7

Exception Handling

Date:

Aim:

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

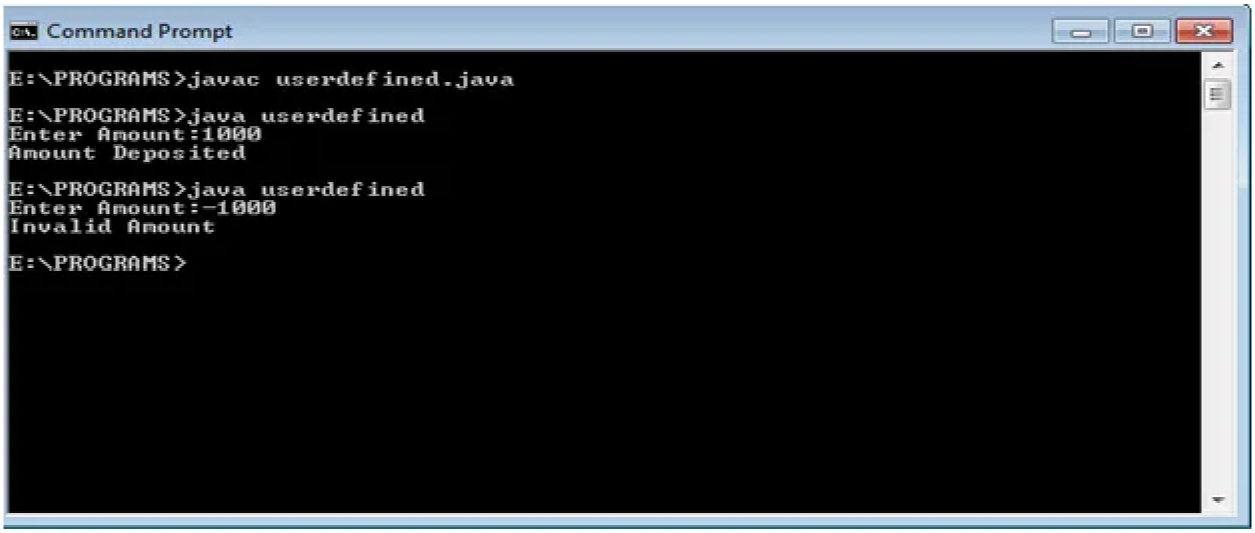
Algorithm:

1. Start the program.
2. Create a class which extends exception class.
3. Create a constructor which receives the string as argument
4. Get the amount as input from the user
5. If the amount is negative, the exception will be generated
6. Using the exception handling mechanism, the thrown exception is handled by the catch construct.
7. After the exception is handled, the string "Invalid Amount" will be displayed
8. If the amount is greater than 0, the string "Amount Deposited" will be displayed
9. Stop

Program 1:

```
import java.util.Scanner;
 class NegativeAmtException extends Exception
 {
 String msg;
NegativeAmtException(String msg)
 {
      this.msg=msg;
 }
 public String toString()
 {
       return msg;
 }
}
public class userdefined
{
 public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.print("Enter Amount:");
int a=s.nextInt();
try
{
if(a<0)
 {
throw new NegativeAmtException("Invalid Amount");
 }
System.out.println("Amount Deposited");
 }
catch(NegativeAmtException e)
{
System.out.println(e);
}
}
}
```

Output:



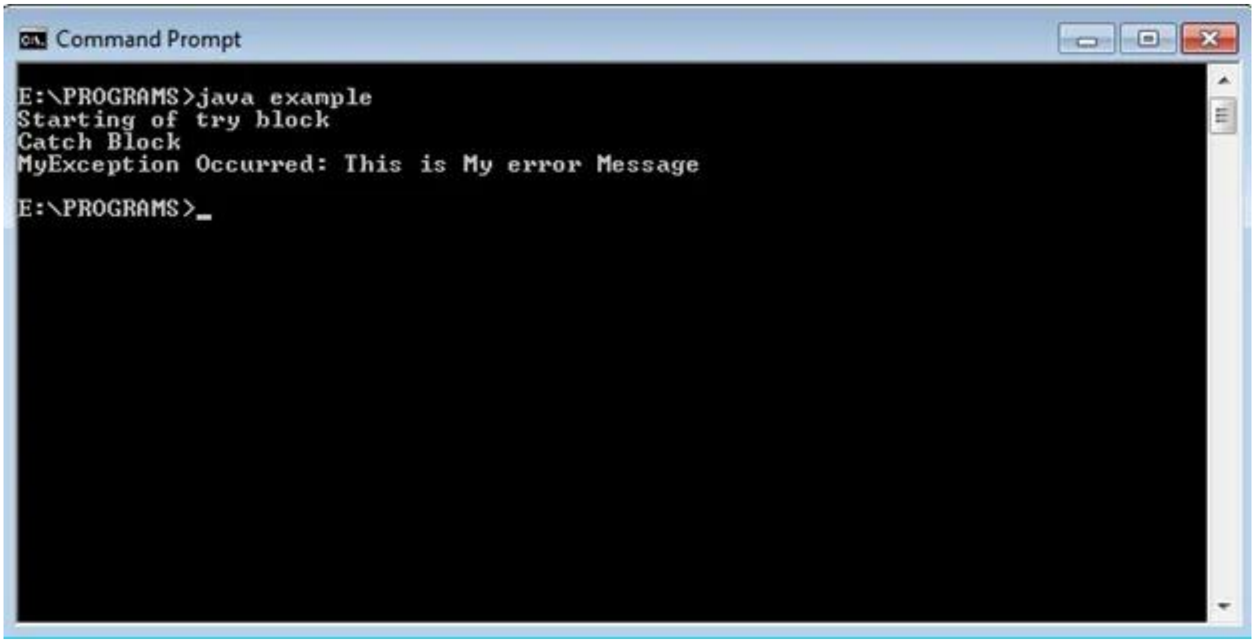
```
E:\PROGRAMS>javac userdefined.java

E:\PROGRAMS>java userdefined
Enter Amount:1000
Amount Deposited

E:\PROGRAMS>java userdefined
Enter Amount:-1000
Invalid Amount

E:\PROGRAMS>
```

Program 2:

```
class MyException extends Exception{
String str1;
MyException(String str2)
 {
str1=str2;
  }
  public String toString()
 {
return ("MyException Occurred: "+str1) ;
 }
}
class example
```

```
{
 public static void main(String args[])
{
try
{
System.out.println("Starting of try block");
throw new MyException("This is My error Message");
}
catch(MyException exp)
{
System.out.println("Catch Block") ;
System.out.println(exp) ;
}
}}
```

Output:



```
E:\PROGRAMS>java example
Starting of try block
Catch Block
MyException Occurred: This is My error Message

E:\PROGRAMS>_
```

Result:

Thus a java program to implement user defined exception handling has been implemented and executed successfully.

Ex. No:8

Multithreaded Application

Date:

Aim:

To WriteaJavaprogramthatimplementsamulti-threadapplicationthathasthree threads. First thread generates a random integer for every 1 second; secondthreadcomputesthesquareofthenumberandprints;thirdthreadwillprintthevalue of cube of the number.

Algorithm:

1. Start the program.
2. Create a class even which implements first thread that computes the square of the number.
3. run() method implements the code to be executed when thread gets executed.
4. Create a class odd which implements second thread that computes the cube of the number.
5. Create a third thread that generates random number. If the number is even, it displays the square of the number. If the random number generated is odd, it displays the cube of the given number
6. The multithreading is performed and the task is switched between multiple threads.
7. The sleep() method makes the thread to suspend for the specified time.
8. Stop

Program:

```
import java.util.Random;

classSquareextendsThread

{
    int x;

    Square(intn)

    {

            x=n;
    }

publicvoidrun()
{

int sqr =x*x;
System.out.println("Squareof" +x +"= "+sqr);
}
}
classCubeextendsThread
{

int x;

Cube(intn)

{
x=n;
}

publicvoidrun()
{

intcub=x*x* x;
System.out.println("Cubeof "+x+"= "+cub );
}
}



classNumberextendsThread
{
```

```
publicvoidrun()
{

Randomrandom=new

Random();for(int i =0;i<10;i++)

{

int randomInteger = random.nextInt(100);

System.out.println("RandomIntegergenerated:"+randomInteger);

Squares =newSquare(randomInteger);

s.start();
Cube c = new

Cube(randomInteger);c.start();

try{
Thread.sleep(1000);
} catch (InterruptedException ex)

{

System.out.println(ex);

}
}
}
}



publicclassmultithreaddemo{
publicstaticvoidmain(Stringargs[])
{

Number n = new

Number();n.start();

}
}
```

Output:

```
Random Integer generated : 82

Squareof 82=6724

Cubeof82 =551368
Random Integer generated : 34

Squareof 34=1156

Cubeof34 =39304
Random Integer generated : 17

Squareof 17=289

Cubeof17=4913
Random Integer generated : 84

Squareof 84=7056

Cubeof84 =592704
Random Integer generated :

34Cubeof34=39304

Squareof34=1156
Random Integer generated :

51Squareof 51=2601

Cubeof51 =132651
Random Integer generated :

38Squareof 38=1444

Cubeof38 =54872
Random Integer generated :

94Squareof 94=8836

Cubeof94 =830584
Random Integer generated :

65Cubeof65=274625

Squareof65=4225
Random Integer generated
```

```
:75Squareof 75=5625

Cubeof75 =421875
```

Result:

Thus a java program for multithreaded application has been implemented and executed successfully.

Ex. No:9

File Operations in Java

Date:

Aim:

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

Algorithm:

1. Start the program.
2. Create a class that implements the serializable interface.
3. Open or create a new file using FileOutputStream.
4. Create an objectOutputStream giving the above FileOutputStream as an argument to the constructor.
5. Use objectOutputStream.witeobject method to write the object you want to the file.
6. close the object fn.
7. End the program

Program:

```
import java.io.*;
import java.util.*;
class record
{
Scanner input=new Scanner(System.in);
BufferedReaderbr=new BufferedReader(new InputStreamReader(System.in));
int roll;
String name;

void insert()
{
try
{
System.out.println("\n Enter Roll No:: ");
roll=input.nextInt();
System.out.println("\n Enter Name:: ");
name=br.readLine();
FileOutputStreamfout=new FileOutputStream("Student.txt",true);
byte buff[]=name.getBytes();
fout.write((char)roll);
fout.write('\n');
fout.write(buff);
fout.write('\n');
fout.close();
}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}
void delete(int r)
{
try
{
FileInputStream fin = new FileInputStream("student.txt");
FileOutputStreamfout=new FileOutputStream("temp.txt");


while(true)
{
int a=fin.read();
if(a==-1)
break;
if(a!=r)
{
fout.write((byte)a);
a=fin.read();
```

```
fout.write(a);
while(true)
{
a=fin.read();
if(a==’\n’)
break;
fout.write(a);
}
fout.write(a);
}
else
{
a=fin.read();
do{
a=fin.read();
}while(a!=’\n’);
}
}
fin.close();
fout.close();
File f=new File(“student.txt”);
File f1=new File(“temp.txt”);
if(f.delete())
f1.renameTo(new File(“student.txt”));
else
System.out.println(“Error”);


}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}
void showdata(int r)
{
try
{
FileInputStream fin=new FileInputStream(“student.txt”);
while(true)
{


int a=fin.read();
if(a==-1)
break;
if(a==r)
{
System.out.println(“roll no:: “+a);
```

```
a=fin.read();
System.out.println(“Name:: “);
do
{
a=fin.read();
System.out.print((char)a);
}while(a!=’\n’);
}
else
{
do
{
a=fin.read();
}while(a!=’\n’);
}


}


}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}


void show()
{
try
{


FileInputStream fin=new FileInputStream(“student.txt”);
while(true)
{
int a=fin.read();
if(a==-1)
break;
System.out.println(“roll no:: “+a);
a=fin.read();
System.out.println(“Name:: “);
do
{
a=fin.read();
System.out.print((char)a);
}while(a!=’\n’);
```

```
        }


        }
        catch(Exception e)
        {
        System.out.println(e.getMessage());
        }
        }
        }


public class Student extends record
{
public static void main(String args[])
{
try
{
while(true){
System.out.println(“\nEnter your choice : “);
System.out.println(“1:Insert.”);
System.out.println(“2:Delete.”);
System.out.println(“3:View record.”);
System.out.println(“4:View All.\n”);
int choice=input.nextInt();
record r=new record();
switch(choice)
{
case 1:
r.insert();
break;
case 2:
System.out.println(“Enter roll no. : “);
int roll=input.nextInt();
r.delete(roll);
break;
case 3:
System.out.println(“Enter roll no. : “);
roll=input.nextInt();
r.showdata(roll);
break;
case 4:
r.show();
break;
default:
System.out.println(“Invalid Entry !!”);
break;
}
}
```

```
}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}
}
```

Output:

```
Enter your choice:
1.Insert.
2.Delete.
3.View record.
4.View All.
1
Enter roll no:
10
Enter name:
Amit
```

Result:

Thus a java program for File operations has been implemented and executed successfully.

Ex. No:10

Application to demonstrate features of Generic Class

Date:

Aim:

To develop a Java program to find the maximum value from the given type of elements using ageneric function.

Algorithm:

1. Start the program.
2. Create a class MyClass to implement generic class and generic methods
3. Get the set of the values belonging to specific data type
4. Create the objects of the class to hold integer, character and double values
5. Create the method to compare the values and find the maximum value stored in the array
6. Invoke the method with integer, character or double values. The output will be displayed based on the data type passed to the method.
7. Stop

Program

```
class MyClass<T extends Comparable<T>>

{

        T[] vals;

        MyClass(T[] o)

        {

                vals=o;

        }

        public T min()

        {

                T v=vals[0];

                for(int i=1;i<vals.length;i++)
                if(vals[i].compareTo(v)<0)
                v=vals[i];

                return v;

        }

        public T max()

        {

                T v=vals[0];

                for(int i=1;i<vals.length;i++)
                if(vals[i].compareTo(v)>0)
                v=vals[i];

                return v;

        }

}

public class GenericsNumber

{

        public static void main(String args[])
        {
                Integer inums[]={10,2,5,4,6,1};

                Character chs[]={'v','p','s','a','n','h'};

                Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};

                MyClass<Integer>iob= new MyClass<Integer>(inums);
                MyClass<Character> cob=new MyClass<Character>(chs);
                MyClass<Double> dob=new MyClass<Double>(d);
                System.out.println("Max value in inums: "+iob.max());
                System.out.println("Min value in inums: "+iob.min());
                System.out.println("Max value in chs: "+cob.max());
                System.out.println("Min value in chs: "+cob.min());
                System.out.println("Max value in d: "+dob.max());
                System.out.println("Min value in d: "+dob.min());

        }

    }
```

Output

```
Max value in inums: 10
Min value in inums: 1
Max value in chs: v
Min value in chs: a
Max value in d: 88.3
Min value in d: 10.4
```

Result

The Java program to find the maximum value from the type of elements using generics has been executed successfully

Ex. No: 11

Application for JavaFX Controls, Layouts and Menus

Date:

Aim:

To develop a Java program to generate a JavaFX Application Window

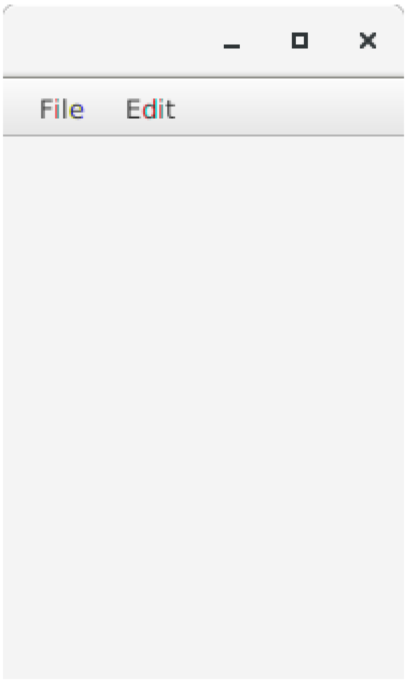
Algorithm:

1. Start the program.
2. Import JavaFX Scene, Layout, Control and Stage.
3. Createa object for Borderpane layout.
4. Create Menubar and menus.
5. Add the menus to the menubar using objects.
6. Add the Menubar to the borderpane layout.
7. Create the Borderpane layout using setTop().
8. Add the layout to the Scene.
9. Display the menu using primaryStage.show().
10. Stop.

Program

```
Package application;

importjavafx.application.Application;
importjavafx.scene.Scene;
importjavafx.scene.control.*;
importjavafx.scene.layout.BorderPane;
importjavafx.stage.Stage;
public classMenuExampleextendsApplication

{
   public static voidmain(String[] args)

       {
           launch(args);
       }

  public voidstart(Stage primaryStage) throwsException

  {


       BorderPane root = newBorderPane();
       Scene scene = newScene(root,200,300);
       MenuBarmenubar = newMenuBar();
       Menu FileMenu = newMenu("File");
       MenuItem filemenu1=newMenuItem("new);
       MenuItem filemenu2=newMenuItem("Save");
       MenuItem filemenu3=newMenuItem("Exit");
       Menu EditMenu=newMenu("Edit");
       MenuItem EditMenu1=newMenuItem("cut");
       MenuItem EditMenu2=newMenuItem("copy");
       MenuItem EditMenu3=newMenuItem("paste");
       EditMenu.getItems().addAll(EditMenu1,EditMenu2,EditMenu3);
       root.setTop(menubar);
       FileMenu.getItems().addAll(filemenu1,filemenu2,filemenu3);
       menubar.getMenus().addAll(FileMenu,EditMenu);
       primaryStage.setScene(scene);
       primaryStage.show();

  }

}
```

Output



Result

The JavaFX program has been executed successfully.

Ex. No: 12 **MINI PROJECT**

Date:

AIM:

Todevelop ajavaprogram todesigna calculatorusing event-drivenprogrammingparadigmofJavawiththefollowingoptions.

a) Decimalmanipulations

b) Scientificmanipulations

ALGORITHM:

1. Importthejavapackages.
2. CreatetheclasscalculatorbyimplementingtheclassJframeandinterface actionListener.
3. Declarethebuttons requiredusingJButton.
4. Designthelayoutofthe calculatorusingthesetLayout,textpanel(),Panel(),Jtextfield(),se tfont()methods.
5. Definetheactionsto beperformed foreach keyusingActionListener.
6. Enablethescientificor standardcalculator usingthemethodmethodadd().
7. Definethemathematicaloperationstobeperformedforthemathematicalsymbols.
8. Selecttherequiredmathematicaloperationsusingswitchasthecalculator.
9. Passtheparametersforthemethods used.
10. Makethe frame visible byusingthe method setVisible().

PROGRAM:

```
import java.awt.*;
import javax.swing.*;
importjava.awt.event.*;
importjavax.swing.event.*;
public class ScientificCalculator extends JFrame implements ActionListener
{
        JTextField tfield;
        double temp, temp1, result,
        a;static doublem1, m2;
        int k = 1, x = 0, y = 0, z =
        0;charch;
        JButtonb1, b2, b3, b4,b5, b6, b7, b8, b9,zero,clr, pow2,exp,plus, min, div, log,rec,mul,eq,
dot, sqrt, sin, cos, tan;
        Container cont;
        JPanel textPanel,buttonpanel;
        ScientificCalculator()
        {
                cont=getContentPane();
                cont.setLayout(newBorderLayout());
                JPaneltextpanel = new JPanel();
                tfield=new JTextField(25);
                tfield.setHorizontalAlignment(SwingConstants.RIGHT);
                tfield.addKeyListener(newKeyAdapter()
                 {
                        publicvoidkeyTyped(KeyEventkeyevent)
                        {charc=keyevent.getKeyChar();
                                if(c>= '0'&& c<='9'){
                                }else {
                                        keyevent.consume();
                                }
                        }
                });

                textpanel.add(tfield);
                buttonpanel=newJPanel();
                buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
                booleant = true;

                b1 = new JButton("1");
                buttonpanel.add(b1);
                b1.addActionListener(this);

                b2 = new JButton("2");
                buttonpanel.add(b2);
                b2.addActionListener(this);

                b3 = new JButton("3");
                buttonpanel.add(b3);
                b3.addActionListener(this);

                b4 = new JButton("4");
                buttonpanel.add(b4);
                b4.addActionListener(this);

                b5 = new JButton("5");
                buttonpanel.add(b5);
                b5.addActionListener(this);
```

```
b6 = new JButton("6");
buttonpanel.add(b6);
b6.addActionListener(this);
b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);

b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);

b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);

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

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

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

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

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

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

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

rec = new JButton("1/x");
buttonpanel.add(rec);
```

```
        rec.addActionListener(this);
        sqrt = new JButton("Sqrt");
        buttonpanel.add(sqrt);
        sqrt.addActionListener(this);

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

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

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

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

        pow2 = new JButton("x^2");
        buttonpanel.add(pow2);
        pow2.addActionListener(this);

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

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

        cont.add("Center", buttonpanel);
        cont.add("North",textpanel);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public void actionPerformed(ActionEvent e)
        {Strings =e.getActionCommand();

          if(s.equals("1")){
                 if(z==0) {
                        tfield.setText(tfield.getText()+"1");
                }else {
                        tfield.setText("");
                tfield.setText(tfield.getText()+"1");
                z=0;
        }
}

  if(s.equals("2")){
          if(z==0) {
                tfield.setText(tfield.getText()+"2");
        }else {
                tfield.setText("");tfield.setText(tfield.getText()+"2");
                z=0;
        }
}

  if(s.equals("3")){
          if(z==0) {
                tfield.setText(tfield.getText()+"3");
        }else {
                tfield.setText("");tfield.setText(tfield.getText()+"3");
                z=0;
        }
}

  if(s.equals("4")){
           if(z==0){
                tfield.setText(tfield.getText()+"4");
        }else {
                tfield.setText("");tfield.setText(tfield.getText()+"4");
                z=0;
        }
}

  if(s.equals("5")){
          if(z==0) {
                tfield.setText(tfield.getText()+"5");
        }else {
                tfield.setText("");tfield.setText(tfield.getText()+"5");
                z=0;
        }
}
```

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

   if(s.equals("7")){
              if(z==0) {
                     tfield.setText(tfield.getText()+"7");
           }else {
                     tfield.setText("");tfield.setText(tfield.getText()+"7");
                     z=0;
           }
}

   if(s.equals("8")){
              if(z==0) {
                     tfield.setText(tfield.getText()+"8");
           }else {
                     tfield.setText("");tfield.setText(tfield.getText()+"8");
                     z=0;
           }
}

   if(s.equals("9")){
              if(z==0) {
                     tfield.setText(tfield.getText()+"9");
           }else {
                     tfield.setText("");tfield.setText(tfield.getText()+"9");
                     z=0;
           }
}

   if(s.equals("0")){
              if(z==0) {
                       tfield.setText(tfield.getText()+"0");
            }else {
                tfield.setText("");tfield.setText(tfield.getText()+"0");
                z=0;
        }
}

if(s.equals("AC")){
        tfield.setText("");
        x=0;
        y=0;
        z=0;
}

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

if(s.equals("1/x")){
        if (tfield.getText().equals(""))
                {tfield.setText("");
        }else {
                a = 1 / Double.parseDouble(tfield.getText());
                tfield.setText("");tfield.setText(tfield.getText()+a);
        }
}

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

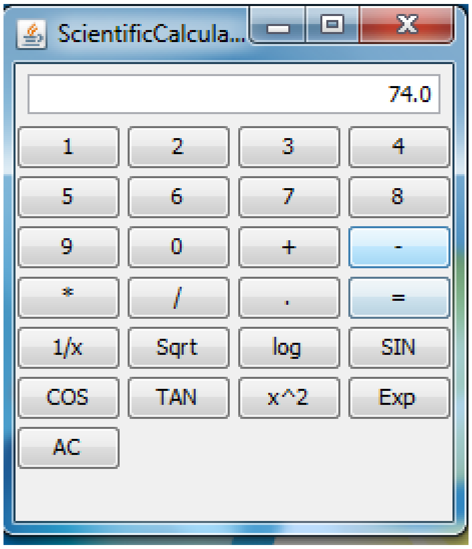
```
if(s.equals("x^2")){
        if (tfield.getText().equals(""))
                {tfield.setText("");
        }else {
                a = Math.pow(Double.parseDouble(tfield.getText()), 2);
                tfield.setText("");
                tfield.setText(tfield.getText()+a);
        }
}

if(s.equals(".")){
        if (y== 0) {
                tfield.setText(tfield.getText() + ".");
                y=1;
        }else {
                tfield.setText(tfield.getText());
        }
}

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

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

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

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

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

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

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

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

if(s.equals("=")){
        if (tfield.getText().equals(""))
                {tfield.setText("");
        }else {
                temp1 = Double.parseDouble(tfield.getText());
                switch(ch){
                case'+':
                        result = temp + temp1;
                        break;
                case'-':
                        result = temp - temp1;
                        break;
                case'/':
                        result=temp /temp1;
```

```
                                        break;
                                case '*':
                                        result = temp * temp1;
                                        break;
                                }
                                tfield.setText("");
                                tfield.setText(tfield.getText() + result);
                                z=1;
                        }
                }
                tfield.requestFocus();
        }

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

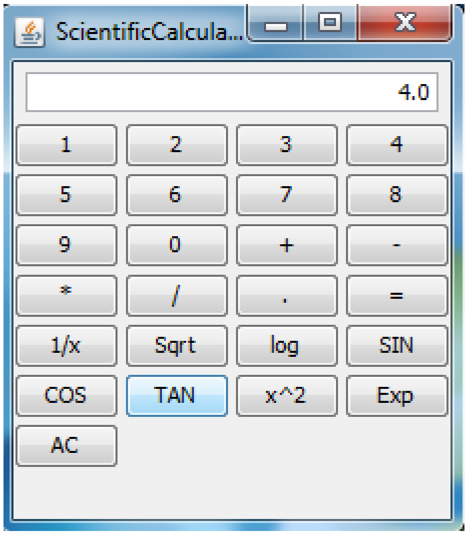
OUTPUT:

Addition[12+64]



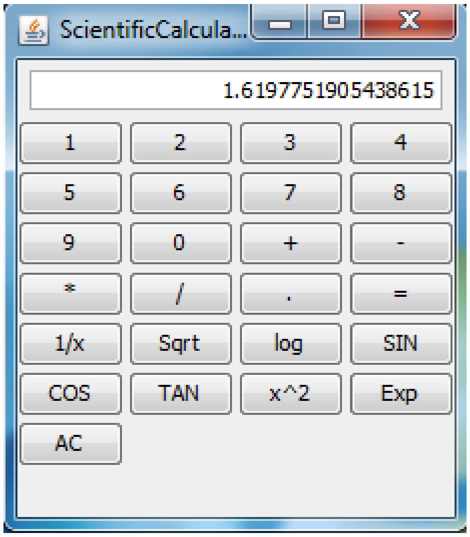
```
74.0
```

SquareRoot [16]



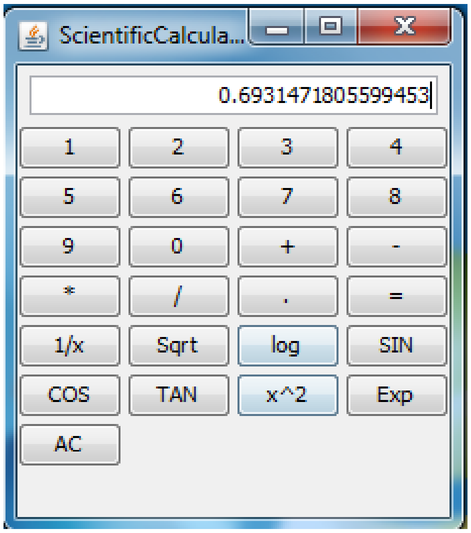
```
4.0
```

Tan45



```
1.6197751905438615
```

Log2



```
0.6931471805599453
```

RESULT:

Thus the Implementation for designing the scientific calculator has been successfullyexecuted.

TOPIC BEYOND THE SYLLABUS

Ex.No.12 **Implementation of JDBC**

Date:

Aim:

To develop a simple system to insert student details into the database using JDBC to connect to a back-end database.

Algorithm:

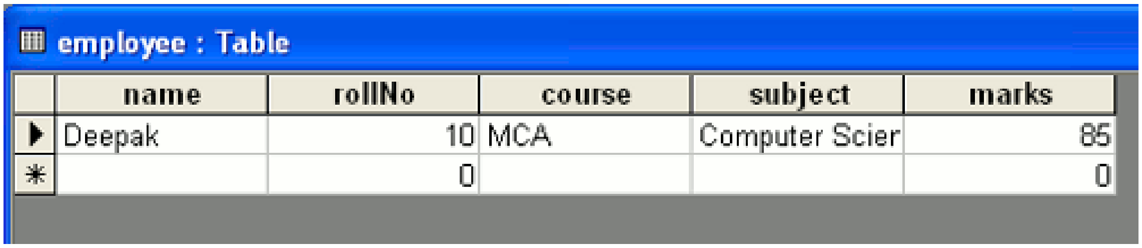
1. Create a Database
2. Create a table employee in Database created.
3. Insert the fields into the employee table with the following fields: name, rollno, course, subject, marks.
4. Create another panel consisting of buttons UserID, BookID, Return/Renewal,Fine.
5. Associate these buttons with listeners(with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

Program:

```
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);

}

}

}

}
```

Output:



employee : Table

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

RESULT:

Thus the Student details are inserted successfully into the database.